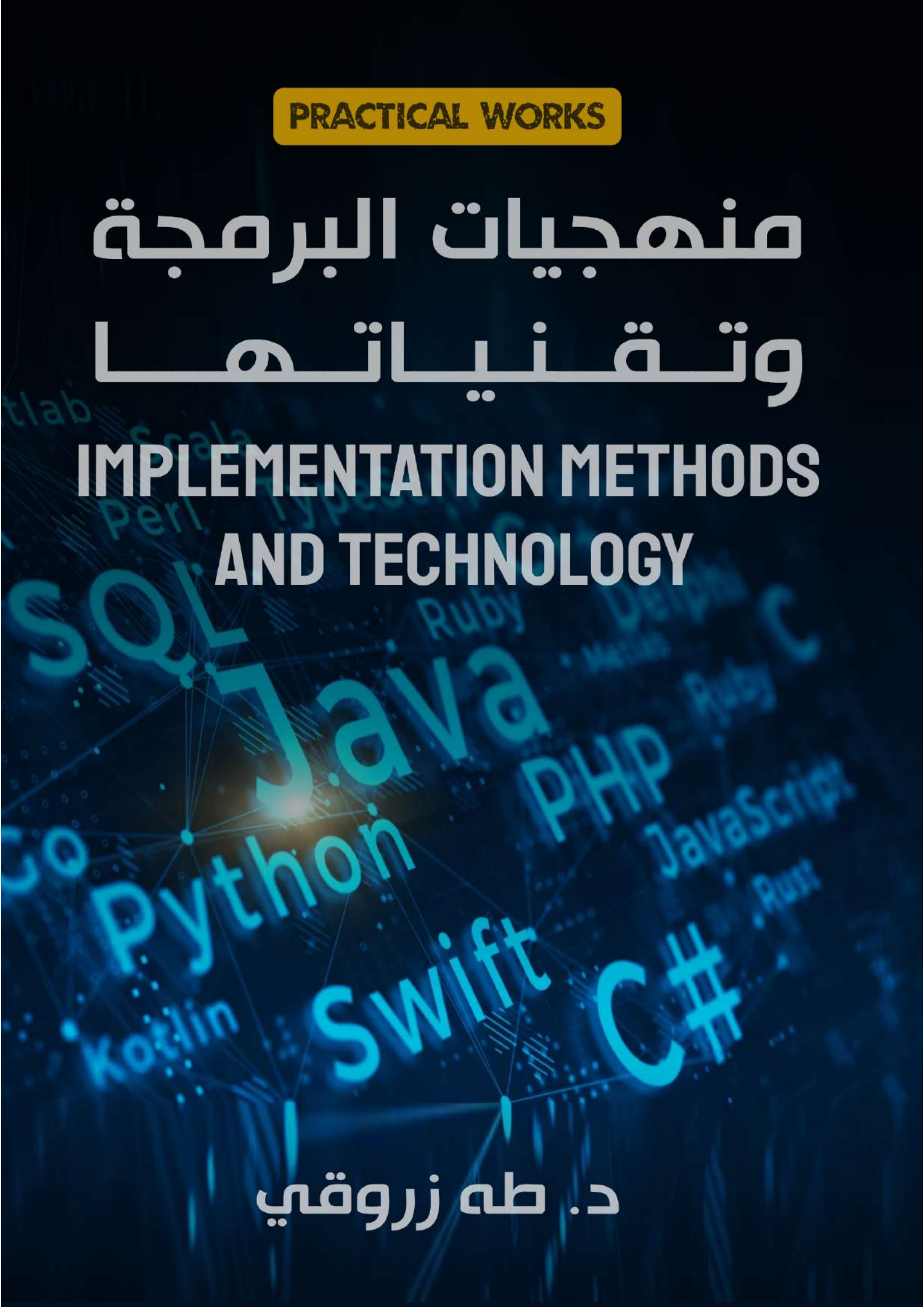
PRACTICAL WORKS

منهجيات البرمجة وتقنياتها

IMPLEMENTATION METHODS AND TECHNOLOGY



د. طه زروقي

Contents

1 Practical works أعمال تطبيقية **1**

2 Practical works أعمال تطبيقية **4**

2.1 How to write a report . . . 5
2.1.1 كيفية إعداد تقرير . . . 6
2.2 Practical works n°1 العمل التطبيقي رقم 1 . . . 8
2.2.1 Part 1 . . . 8
2.2.2 Part 2 . . . 8
2.2.3 Home work . . . 12
2.2.4 References . . . 14
2.2.5 Additional exercises تمارين للتعمق . . . 14
2.3 Practical works n°2 العمل التطبيقي رقم 2 . . . 16
2.3.1 Part 1 . . . 16
2.3.2 CSS example : directory.css . . . 18
2.3.3 XSLT Example . . . 19
2.3.4 Homework العمل المنزلي . . . 20
2.3.5 Project : مشروع . . . 21
2.3.6 Addtional questions (optional) أسئلة إضافية (اختيارية) . . . 21
2.3.7 References . . . 21
2.3.8 Additional exercises تمارين للتعمق . . . 22
2.4 Practical works n°3 العمل التطبيقي رقم 3 . . . 23
2.4.1 Part 1 . . . 23
2.4.2 Homework العمل المنزلي . . . 27
2.4.3 Projet : مشروع . . . 27
2.4.4 Addtional questions (optional) أسئلة إضافية (اختيارية) . . . 28
2.4.5 References . . . 28
2.4.6 Additional exercises تمارين للتعمق . . . 28
2.5 Practical works n°4 العمل التطبيقي رقم 4 . . . 29
2.5.1 Part 1 . . . 29
2.5.2 Part 2 . . . 31
2.5.3 Homework العمل المنزلي . . . 33
2.5.4 References . . . 33
2.5.5 Additional exercises تمارين للتعمق . . . 33
2.6 Practical works n°5 العمل التطبيقي رقم 5 . . . 34
2.6.1 Part Practical work . . . 34
2.6.2 Part 1 . . . 34
2.6.3 Part 2 . . . 34
2.6.4 Part 3 . . . 36
2.6.5 Part 4: MVC + graphical Interface . . . 37
2.6.6 Part 5: MVC + graphical interface . . . 39
2.6.7 Homework واجب منزلي . . . 41

2.6.8 Addtional questions (optional) أسئلة إضافية (اختيارية) . . . 41
2.6.9 References مراجع . . . 41
2.6.10 Additional exercises تمارين للتعمق . . . 41
2.7 Practical works n°6 العمل التطبيقي رقم 6 . . . 42
2.7.1 Part TP . . . 42
2.7.2 Part 1 . . . 42
2.7.3 Part 2 Abstract Factory . . . 44
2.7.4 Part 3 Singleton . . . 45
2.7.5 Part 4 Builder Pattern . . . 46
2.7.6 Home work واجب منزلي . . . 49
2.7.7 References مراجع . . . 49
2.7.8 Additional exercises تمارين للتعمق . . . 49
2.8 Practical works n°7 العمل التطبيقي رقم 7 . . . 50
2.8.1 Part TP . . . 50
2.8.2 Adapter المحوّل . . . 50
2.8.3 Facade الواجهة . . . 51
2.8.4 Proxy الوكيل . . . 53
2.8.5 Behavioral design patterns: Strategy . . . 55
2.8.6 Behavioral Design patterns :Observer . . . 57
2.8.7 Homework واجب منزلي . . . 59
2.8.8 Addtional questions (optional) أسئلة إضافية (اختيارية) . . . 59
2.8.9 References مراجع . . . 59
2.8.10 Additional exercises تمارين للتعمق . . . 59
2.9 Practical works n°8 العمل التطبيقي رقم 8 . . . 60
2.9.1 Part TP . . . 60
2.9.2 Fontion type() . . . 60
2.9.3 isinstance() function . . . 61
2.9.4 vars() function . . . 62
2.9.5 dir() function . . . 62
2.9.6 Extracting Python class details استخراج تفاصيل صنف بايثون . . . 63
2.9.7 Dynamic modification of attributes التعديل الديناميكي للخصائص . . . 63
2.9.8 ___call___() Method . . . 64
2.9.9 Application تطبيق . . . 65
2.9.10 Homework واجب منزلي . . . 66
2.9.11 References مراجع . . . 66
2.10 Practical works n°9 العمل التطبيقي رقم 9 . . . 68
2.10.1 Module Creation إنشاء وحدة برمجية . . . 68
2.10.2 Test unit creation إنشاء وحدة فحص . . . 70
2.10.3 Decorators Use استعمال المُزخرفات . . . 70
2.10.4 Aspects oriented programming البرمجة جانبية التوجه . . . 72
2.10.5 Homework واجب منزلي . . . 73
2.10.6 References مراجع . . . 73

3 Tests فحوص 74
3.1 Tests n°1 . . . 75
3.1.1 Topic n°1 . . . 75
3.1.2 Topic n°2 . . . 75
3.1.3 Topic n°3 . . . 76
3.2 Tests n°2 . . . 77
3.2.1 Topic n°1 . . . 77
3.2.2 Topic n°2 . . . 77

4 Test solutions حلول الفحوص **79**
4.1 Chapter one test solutions . . . 80
4.1.1 Solution n°1 . . . 80
4.1.2 Solution n°2 . . . 81
4.1.3 Solution n°3 . . . 83
4.2 Chapter one test solutions . . . 87
4.2.1 Solution n°1 . . . 87
4.2.2 Solution n°2 . . . 89

5 Exams امتحانات **95**
5.1 Exams امتحانات . . . 96
5.1.1 Topic n°1 . . . 96
5.1.2 Topic n°2 . . . 96

6 Exams Solutions حلول الامتحانات **99**
6.1 Exam solutions حلول امتحانات . . . 100
6.1.1 Solution n°1 . . . 100
6.1.2 Solution n°2 . . . 103

7 Appendices ملحقات**112**
7.0.1 Books كتب . . . 113
7.0.2 Courses online دروس أونلاين . . . 113
7.0.3 Software برامج وتطبيقات . . . 113
7.0.4 Educational objectives . . . 114
7.0.5 Content of the material: . . . 114
7.0.6 References . . . 114
7.1 Glossary مسرد . . . 115

Bibliography **116**

Chapter 1

Practical works أعمال تطبيقية

مقدمة Préface

كتاب "منهجيات البرمجة وتطبيقاتها" كتاب أعمال تطبيقية، موجهة لطلبة السنة الأولى ماستر إعلام آلي، تخصص "هندسة أنظمة الإعلام الآلي" بجامعة البويرة و شعب الإعلام الآلي في الجامعات الجزائرية،
ويحتوي في هذا الكتاب على الأعمال التطبيقية الخاصة بـ :

- لغة البرمجة بيثون
- لغة XML
- أنماط التصميم

ويحوي الكتاب عددا من الأعمال التطبيقية الموجهة، ومشاريع وأعمالا منزلية، مقسّمة حسب الفصول، قسم كبير منها محلول، وكذلك قسم خاص بفحوص التقويم المستمر مع تصحيحها، وقسم آخر للامتحانات.
ويأتي هذا الكتاب ثمرة لخبرة اكتسبتها في التدريس في جامعة البويرة لسنوات عديدة في قسم الإعلام الآلي.
أتمنى أن يلقى هذا الكتاب القبول، ونرحب بالملاحظات والتوصيات لتحسينه مستقبلا.
المؤلف: د. طه زروقي
gmail(dot)com (at) taha(dot)zerrouki

عن المؤلف

الدكتور طه زروقي، أستاذ بجامعة البويرة في قسم علوم الحاسوب، متخرج من المدرسة الوطنية العليا للإعلام الآلي، مطوّر برمجيات حرة مفتوحة المصدر خاصة باللغة العربية مهتم ب :

- المعالجة الآلية للغات الطبيعية
- المصادر المفتوحة

قدّم دروسا في :

- بنية الآلة ومعمارية الحاسوب،
- برامج إدارة المشاريع
- لغات البرمجة

موقع : http://tahadz.com

This Book uses the ”mathbook.cls v1.41” class developped by Stéphane PASQUET.

The cover page made by Haithem Benhalima : haithem_bhm @ intagram

Many exercises and solutions were generated automaticaly by ”STRM-Test” project developed by the Author, available on github [1].

جزيل الشكر للطالب هيثم بن حليمة لمساهمته في تصميم الغلاف، والشكر موصول لكل من ساهم من قريب أو بعيد في صياغة هذا الكتاب.

تم إعداد المصطلحات بتصرف وفقا للمصادر الآتية، مع مراعاة مطابقتها للمنهاج الدراسي في الثانوية: (**compulex1990**), (**inforba2004**), (Zerrouki, 2012), (**zerrouki2013**).

version 1.0, date: September 9, 2023.

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.

[1] https://github.com/linuxscout/strm-tests

Chapter 2

Practical works أعمال تطبيقية

2.1 How to write a report

Note : The recommendations for preparing a report must be followed.

2.1.0.1 Title page

You must specify the following information on the guard page:

- Binomial names
- Specialty and Group
- Module
- Practical work number + Project Title.

2.1.0.2 Report Structure :

1. **Introduction** : explaining the problem to be solved and the questions asked
2. **Methodology** : the technique used (methods and algorithms)
3. **Tools** : to cite the libraries, frameworks, tools to realize the work.
   Avoid representing tools that are presented in lesson or already in use.
4. **Tests and results**: present examples of input data and results after execution.
5. **Insert code in appendices** : structure the code and provide it with comments explaining the important things.

2.1.0.3 Remark :

- No more than two people (a pair)
- Use only clamps (without bookbinder).
- Avoid plagiarism (all similar works are rejected and rated to Zero)
- If you want to help your colleague, help him learn.
- Colors are not necessary.

2.1.0.4 Submission :

- The report must be submitted within the scheduled dates.
- Submission of the report online under the platform.
- The delay in submitting the report is taken into account.
  To submit:
  → The report of the work done during the practical session.
  → Home working.

2.1.1 كيفية إعداد تقرير

ملحوظة : يجب اتباع توصيات إعداد تقرير العمل التطبيقي.

2.1.1.1 صفحة الغلاف

في **صفحة الغلاف** ، يجب عليك تحديد المعلومات التالية:

- أسماء أعضاء الفريق
- المستوى التخصص / الفوج
- اسم المادة
- رقم العمل التطبيقي و عنوان المشروع.

2.1.1.2 هيكل التقرير:

1. **مقدمة:** شرح المسألة والمطلوب
2. **منهجية العمل:** التقنية المستخدمة (الطرائق والخوارزميات)
3. **الأدوات:** اذكر المكتبات والأدوات وأطر العمل اللازمة لإنجاز المطلوب.
   تجنب ذكر الأدوات التي سبق تقديمها في الدرس أو سبق استخدامها.
4. **الاختبارات والنتائج:** تقديم أمثلة لبيانات الإدخال والنتائج بعد التنفيذ.
5. **متن البرنامج المصدري في الملحق:** صُغ البرنامج صياغة جيدة، وزوّده بتعليقات تشرح الأشياء المهمة.

2.1.1.3 ملاحظات

- لا يزيد الفريق عن عضوين اثنين
- أمسك الورق بالدبّاسة (لا داعي لاستعمال تغليف).
- تجنب الغش والانتحال (تُرفض جميع الأعمال المتماثلة وتُصفّر)
- إذا كنت ترغب في مساعدة زميلك ، ساعده على التعلم.
- الطباعة الملونة ليست ضرورية.

2.1.1.4 تسليم

- يجب تسليم التقرير في المواعيد المحددة.
- إعادته إلى الإنترنت تحت المنصة
- يُحتسب التأخر في التسليم

- يجب تسليم :

→ تقرير الأعمال المنفذة خلال حصة الأعمال التطبيقية.
→ الواجب المنزلي.

2.2 Practical works n°1 العمل التطبيقي رقم 1

- **Chapter 1**: Programming Language typology الفصل 1

2.2.1 Part 1

Objectives : Learn functional programming

Toolsأدوات : Microsoft Excel

A first-year master's student is admitted to the second year, according to the following conditions:

- The two semesters are acquired with an average greater than 10 for each semester.
- There is no compensation between the two semesters (update in new conditions).
- For the debt transition, the student must have 45 credits.

Word to do

- Create the formula in Excel, which gives the overall averages and the results of the students given.
- Calculate the number of students admitted, use the Countif function (in french(NB.SI)).
- Determine the name of the promotion major using the vlookup function (in french(RechercheV)).

ينتقل الطالب في الماستر من السنة الأولى إلى الثانية حسب ما يلي:

- معدل السداسيين أعلى من أو يساوي 10 لكل سداسي.
- لا تعويض بين السداسيين (تم تعديلها).
- ينتقل بالديون إذا حصل على 45 رصيدًا.

المطلوب

- أعط معادلة في الأكسيل لحساب المعدل ونتيجة الطلبة في الجدول. استخدم الدالة SI
- احسب عدد الطلاب المقبولين ، استخدم دالة COUNTIF (بالفرنسية NB.SI)
- حدد اسم الطالب المتفوق باستخدام وظيفة VLookup (بالفرنسية RechercheV)

Name	S1	S2	Credits	Result
Ahmed ben Abdelkader	10	12	60	
Boualem Salim	12.5	9.5	47	
Hamdani Hamid	14	8	44	
Mansour Tarek	9.5	9.99	30	

2.2.2 Part 2

Objectives : Learn Python language

Toolsأدوات : Python, PyScripter/PyCharm

Work to do للعمل

- Install Python under Windows or Linux تثبيت بيثون على وندوز أو لينكس
- Run a Python script تنفيذ برنامج بلغة بيثون
- Write programs with Python كتابة برامج بلغة بيثون

Run the following code نفّذ ما يلي

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# character encoding declaration, often used UTF-8,
# to support all multilingual.
print("Salam Alykom")
#print(u"    ("

#  Variables declaration
# no explicit declaration
# just assign a value to specify the type of a variable
age = 125 # integer
print ("type of age", type(age))
salaire = 3.14
print ("type of salaire", type(salaire))
name = "Ahmed"
print ("type of name", type(name))

#substitution
s2 = "I am %s"%name
s3 = "I am %d old "%age
print(s2, s3)


# listes
liste = [1,2,31,4]
print ("type of ls", type(liste))
print (liste)
# add an element
liste.append(5)
print (liste)

# word list

wordlist = ["I", "am", "learning", "english"]



# Flows Control
a = 15
if a > 15:
    # pay attention to tabs
    print ('a is greater than 15')
```

```
elif a <= 15 and a >= 10:
    print ('a between  10 and 15')
else:
    print ('a is less than 10')



# test if an element belongs to a list
a = 5
if a in liste:
    print ("a exists in list")


# Loops
for word in wordlist:
    print (word)

# Loops and tests
# Count occurrences of a character in a string
chaine = """The sky, oh the sky! It stretches above us, vast and endless.
The sky is a canvas, a canvas of blue, blue like the ocean on a clear summer day.
 It's a canvas of possibilities, possibilities as vast as the sky itself."""
caractere="e"
cpt = 0
for c in chaine:
    if c == caractere:
        cpt += 1
print ("Char %s exists %d times"%(caractere, cpt))
```

Run the following code: be careful the code contains errors.
نفّذ البرنامج الآتي، انتبه فيه بعض الأخطاء.

```
def clean_text(text):
     "
        remove punctuation  form a text
     '
     punct = '+.*/?,;\'#
       for  p  in  punct
              text  =  text.replace(p,")
      return   text

def  tokenize(text)
    "
       Convert  text into tokens,  return  a list of  tokens(words)
       "''
      return  text.split( )
def tokenize2(text)
      '
      Convert  text  into tokens,  return a list  of tokens(words)
       "'
      spaces = r\s+
```

```
    words = re.split(spaces,  text)
        return word


def  word_freq(words):

      Counts  words and return a  dictionary of words  with  their occurrences

       index  = {}
    for w in  words:
             if w in  index
              index[w]  += 1
           else
                index[w] =  1
      return index
def most_common_word(words_freq_table)
       '
    Counts words  and return a  dictionary of  words with their  occurrences
      ''
      frequent =
     frequency  = 0
       for  word in  words_freq_table
            if  words_freq_table[word] >  frequency:
              frequent  =  word
                 frequency =  words_freq_table[word]
      return frequent



def read_file(filename)
     ''  Read  a  text from  file

      text =  "'
      try
            fl =  open(filename)
      except:
              print(Can't open  file,  ",  filename)
            sys.exit()
     #  if  success
       text  = fl.read()
      lines = fl.readlines()
     return text

def  main(args):
    text =  ""  Surprise steepest recurred landlord mr  wandered amounted  of.
     Continuing  devonshire but  considered its.  Rose  past oh shew  roof
    is song neat. Do depend  better praise do friend garden  an wonder  to.
        Intention age nay otherwise  but  breakfast.
        Around garden beyond  to  extent by.
      "
     text =  clean_text(text)
```

```
72          print(text)
73        #  tokenize  text
74         words  = tokenize(text)
75        print(words)
76          words_nb = word_freq(words)
77         print(words_nb)
78       data  = read_file("data.txt)
79         print(data)
80         text  =  data
81        words  =  tokenize(text)
82         print(words)
83         word_freq_table = word_freq(words)
84        print(word_freq_table)
85         freqw  =  most_common_word(word_freq_table)
86          print(most frequent word is  ',  freqw,  word_freq_table[freqw]  )
87         return  0
```

2.2.3 Home work

Realize the following project using Excel, then Python (without using libraries).

- Writing a python program to read data from a text file.
- The data is stored in form of fields separated by tabs.
- The result is displayed as a web page.

2.2.3.1 Project :

Management of student mentoring in a module

Data :

Student list : Level, specialty, group, student number in the group, first name, family name.

Mentoring :

- Number of sessions done by the group عدد الحصص.
- The number of lessens taken by the student (a justification is accounting as 0.5).
- Test scores (two tests).
- participation scores.
- Exam note / make up exam note علامة الامتحان والاستدراك.

Fields to calculate :

- Assiduity rating: a credit of 3 points, each unjustified absence subtract one point, one justification is accounted for as 0.5. Minimum rating is 0.
- Exclusion: A zero in presence results in an exclusion.
- TD rating: attendance on 3, participation on 2,
- Average of module معدل المادة.

Work to do :

- Calculate student scores.
- Show list of non-admitted students.
- Show student statistics by results per session.

Additional questions (optional)

- Redo the same work with Pandas library.

أنجز المشروع باستخدام Excel ثم باستخدام بايثون (دون مكتبات خاصة):

- اكتب برنامج بيثون لقراءة البيانات من ملف نصي
- البيانات مخزنة كحقول مفصولة بعلامات جدولة.
- عرض النتيجة كصفحة ويب.

مشروع : إدارة متابعة الطلاب في مادة معينة
المعطيات : : **قائمة الطلاب:** المستوى ، التخصص ، الفوج، المادة ، رقم الطالب في المجموعة ، اللقب ، الاسم الأول،

المتابعة :

- عدد الحصصة المنجزة في الفوج
- عدد الحصص التي حضرها الطالب (يحسب التبرير 0.5)
- علامة الفحوص (فحصان)
- علامة المشاركة.
- علامة الامتحان / الاستدراك

الحقول المطلوب حسابها: :

- علامة الحضور: رصيد قدره 3 نقاط ، ويطرح كل غياب غير مبرر نقطة ، ويحسب التبرير 0.5. الحد الأدنى للدرجات 0.
- الإقصاء: علامة صفر في الحضور تقصي.
- علامة الأعمال الموجهة: الحضور من 3 ، المشاركة من 2 ،
- معدل المادة.

العمل المطلوب :

- حساب نتائج الطلاب
- عرض قائمة الطلاب غير المقبولين.
- عرض إحصائيات الطلاب بناءً على نتائج كل دورة (عادية/استدراكية).

أسئلة إضافية (اختياري) : :

- إعادة العمل بمكتبة الباندا.

2.2.4 References

- علي آل ياسين، مختصر دليل لغات البرمجة، 2014 (علي آل ياسين, citepalyassen20162016)
https://alyassen.github.io/Brief-guide-to-programming-languages-v1.2.4.pdf

2.2.5 Additional exercises تمارين للتعمق

2.2.5.1 Exercise

Represent the following algorithm in functional language: We want to calculate the Zakate for an amount with a period of savings expressed in the number of months. The zakate is calculated from 2.5 % of the amount if it exceeds the quorum and a period of savings more than one year. Otherwise 0.

اكتب خوارزمية بلغة وظيفية :
لحساب الزكاة لمبلغ مُدّخر منذ فترة نعبّر عنها بالأشهر. مبلغ الزكاة هو 2.5 % من المبلغ المُدّخر إذا بلغ النصاب، وحال عليه الحول. الحول. الحول. الحول. الحول. الحول. الحول. الحول.

2.2.5.2 Exercise

For module M, one student passes three tests of 15 points each, the final score of the test is the average of the two best scores. Give the Excel formula for calculating this note.
في مادة معينة، يجتاز الطالب ثلاثة فحوص كل منها على 15 نقطة. تحسب العلامة النهائية بالمعدّل بين أفضل علامتين.
عبّر عنها بصيغة إكسيل.

2.2.5.3 Exercise

The given code represents the McCathy function in the functional language "Lisp"; convert the code into Python.

إليك دالة ماك كاثي مكتوبة باللغة الوظيفية Lisp، ترجمها إلى لغة بيثون

```
(defun mc91 (n)
     (cond ((<= n 100) (mc91 (mc91 (+ n 11))))
       (t (- n 10))))
```

2.2.5.4 Exercise

Let have the follwing code for Fibonacci written in « Lisp » functional language,

- Run the code pour n = 5, track results.
- Translate the code into Python.

لتكن دالة فيبوناتشي مكتوبة باللغة الوظيفية "Lisp"، نفّذ الدالة من أجل n=5 ، ثم ترجمها إلى لغة بيثون.

```
(defun fb(n)
    (cond
      ((eq n 1) 0)
      ((eq n 2) 1)
    ((+ (fb (- n 1)) (fb (- n 2))))
 ))
```

Notes : cond : if ; eq : equal

2.2.5.5 Exercise

Let the functional language LF denote the operators (+,-,*, /, ^, loop for) by (sum, sub, mult, div, pow, for) respectively. Express the following equation in LF.

لتكن اللغة الوظيفية LF، حيث نعرّف العوامل (+,-,*, /, ^، حلقة التكرار المعدود "من أجل") بالدوال for pow, div, mult , sub, sum, على الترتيب.

عبّر عن المعادلة الآلية بلغة LF .

$$f(x) = x^n - \sum_{i=0}^{n} \frac{a+i}{n^2}$$

2.2.5.6 Exercise

Write a python program using the ”re” library to:

- Verify an email address.
- Verify a phone number.
- Check the strength of a password.

اكتب برنامجا بلغة بيثون واستعمل مكتبة re للتحقق من صحة عنوان بريد الكتروني، وصلاحية رقم هاتف وقوة كلمة السر.

2.3 Practical works n°2 العمل التطبيقي رقم 2

- **Chapter 2**: XML الفصل 2

2.3.1 Part 1

Objectives : Handle XML/DTD/CSS/XSL

Data : directory.xml

Toolsأدوات : xml copy editor

Questions : Here is the following file directory.dtd
إليك ملف تعريف أنواع البيانات directory.dtd الآتي:

```
<!ELEMENT directory (person*)>
<!ELEMENT person (identity)>
<!ELEMENT identity (familyname,firstname,address)>
<!ELEMENT address (street,zip,city,phone)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT zip (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT address (tel)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
```

1 Check th validity of the directory-v0.1.xml file against the DTD:
تحقق صلاحية الملف directory-v0.1.xml بالنسبة إلى ملف التعريف DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE directory SYSTEM "directory.dtd">
<directory>
    <person>
        <identity>
            <familyname>Achour</familyname>
            <firstname>Hamza</firstname>
            <address>
                <street>25 avenue de 05 juillet 1962 </street>
                <zip-code>16000</zip-code>
                <city>Alger</city>
                <phone type="mobile">071002500</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>AOmar</familyname>
            <firstname>Ahmed</firstname>
```

```
            <address>
                <street>18 avenue du Ibn Badis</street>
                <zip-code>10000</zip-code>
                <city>Bouira</city>
                <phone type="fix">026942525</phone>
                <phone type="mobile">050252525</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>Alouache</familyname>
            <firstname>Ayoub</firstname>
            <address>
                <street>18 avenue du Didouche Mourad</street>
                <zip-code>42000</zip-code>
                <city>Tipaza</city>
                <phone type="fix">025942525</phone>
                <phone type="mobile">051252525</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <firstname>Razika</firstname>
            <familyname>Khaladi</familyname>
            <address>
                <street>18 Ahmed Ben Aissa</street>
                <zip-code>10000</zip-code>
                <city>Bouira</city>
                <phone type="fix">026666666</phone>
                <phone type="mobile">052525250</phone>
            </address>
        </identity>
    </person>
    <person>
        <identity>
            <familyname>Benkadour</familyname>
            <firstname>Ahmed</firstname>
            <address>
                <street>18 avenue du Ibn Badis</street>
                <zip-code>10000</zip-code>
        </address>
        </identity>
    </person>
        <person>
                <identity>
            <familyname>Ribiai</familyname>
            <firstname>Hamada</firstname>
            <address>
                <street>18 avenue du Ibn Badis</street>
                        <city>Bouira</city>
```

```
                                </address>
        </identity>
    </person>
</directory>
```

- Use XMLcopy editor, menu XML/validate
- Under Notepad++ go to Plugins > Plugin manager > Show Plugin Manager then find Xml Tools plugin. Tick the box and click Install.
- في برنامج Notepad++ إلى إضافات / إدارة الإضافات/ اعرض مدير الإضافات ثم ابحث عن أدوات Xml. اخترها ثم اضغط للتثبيت.
- Under linux : على لينكس استعمل

```
xmllint --noout --dtdvalid test.dtd test.xml
```

2 Update DTD, and correct possible errors on the XML file.

حدّث ملف التعريف DTD وصحّح الأخطاء المحتملة في ملف XML.

2.3.2 CSS example : directory.css

```
Directory{ background-color: #8A1619 }
person {    display: table; }
familyname {     color: #fff;
    font-family: 'Righteous', cursive;
    font-size: 40px;
    line-height: 60px;
    text-transform: uppercase;
    text-shadow: 2px 2px 0 #000, margin: 10px 0 24px;
}
firstname {font-size: 30px;
    color: #DFBF84;
    text-shadow: 1px 1px 0 #000, margin: 10px 0 24px;
    line-height: 50px;
}
address {display: table;
    color: #f6f6f6;
    font-size: 16px;
    line-height: 24px;
}
zip-code { padding: 15px; font-size: 20px; }
phone { color: #e8d4a7;  }
```

Link directory.css style sheet to directory.xml by using the following tag:

اربط ورقة النمط directory.css بالملف directory.xml باستعمال الوسم الآتي

```
<?xml-stylesheet type="text/css" href="directory.css"?>
```

- Can we modify the CSS file to display only the last name, first name, and city?
- Can we modify the CSS file to display the first name before the family name?

هل يمكننا تعديل ملف CSS لعرض الاسم واللقب والمدينة فقط؟

هل يمكننا تعديل ملف CSS لعرض الاسم قبل اللقب؟

2.3.3 XSLT Example

- Create the directory.xsl file — أنشئ ملف directory.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" encoding="utf-8"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"/>
<xsl:template match="/">
<html xmlns="http://www.w3.org/1999/xhtml">
<body>
<h1 align="center">Directory</h1>
<h2>People List</h2>
<xsl:for-each select="directory/person">
    <h3><xsl:value-of select="identity/familyname"/>
     <xsl:value-of select="identity/firstname"/>
     </h3>
    <p><xsl:value-of select="identity/address"/></p>
<hr/>
    </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Link directory.xsl style sheet to directory.xml by using the following tag:

اربط ملف النمط directory.xsl بملف البيانات directory.xml باستعمال الوسم:

```
<?xml-stylesheet type="text/xsl" href="directory.xsl"? >
```

- Peut-on modifier le fichier XSL pour afficher que le nom, le prénom, et la ville ?
- Peut-on modifier le fichier XSL pour afficher le prénom avant le nom?
- Can we modify the XSL file to display only the family name, first name, and city?
- Can we modify the XSL file to display the first name before the family name?

هل يمكننا تعديل ملف XSL لعرض الاسم واللقب والمدينة فقط؟

هل يمكننا تعديل ملف XSL لعرض الاسم قبل اللقب؟

Modify the code to display each person as follows:

عدّل الكود ليظهر كل شخص كما يلي:

Family name: Achour
First Name: Hamza
Street: 25 avenue de 05 juillet 1962
City: Alger
Phone:071002500

Modify the code to display a table that contains the list of people, as follows:
عدّل الكود ليظهر جدولا فيه قائمة الأشخاص كما يلي:

Directory

People List

Family Name	First Name	Address
AOmar	Ahmed	18 avenue du Ibn Badis 10000 Bouira 026942525 050252525
Benkadour	Ahmed	18 avenue du Ibn Badis 10000 Bouira
Ribiai	Hamada	18 avenue du Ibn Badis 10000 Bouira
Khaladi	Razika	18 Ahmed Ben Aissa 10000 Bouira 026666666 052525250

Modify the line
عدّل السطر :

```
<xsl:for-each select="directory/person">
```

by
بـ :

```
<xsl:for-each select="directory/person[identity/address/city='Bouira']">
```

You can see that we can show that people living in Bouira.
يمكننا أن نعرض قائمة من يعيشون في مدينة البويرة.

2.3.4 Homework العمل المنزلي

Perform the next project using Python and XML.

- Write a python program to read data from a text file.
- The data is stored in the form of fields separated by tabs.
- The result is displayed as an XML document.
- Define the DTD of the document, then validate the resulting document.
- Use a CSS style sheet for the presentation of the mentoring file.
- Use an XSLT style sheet for the presentation of the mentoring file.
- Use an XSLT style sheet to calculate the calculated fields.
- Use an XSLT style sheet to calculate the requested statistics.

أنجز العمل بلغة بيثون وXML

- اكتب برنامج بيثون لقراءة البيانات من ملف نصي
- البيانات مفصولة بعلامة جدولة.
- الناتج يعرض في شكل مستند XML.
- عرّف هيكلة البيانات بوساطة DTD ثم تحقق من صلاحية المستند الناتج.

- استعمل ورقة أنماط CSS لعرض ملف المتابعة.
- استعمل ورقة أنماط XSL/T لعرض ملف المتابعة.
- استعمل ورقة أنماط XSL/T لحساب الحقول المحسوبة .
- استعمل ورقة أنماط XSL/T لحساب الإحصائيات المطلوبة .

2.3.5 Project : مشروع

مشروع متابعة الطلاب في مادة ما، (work practical first to (Refer module a in mentoring student of Management (راجع العمل التطبيقي الأول)

2.3.6 Addtional questions (optional) أسئلة إضافية (اختيارية)

Let have the following file Quran-juza-amma.xml جزء عمّ

add refere into Qura amma.xm

- Define a DTD.
- Create a CSS style sheet.
- Create a XSL style sheet.
  - → Improve the display of sourates.
  - → Show the index of the sourates with their attributes (Makkia, Madania, Number of ayats, number of sourate).
  - → Add links to audio files for each sourate.
  - → Improve display for mobile devices.

ليكن الملف Quran-juza-amma.xml جزء عمّ

- → عرّف DTD للملف
- → أنشئ روقة أنماط CSS.
- → أنشئ روقة أنماط XLS/T.
  - * حسّن عرض السور
  - * اصنع فهرسا للسور مع خصائصها (مكية، مدنية، عدد الآيات، رقم السورة)
  - * أضف روابط لملفات صوتية لكل سورة
  - * حسّن العرض للأجهزة المحمولة

2.3.7 References

→ Quran Metadata from Tanzil project: https://tanzil.net/docs/quran_metadata
→ XML copy editor: https://xml-copy-editor.sourceforge.io
→ XSLT Elements Reference: https://www.w3schools.com/xml/xsl_elementref.asp

2.3.8 Additional exercises تمارين للتعمق

To be continued ... يُتبع

2.4 Practical works n°3 العمل التطبيقي رقم 3

- **Chapter 2**: XML الفصل 2

2.4.1 Part 1

Objectives : Handle XML Schema

Data : directory.xml, bank.xml, books.xml, quran-juz-amma.xml

Toolsأدوات : xml copy editor

Questions : Here is directory.xsd إليك ملف التعريف directory.xsd

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="directory">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="person" minOccurs="0" maxOccurs="unbounded">
    <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="identity">
      <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="familyname" type="xsd:string"/>
        <xsd:element name="firstname" type="xsd:string"/>
        <xsd:element name="address">
        <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="street" type="xsd:string"/>
          <xsd:element name="zip-code" type="xsd:string"/>
          <xsd:element name="city" type="xsd:string"/>
           <xsd:element name="phone" type="xsd:string"
           minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      </xsd:complexType>

      </xsd:element>
    </xsd:sequence>
    </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

1 Check th validity of the directory-v0.1.xml file against the XSD:

تحقق صلاحية الملف directory-v0.1.xml بالنسبة إلى ملف التعريف XSD:

- Use XMLcopy editor, menu XML/validate
- Under Notepad++ go to Plugins > Plugin manager > Show Plugin Manager then find Xml Tools plugin. Tick the box and click Install.
- في برنامج Notepad++ إلى إضافات / إدارة الإضافات/ اعرض مدير الإضافات ثم ابحث عن أدوات Xml. اخترها ثم اضغط للتثبيت.
- Under linux : على لينكس استعمل

```
xmllint --schema yourxsd.xsd yourxml.xml --noout
```

2 Correct possible errors on the XML file.

صحّح الأخطاء المحتملة في ملف XML.

3 Update the XSD to add attributes for phone element.

حدّث ملف التعريف XSD لإضافة خصائص لعنصر الهاتف.

```
<xsd:element name="phone" minOccurs="0" maxOccurs="unbounded">
<xsd:complexType>
<xsd:simpleContent>
  <xsd:extension base="xsd:string">
  <xsd:attribute name="type" type="xsd:string"/>
  </xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

Exercise 2 : Define the DTD and XML schema for the bank.xml file to meet the following requirements:

أنشئ ملف تعريف DTD ومخطط XSchema للملف bank.xml لتلبية المتطلبات التالية:

- There are two types of accounts: checking and savings accounts حساب جاري أو إدخار .
- the account ID is unique.
- the client ID is unique.
- 'c_id' refers to the client and 'ac $_id'to the account. The credit must be greater than -5000.$
- Use inheritance to derive current (checking) and savings accounts from a basic account type.

- الحسابات نوعان: حسابات جارية وحسابات إدخار.
- معرف الحساب (ID) وحيد.
- معرف الزبون (ID) وحيد.
- 'c_id' يشير إلى الزبون و'ac_id' يشير إلى الحساب.
- يجب أن يكون الرصيد أعلى من -5000 .
- استخدم الوراثة لاشتقاق الحسابات الجارية (checking) وحسابات الإدخار (saving) من نوع الحساب الأساسي.

```
<?xml version="1.0" encoding="UTF-8"?>
<bank>
  <accounts>
    <savings_accounts>
      <savings_account id="a1" interest="0.03">
```

```
        <balance>2500</balance>
      </savings_account>
      <savings_account id="a2" interest="0.03">
        <balance>15075</balance>
      </savings_account>
    </savings_accounts>
    <checking_accounts>
      <checking_account id="a3">
        <balance>4025</balance>
      </checking_account>
      <checking_account id="a4">
        <balance>-125</balance>
      </checking_account>
      <checking_account id="a5">
        <balance>325</balance>
      </checking_account>
    </checking_accounts>
  </accounts>
  <customers>
    <customer id="c1">
      <name>Ahmed Ben Rachidon</name>
      <address>Chaabet Yakhlaf, Bouira</address>
    </customer>
    <customer id="c2">
      <name>Marzouk Wail</name>
      <address>M'chadallah, BP 205</address>
    </customer>
    <customer id="c3">
      <name>Anfel Saidi</name>
      <address>Lakhdaria, Krariche 15b</address>
    </customer>
  </customers>
  <customer_accounts>
    <customer_account c_id="c1" ac_id="a2"/>
    <customer_account c_id="c1" ac_id="a3"/>
    <customer_account c_id="c2" ac_id="a4"/>
    <customer_account c_id="c3" ac_id="a1"/>
    <customer_account c_id="c3" ac_id="a5"/>
  </customer_accounts>
</bank>
```

Exercise 3 : Create an XML Schema for the document Book.xml

انشئ ملف Schema XML للمستند Book.xml.

Specifications مواصفات

1. the book item must have one or more books (one at least)
2. Each book must contain the following items in sequence: title, edition, author (one or more), publishedDate, type, and price.
3. Each book must contain an ISBN as a mandatory key attribute.
4. ”title” contains any character (type string).
5. ”title” contains a ”type” attribute that takes only the values ”P” and ”H”.

6 ”edition” is an optional element, and must contain a number not exceeding one byte.

7 There may be one or more authors (at least one author) for each book. Each author must have the following child elements: First, middle, Last. Items must be in the order First, middle, and Last, but the Middle is an optional element. The First, Middle, and Last elements are all string types, but the Middle must have no more than one character.

8 The publishedDate month element is defined as a type of unsignedByte with a minimum inclusive value of 1 and a maximum value of 12.

9 The publishedDate day element is defined as a type of unsignedByte with a minimum inclusive value of 1 and a maximum value of 31.

10 The ”publishedDate year” element is defined as a type of gYear.

11 The ”type” element is a string of characters and must be either ”fiction” or ”nonfiction” to be valid.

12 The price element is defined as a decimal type with only three characters to the left of the decimal comma and two to the right. The amount must be greater than zero.

1 يجب أن يحوي عنصر الكتب على كتاب واحد أو أكثر (واحد على الأقل).

2 يجب أن يحتوي كل كتاب على العناصر التالية بالتسلسل: العنوان، الطبعة، المؤلف (واحد أو أكثر)، تاريخ النشر، النوع، والسعر.

3 يجب أن يحتوي كل كتاب على رقم ISBN (الرقم الدولي للكتاب) كخاصية رئيسية إجبارية.

4 "العنوان" سلسلة حروف.

5 "العنوان" يحتوي على خاصية "type" تأخذ قيمتين "P" و"H" فقط.

6 "edition" الطبعة عنصر اختياري، وهو عدد لا يتجاوز بايت واحد.

7 للكتاب مؤلف واحد على الأقل. في اسم المؤلف نضع ثلاث عناصر، الاسم الأول، الأوسط والأخير بالترتيب. كل أقسام الاسم سلاسل حروف. الاسم الأوسط اختياري وبه حرف على الأكثر.

8 في "تاريخ النشر" نعرّف ثلاث قيم (الشهر، اليوم، السنة) العنصر شهر Month نوعه unsignedByte بقيمة دنيا 1 وقيمة قصوى 12.

9 اليوم من نوع unsignedByte ما بين 1 و 31.

10 السنة نعرّفها بالنوع gYear (السنة الغريغورية).

11 العنصر "النوع" عبارة عن سلسلة حروف تأخذ قيمتين فقط هما خيالي "fiction" أو واقعي "nonfiction".

12 نعرّف السعر بعدد عشري ذو ثلاثة أرقام قبل الفاصلة، ورقمين بعدها، ولا يكون معدومًا.

```
<?xml version="1.0" standalone="no"?>
<books>
<book isbn="AB9781418860646">
   <title type="P">XML Comprehensive Step by Step</title>
   <edition>2</edition>
   <author>
      <first>Ahmed</first>
      <middle></middle>
      <last>Amara</last>
   </author>
   <publishedDate>
      <month>7</month>
      <day>9</day>
      <year>2007</year>
   </publishedDate>
```

```
    <type>nonfiction</type>
    <price>70</price>
</book>
<book isbn="AB9780132240857">
    <title type="H">Systems Analysis and Design for Dummies</title>
    <edition>7</edition>
    <author>
        <first>Kemal</first>
        <middle>B</middle>
        <last>Kendali</last>
    </author>
    <author>
        <first>Djamel</first>
        <middle>T</middle>
        <last>Kendali</last>
    </author>
    <publishedDate>
        <month>3</month>
        <day>27</day>
        <year>2007</year>
    </publishedDate>
    <type>nonfiction</type>
    <price>166.67</price>
</book>
<book isbn="AB9780805001112">
    <title type="P">The World Map Trap</title>
    <author>
        <first>Peri</first>
        <middle>F</middle>
        <last>Rais</last>
    </author>
    <publishedDate>
        <month>9</month>
        <day>15</day>
        <year>1986</year>
    </publishedDate>
    <type>fiction</type>
    <price>14</price>
</book>
</books>
```

2.4.2 Homework العمل المنزلي

Use XML schema to define Student mentoring file.

استعمل مخطط Xschema لتعريف بيانات تسيير الطلبة ومتابعتهم.

2.4.3 Projet : مشروع

Management of student mentoring in a module (Refer to first practical work)

مشروع متابعة الطلاب في مادة ما، (راجع العمل التطبيقي الأول)

2.4.4 Addtional questions (optional) أسئلة إضافية (اختيارية)

Use XML schema to define Quran-juza-amma.xml جزء عمّ file.

عرّف هيكلة ملف القرآن الكريم بصيغة Xschema

2.4.5 References

- Quran Metadata from Tanzil project: https://tanzil.net/docs/quran_metadata
- XML copy editor: https://xml-copy-editor.sourceforge.io
- XML Schema Elements Reference: https://www.w3schools.com/xml/schema_intro.asp
- xsd-cheat-sheet https://github.com/DavidAbram/xml-schema-cheat-sheet/blob/master/xsd-cheat-sh
  md
- xsd-cheat-sheet https://github.com/davidabram/xml-schema-cheat-sheet/blob/master/cheat-sheet
  pdf
- XML Schema - Structures Quick Reference https://www.cheat-sheets.org/saved-copy/SchemaStructure
  pdf

2.4.6 Additional exercises تمارين للتعمق

To be continued ... يُتبع

2.5 Practical works n°4 العمل التطبيقي رقم 4

- **Chapter 2**: XML الفصل 2

2.5.1 Part 1

Objectives : Handle XML with DOM API

Data : directory.xml

Toolsأدوات : Python DOM API

Questions : Here is a Python script إليك برنامج بيثون الآتي.

```
#!/usr/bin/env python
#- * -coding: utf - 8 - * -##testdom2.py##
DATA_FILE = 'TP/directory-v0.2.xml'
import xml.dom.minidom as minidom
import sys
def main():
  try:
     xmldoc = minidom.parse(DATA_FILE)
  except:
     print "Can't Open the file",
     sys.exit()
  print xmldoc.toxml()
  return 0
if __name__ == '__main__':
  main()
```

What does this script do?
Create the following function, execute it in the main function:

ماذا يفعل هذا البرنامج؟
أنشئ الدالة التالية ونفذّها في الدالة الرئيسية main:

```
def treat_doc(xmldoc):
    # get the directory list
    directory = xmldoc.getElementsByTagName('directory')[0]
    print directory
    cpt = 0
    # display person by person
    for person  in directory.childNodes:
        print "-"*40
        print "person n°", cpt
        print person.toxml()
        cpt += 1
```

Explain the results
We want to display all the phone numbers in our file, we use the function

اشرح النتائج

نريد عرض جميع أرقام الهواتف في ملفنا، استخدم الدالة

```
def display_tel(xmldoc):
    # display only telephones
     # get the tel list
    telephones = xmldoc.getElementsByTagName('phone')
    print telephones
    cpt = 0
    # display tel by tel
    for tel  in telephones:
        print "-"*40
        print "Tel n°", cpt
        print tel.toxml()
        print "N°:",tel.firstChild.data
        print "Type:",tel.getAttribute("type")
        cpt += 1
```

Show phone numbers by person

اعرض أرقام الهاتف حسب الشخص

```
def display_tel_person(xmldoc):
    #get the persons list
    persons = xmldoc.getElementsByTagName('person')
    print persons
    cpt = 0
    # display telephone by person
    for person  in persons:
        print "-"*40
        print "person n°", cpt
        familyname    = person.getElementsByTagName('familyname')[0]
        firstname = person.getElementsByTagName('firstname')[0]
        tels   = person.getElementsByTagName('phone')
        print "*"*20
        print "Familyname:\t",familyname.firstChild.data
        print "Firstname:\t",firstname.firstChild.data
        for tel in tels:
            print "-"*20
            print "N°:",tel.firstChild.data
            print "Type:",tel.getAttribute("type")
        cpt += 1
```

Add new attribute to person element

أضف خاصية جديدة للعنصر person

```
def add_id_person(xmldoc):
    #get the persons list
    persons = xmldoc.getElementsByTagName('person')
    print persons
    cpt = 0
    # display person by person
    for person  in persons:
```

```
        print "-"*40
        print "person n°", cpt, person.nodeValue, person.nodeType,
        person.setAttribute('id', str(cpt))
        cpt += 1
        print person.toxml()
```

2.5.2 Part 2

Give the following code DOM tree

ارسم شجرة نموذج المستند DOM للكود الآتي

```
<html>
   <body>
      <div>
         <b>Familyname:</b> <span id="FirstName"></span><br/>
         <b>Firstname:</b> <span id="LastName"></span><br/>
         <b>Tel:</b> <span id="phone"></span><br>
         <b>city:</b> <span id="town"></span>
      </div>
</body>
</html>
```

Add the following javascript code, execute it.

أضف كود جافا سكريبت التالي، ثم نفّذه

```
 <script>
        //modify the HTML
        document.getElementById("firstname").innerHTML= "Taha";
        document.getElementById("lastname").innerHTML= "Zerrouki";
      </script>
```

Modifier le code javascript pour afficher le tel et la ville
Modify the javascript code to display the phone and the city

عدّل كود جافاسكريب الآتي لعرض الهاتف والمدينة.

```
//Create a <p> element
       var para = document.createElement("H1");
// Create a text node
        var t = document.createTextNode("This is a paragraph");
 // Append the text to <p>
        para.appendChild(t);
       para.setAttribute("align","center");
        document.body.appendChild(para);
```

The code that allows you to load an XML file

الكود الذي يحمّل ملف xml

N.B. To avoid any security issues when opening a file from the system, please run the file through a web server like Wamp or Xamp.

لتجنب أي مشاكل أمنية تتعلق بفتح ملف من النظام، يرجى تشغيل الملف من خلال خادم ويب مثل Wamp أو Xamp .

```
<script>
    if (window.XMLHttpRequest)
```

```
    //if browser supports XMLHttpRequest
    {// Create an instance of XMLHttpRequest object.
    // code for IE7+, Firefox, Chrome, Opera, Safari
     xmlhttp = new XMLHttpRequest();
    }
    else
    {// code for IE6, IE5
     xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    // sets and sends the request for calling "annuaire.xml"
    try {
      xmlhttp.open("GET","directory-v0.2.xml",async = false);
      xmlhttp.send();
       }
    catch(ex){
       alert("Exception: Can't open xml file");
       alert(ex.message);
      }

    // sets and returns the content as XML DOM
    xmlDoc=xmlhttp.responseXML;
    // parseing dom
</script>
```

Retrieve the data — استخرج البيانات

```
//parsing the DOM object
  document.getElementById("firstname").innerHTML  =
    xmlDoc.getElementsByTagName("familyname")[0].childNodes[0].nodeValue;
</script>
```

Display first name, phone and city. — اعرض الاسم والهاتف والمدينة

Display information in a table — اعرض المعلومات في جدول

```
<table  id="table_personnes" border=1>
<tr> <th>Familyname</th><th>Firstname</th><th>Phone</th><th>City</th></tr>
</table>
```

Displaying information in a table — عرض المعلومات في جدول

```
var identities = xmlDoc.getElementsByTagName("identity")
var table = document.getElementById("table_person")
 for(i in identities)
  {
  var tr = document.createElement("tr");          // create  a row
  var td = document.createElement("td");          // create  a row
  var familynames  = identities[i].getElementsByTagName("familyname");
  td.innerHTML = familynames[0].childNodes[0].nodeValue;
    tr.appendChild(td);       // add td to tr
     // redo for another data
```

```
11          tr.appendChild(td);
12          table.appendChild(tr);
13      }
```

Display first name, phone and city. اعرض الاسم والهاتف والمدينة

2.5.3 Homework العمل المنزلي

Redo the work of practical work no. 2 using the DOM API (Student Monitoring project).
أعد عمل العمل التطبيقي رقم 2 باستخدام API DOM (مشروع متابعة الطلاب).

2.5.3.1 Addtional questions (optional) أسئلة إضافية (اختيارية)

Use DOM API to manipulate the Quran-juza-amma.xml جزء عمّ file.
طبّق واجهة DOM البرمجية على مستند القرآن الكريم

2.5.4 References

- Quran Metadata from Tanzil project: https://tanzil.net/docs/quran_metadata
- Python xml.dom — The Document Object Model API: https://docs.python.org/3/library/xml.dom.html

2.5.5 Additional exercises تمارين للتعمق

We want to learn how to use remote APIs offered by websites in XML or JSON like:
نريد أن نتعلم كيفية استخدام واجهات برمجة التطبيقات API البعيدة التي تقدمها مواقع الويب بتنسيق XML أو JSON مثل:

- Meteo : https://openweathermap.org/
- geographical map : https://www.openstreetmap.org
- FootBall: https://rapidapi.com/api-sports/api/api-football/
- More API on https://rapidapi.com

We want to generate technical sheets from the API results, choose a theme:
نريد إنشاء بطاقات فنية من نتائج واجهة برمجة التطبيقات (API)، اختر موضوعًا:

- دليل سياحي (وصف المكان، الخريطة، الطقس، أوقات الصلاة، الخ)
- فريق كرة القدم (وصف، الصور، النتائج...)
- مقارنة الأسعار بين المواقع
- فهرس الأفلام

- A tourist brochure (description of the place, Map, Weather, Prayer times, etc.)
- Football team (description, images, results....)
- Price comparison between sites
- Movie index.
- etc...

2.6 Practical works n°5 العمل التطبيقي رقم 5

- **Chapter 2**: Design patterns (**MVC**) الفصل 2
- **Objectives** أهداف: MVC architecture
- **Data** بيانات: directory
- **Tools** أدوات : Python

2.6.1 Part Practical work

2.6.2 Part 1

Execute the following program: نفّذ البرنامج الآتي

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#  mvc-ann.py
# A directory in format of list [] of dictionaries {}
directory = [
        {"firstname":'Ahmed',  "familyname":'Mahdi', 'tel':'0778787887'},
        {"firstname":'Mohamed', "familyname":'Mahdi','tel':'0778787887'},
        {"firstname":'Mounir', "familyname":'Katibi','tel':'0778787887'},
        {"firstname":'Noui', "familyname":'Brahimi','tel':'0778787887'},
        ]
def main():
    print("Finding a telephone")
    print("Give a name")
    familyname = input()
    # found elements number
    nb_found = 0
    # browse names
    for person in annuaire:
         # show all people with the given name
        if person['familyname'] == familyname:
            print(familyname, person['prefamilyname'], person['tel'])
            nb_found += 1
    if not nb_found:
        print("This familyname %s doesn't exist "%familyname)

if __name__ == '__main__':
    main()
```

Que fait ce programme ? What does program do? ماذا يفعل البرنامج؟

2.6.3 Part 2

1 We want to reformulate the previous code by separating the display aspect from the data aspect.
نريد إعادة صياغة الكود السابق من خلال فصل جانب العرض عن جانب البيانات

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# A directory in format of list [] of dictionaries {}
class model:
    """ data model class"""
    def __init__(self):
        self.directory = [
        {"firstname":'Ahmed',  "familyname":'Mahdi', 'tel':'0778787887'},
        {"firstname":'Mohamed', "familyname":'Mahdi','tel':'0778787887'},
        {"firstname":'Mounir', "familyname":'Katibi','tel':'0778787887'},
        {"firstname":'Noui', "familyname":'Brahimi','tel':'0778787887'},
        ]
    def search(self, nom):
        """ lookup for a tel by lastname"""
        # found people list
        persons  = []
        for persn in self.directory:
         # show all people with the given name
            if persn["familyname"] == nom:
                persons.append(persn)
        # return a list of dictionaries
        return persons

class view:
    def __init__(self,):
        pass
    def input(self,):
        """ retrieve the name to search for"""
        print("Finding a phone")
        print("Give a name")
        name = input()
        return name
    def output(self, persons):
        """ display information from a list of persons"""
        print("List of found names")
        print(" %d found people "%len(persons))
        for pers in persons:
            print(pers["familyname"], pers["firstname"], pers['tel'])

def controller():
    # This program allows you to search the number
    # in a table by its name, and display
    # read data from the keyboard
    data_model =  model()
    myviewer = view()
    """  search a name """
    name = myviewer.input()
        persons = data_model.search(name)
    myviewer.output(persons)

if __name__ == '__main__':
```

```
    controller()
```

2 What are the advantages and disadvantages of this restructuring?

ما محاسن هذه الهيكلة ومساوئها؟

3 Modify the Model class to display all people in the directory.

عدّل صنف Model لعرض كل الأشخاص.

```
    def get_all(self):
        """ Find a tel by name"""
        # List of found people
        persons  = []
        for persn in self.directory:
         # display all people
            persons.append(persn)
        # a table with field names
        return persons
```

4 Modify the controller function, so that we can display all people,

عدّل دالة التحكم، لعرض جميع الأشخاص.

5 What should be modified in the View class in order to display all people.

ما الذي علينا تعديله في صنف العرض View لعرض جميع الأشخاص.

6 We now want to create a control class called controller instead of the controller function.

نريد الآن إنشاء صنف تحكم class Controller تسمى المُتحكّم بدلاً من دالة التحكم.

```
class controller:
    def __init__(self,):
        """ controller constrctor"""
        # init data model
        #~ self.data_model =  model()
        self.data_model =  model_file()
        self.viewer = view()

    def search(self):
        """  search a name """
        name = self.viewer.input()

        persons = self.data_model.search(name)

        self.viewer.output(personnes)
if __name__ == '__main__':
    contro = controller()
    contro.search()
```

2.6.4 Part 3

1 Now we want to add a new feature to our system, adding new people:

- Create an ”add” method in the data model.

- Create ”add” interface method
- Add a method to the controller that allows adding.

نريد الآن إضافة دالة جديدة إلى نظامنا، إضافة أشخاص جدد:

- إضافة الدالة "add" في نموذج البيانات.
- إضافة الدالة "add" إلى الواجهة
- إضافة دالة إلى المتحكم تسمح بالإضافة.

- ”Add method” in data model . دالة الإضافة في نموذج البيانات

```
    def add(self, name, firstname, tel):
        """ Add a person """
        person ={}
        person["familyname"]    = name
        person["firstname"] = firstname
        person["tel"] = tel
        self.directory.append(person)
```

- ”Add method” in View interface . دالة الإضافة في الواجهة

```
    def input_add(self,):
        """get name to add"""
        print("Add a  person")
        print("Give a name")
        name = input()
        print("Give a firstname")
        firstname = input()
        print("Give a firstname")
        tel = input()
        return (name, firstname, tel)
```

- ”Add method” in controller class . دالة الإضافة في المتحكم

```
    def add(self):
        """  Add a new name """
        name, firstname, tel = self.affichage.input_add()

        self.data_model.add(name, firstname, tel)
        persons = self.data_model.get_all()
        self.myviewer.output(persons)
        tel = input()
        return name, firstname, tel)
```

2 Edit the main program to add a person. عدّل البرنامج الرئيسي لإضافة شخص واحد

3 Edit the main program to add multiple people. عدّل البرنامج الرئيسي لإضافة أشخاص

2.6.5 Part 4: MVC + graphical Interface

1 We want to change the data model in order to store the information in a file. The data file is a text file that contains the fields (last name, first name, phone) separated by tabs.
Which part to modify (the model, the view, or the controller)?

نريد تغيير نموذج البيانات لتخزين المعلومات في ملف نصي.

ملف البيانات به حقول مفصولة بعلامات جدولة (الاسم الأخير، الاسم الأول، الهاتف).

ما الذي يجب تعديله (النموذج، العرض، أو المتحكم)؟

```
class model_file:
    """ Data Model class"""
    def __init__(self):
        self.directory = []
        try:
            myfile = open("directory.txt")
        except:
            print("Can't open DataFile")
            sys.exit()
        lines = myfile.readlines()
        for line in lines:
            line = line.strip('\n')
            fields = line.split('\t')
            person ={}
            person["familyname"] = fields[0]
            person["lastname"] = fields[1]
            person["tel"] = fields[2]
            self.directory.append(person)
        myfile.close()
    def __del__(self,):
        """ Destructor of the Model class,
        It is called at the end of the program
        to reload the data in the file """
        try:
            myfile = open("directory.txt", "w")
        except:
            print("Can't open DataFile")
            sys.exit()
        for person in self.directory:
            line = "\t".join([person['familyname'],
                              person['firstname'],
                              person['tel'],
                              ])
            line += "\n"
            myfile.write(line)
        myfile.close()
```

The following methods are identical to the methods of the “model” class.

الدوال الآتية مماثلة لدوال صنف "model".

```
def search(self, name):
def get_all(self):
def add(self, name, firstname, tel):
```

2 Modify the ”main” program to use the store-to-file model.

عدّل الدالة "main" لاستخدام نموذج البيانات الذي يحفظ في ملف.

3 What part should we modify if we want to store our data in a database.

ما الذي علينا تعديله إن أردنا تخزين بياناتنا في قاعدة البيانات.

4 What part should we modify if we want to create a graphical interface for our program, or a web interface?

ما الذي علينا تعديله إن أردنا إنشاء واجهة رسومية لبرنامجنا أو واجهة ويب؟

5 What are the advantages of this design method?

ما مزايا هذه الطريقة في التصميم؟

2.6.6 Part 5: MVC + graphical interface

```
from tkinter import Tk, Label, Button, Entry, IntVar, END
from tkinter import  W, E, Text, Listbox, OptionMenu
from tkinter import StringVar
from model_file import model_file
class view_tk:
    def __init__(self, model):
        master = Tk()
        self.master = master
        self.model = model
        master.title("Directory")
        self.entered_number = 0
        #~ master.withdraw()
        master.clipboard_clear()
        master.clipboard_append('i can has clipboardz?')

        self.output_label = Label(master, text="Output")

        self.label = Label(master, text="Input:")
        self.entry = Text(master, height=1, width=50)
        self.label_firstname = Label(master, text="First Name:")
        self.entry_firstname = Text(master, height=1, width=50)
        self.label_tel = Label(master, text="Tel:")
        self.entry_tel = Text(master, height=1, width=50)
        self.output = Text(master, height=10, width=50)
        self.search_button = Button(master, text="Search",
                                    command=lambda: self.update("search"))
        self.add_button = Button(master, text="Add",
                                 command=lambda: self.update("add"))
        self.all_button = Button(master, text="Get All",
                                 command=lambda: self.update("get_all"))

        # LAYOUT
        self.label.grid(row=0, column=0, sticky=W)
        self.label_firstname.grid(row=0, column=1, sticky=W)
        self.label_tel.grid(row=0, column=2, sticky=W)
        #input familyname
        self.entry.grid(row=1, column=0, columnspan=10, sticky=W+E)
        # firstname et tel
        self.entry_firstname.grid(row=1, column=1, columnspan=10, sticky=W+E)
        self.entry_tel.grid(row=1, column=2, columnspan=2, sticky=W+E)


        # command
```

```
        self.search_button.grid(row=2, column=0)
        self.add_button.grid(row=2, column=1)
        self.all_button.grid(row=2, column=2)

        # Output
        self.output_label.grid(row=3, column=0, sticky=E)

        # output
        self.output.grid(row=4, column=0, columnspan=5, sticky=W+E)

    def update(self, method):
        familyname = self.entry.get("1.0",END).strip()
        firstname = self.entry_tel.get("1.0",END)
        tel = self.entry_tel.get("1.0",END)
        # clean ends of words
        familyname = familyname.strip()
        firstname = firstname.strip()
        tel = tel.strip()
        if method == "search":
            result = self.search(familyname)
        elif method == "add":
            result = self.add(familyname, firstname, tel)
        elif method == "get_all":
            result = self.get_all()
        else: # reset
            print("method", method)
            result = familyname
        self.output.insert(END, str(result))

    def add(self, familyname, firstname, tel):
        return self.model.ajouter(familyname, firstname, tel)
    def search(self, familyname):
        result = self.model.rechercher(familyname)
        if result:
            return result
        else:
            return "'%s' not found"%familyname
    def get_all(self):
        return self.model.get_all()

    def run(self,):
        self.master.mainloop()


def controller3():
    # This program allows you to search the number
    # in a table by its name, and display
    # and read data from the keyboard
    data_model =  model_file()
    my_viewer = view_tk(data_model)
```

```
    my_viewer.run() # launch GUI
```

2.6.7 Homework واجب منزلي

Redo the student mentoring project with the MVC architecture.

- **Language** : to choose : Python/Java/PHP.
- **Data models** : XML file with Access using the DOM API.
- **View** : to choose : Console/ graphical interface/ web.
- **Controller** : Student monitoring operations.

أعد مشروع متابعة الطلاب باستخدام بنية MVC.

- لغة البرمجة: اختر من بين Python/Java/PHP
- نموذج البيانات: ملف xml وقراءته بواسطة API DOM
- واجهة: على الخيار: سطر أوامر/ واجهة رسومية/ وب
- متحكّم: عمليات متابعة الطلبة.

2.6.8 Addtional questions (optional) أسئلة إضافية (اختيارية)

Use the Flask web framework to manipulate and display the Mus'haf based on the Quran file Quran-juza-amma.xml جزء عمّ

استعمل إطار العمل Flask لعرض المصحف مبنيا على مستند القرآن الكريم بصيغة XML

2.6.9 References مراجع

- Tutorial points, MVC Framework - Introduction [1] (point, 2022).
- وجدي عصام، تصميم البرمجيات باستخدام نمط Model-View-Controller [2](Issam, 2013).
- المبرمج العربي، نمط التصميم MVC [3](المبرمج العربي, 2022)
- Applications orientées données (Cours CNAM NSY 135) [4] (Philippe Rigaux, 2021).

2.6.10 Additional exercises تمارين للتعمق

- Discover design patterns similar to MVC: MVP, MVVM.
- List the best frameworks that use the MVC architecture.
- Use Flask or Django to create a web interface for the student mentoring project.

- اكتشف النماذج المشابهة لMVC : MVP, MVVM
- اذكر أفضل أطر العمل التي تستخدم معمارية MVC.
- استعمل Flask أو Django لإنجاز واجهة وب لمشروع متابعة الطلبة.

[1] https://www.tutorialspoint.com/mvc_framework/

[2] https://informatic-ar.com/mvc_java/

[3] https://arabicprogrammer.com/article/6434275856/

[4] http://orm.bdpedia.fr/mvc.html

2.7 Practical works n°6 العمل التطبيقي رقم 6

- **Chapter 3**: Design patterns الفصل 3
- **Objectives** أهداف: Design patterns (Factory method, Abstract Factory Method)
- **Data** بيانات:
- **Tools** أدوات : Python

2.7.1 Part TP

2.7.2 Part 1

2.7.2.1 Without Design patterns

1 Run the following program: نفّذ البرنامج الآتي:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#http://python-3-patterns-idioms-test.readthedocs.io/en/latest/Factory.html
# A simple static factory method.
import sys
class Circle(Shape):
    def draw(self): print("Circle.draw")
    def erase(self): print("Circle.erase")

class Square(Shape):
    def draw(self): print("Square.draw")
    def erase(self): print("Square.erase")
if __name__ == "__main__":
  for type in ("Circle", "Square" ,"Circle", "Square"):
    if type == "Circle":
        shape = Circle()
    elif type == "Square":
        shape = Square()
    else:
        print "Bad shape creation: " + type
        sys.exit()
    shape.draw()
    shape.erase()
```

2 Edit the code to add a triangle and rectangle shape. عدّل الكود لإضافة شكل مثلث ومستطيل.

2.7.2.2 Simple Factory method

1 We want to encapsulate the object instantiation operation, using a ShapeFactory class
نريد تغليف عملية إنشاء الكائن باستخدام صنف ShapeFactory

```
class ShapeFactory:
    @staticmethod
    def createShape(type):
```

```
        if   type == "Circle":  return Circle()
        elif type == "Square":  return Square()
        else:
            print "Bad shape creation: " + type
            sys.exit()
if __name__ == "__main__":
  for type in ("Circle", "Square" ,"Circle", "Square"):
    shape = ShapeFactory.createShape(type)
    shape.draw()
    shape.erase()
```

2 Edit the code to add a triangle and rectangle shape. عدّل الكود لإضافة شكل مثلث ومستطيل.

2.7.2.3 Factory method

We want to create a specialized Factory (ShapeFactory_SCT) which only produces squares, circles, and triangles. Another Factory (ShapeFactory_SCR) which only produces squares and circles and rectangles.
نريد إنشاء مصنع متخصص (ShapeFactory_SCT) ينتج المربعات والدوائر والمثلثات فقط. مصنع آخر (ShapeFactory_SCR) لا ينتج إلا المربعات والدوائر والمستطيلات.

Application : We want a stock management application, our client requires having three types of accounts:

- a manager account, to manage the stock, make purchases and orders, see the total incomes.
- a seller account, to allow sellers to sell, make purchases and orders.
- A visitor account to allow customers to view stock, make purchases and place orders.

1 Propose a design pattern, taking inspiration from the class diagram.

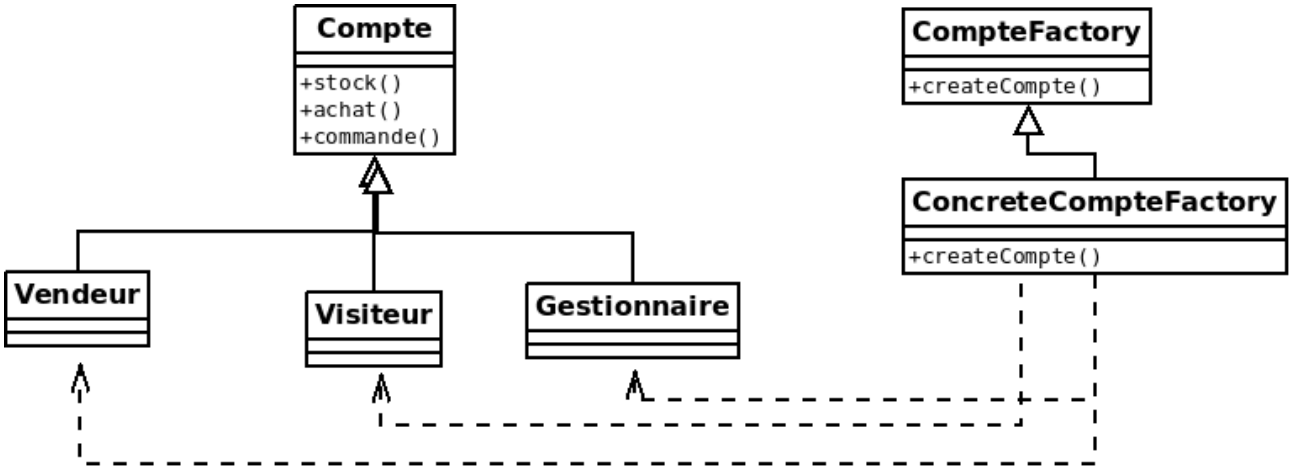
2 Implement base classes and creation class.

نريد تطبيقًا لإدارة المخزون، ويطلب الزبون أن يكون فيها ثلاثة أنواع من الحسابات:

- حساب مدير، لإدارة المخزون وإجراء عمليات الشراء والطلبات، راجع الحصيلة.
- حساب بائع، للسماح للبائعين بالبيع وإجراء عمليات الشراء والطلبات.
- حساب زائر للسماح للعملاء بعرض المخزون وإجراء عمليات الشراء وتقديم الطلبات.

1 اقترح نمط تصميم مستوحى من الرسم التخطيطي للصنف.

2 برمج الأصناف الأساسية وصنف الإنشاء.



2.7.3 Part 2 Abstract Factory

We return to the classes of the first part.
We want to Create 2D shapes and 3D shapes, according to the following diagram.
نعود إلى اصناف القسم الأول، نرغب في إنشاء أشكال ذات بُعدين، وأشكال ذات ثلاثة أبعاد 3D، حسب المخطط الموالي:

1. create the subclasses square2D, Square3D, circle2D, circle3D.
2. Create the abstract Factory, ShapeFactory
3. Create concrete factories ShapeFactory2D, and ShapeFactory3D
4. The following code just illustrates examples.

1. أنشئ الأصناف الفرعية square2D, Square3D, circle2D، circle3D
2. أنشئ المصنع الافتراضي (abstract ShapeFactory Factory)
3. أنشئ المصنعين الواقعيين ShapeFactory2D و ShapeFactory3D
4. يعرض الكود الموالي بعض الأمثلة فقط.

```
class ShapeFactory:
    @staticmethod
    def createShape(type):
        pass

class ShapeFactory2D(ShapeFactory):
    @staticmethod
    def createShape(type):
        if   type == "Circle":  return Circle2D()
        elif type == "Square":  return Square2D()
        else:  return None

#Exemple
class Circle2D(Circle):
    def draw(self) : print("Circle.draw2D")
    def erase(self): print("Circle.erase2D")


# 2D objects creation
for type in ("Circle", "Square" ,"Circle", "Square"):
    shape = ShapeFactory2D.createShape(type)
    shape.draw()
    shape.erase()
```

2.7.4 Part 3 Singleton

Test the following code

جرّب الكود الآتي

```
class  Singleton (object):
    instance = None   # static attribute of the class
    def __new__(cls):
        """standard Python construction method"""
        if cls.instance is None:
            cls.instance = object.__new__(cls)
        return cls.instance
# Use
monSingleton1 =  Singleton()
monSingleton2 =  Singleton()
# monSingleton1 and monSingleton2 return the same instance
assert monSingleton1 is monSingleton2
print(monSingleton1, monSingleton2)
```

1. Propose a class that allows you to open a file for writing, It is necessary to ensure that access to the file is unique
2. Propose a class that allows you to connect to a database then close it, It is necessary to ensure that access to the database is unique
3. Improve the solution by allowing simultaneous access to the file/database for a limited number

1. اقترح صنفًا يتيح لك فتح ملف للكتابة، على أن يكون الوصول إلى الملف مقيّدًا لوصول واحد في نفس الوقت.
2. اقترح صنفًا يسمح الاتصال بقاعدة بيانات ثم إغلاقها، يجب التأكد من أن الوصول إلى قاعدة البيانات وحيد في نفس الوقت.
3. تحسين الحل للسماح بالوصول المتزامن إلى الملف /قاعدة البيانات لعدد محدود

2.7.5 Part 4 Builder Pattern

Consider the story class that simulates Facebook, Instagram and Youtube stories.
A Story is composed of photo, video, special effects, decorations, music, but it does not contain all these components at the same time. So we can have several configurations.
نفترض صنف القصة التي تحاكي قصص Facebook وInstagram وYoutube.
القصة فيها صور ومقاطع فيديو ومؤثرات خاصة وزخارف وموسيقى، وقد لا تحوي كلها هذه المكونات في نفس الوقت. لذلك لدينا العديد من الإعدادات.

```
class Story:
    """
    Represent the complex object under construction.
    """
    def __init__(self,):
        self.parts = []
    def add(self, part):
        self.parts.append(part)
    def show(self,):
            print(self.parts)
```

1 We want to create a class that allows you to create the different components of a story.
نريد إنشاء صنف يسمح لك بإنشاء مكونات مختلفة للقصة.

```
class Builder():
    """
    Specify an abstract interface for creating parts of a Product
    object.
    """
    def __init__(self):
        self.story = Story()

    def _build_image(self):
        pass

    def _build_music(self):
        pass

    def _build_effet(self):
        pass
    def _build_video(self):
        pass
```

2 We want to create a class that allows you to configure the creation of stories by
نريد إنشاء صنف يسمح لك بإعداد إنشاء القصص كما يلي:

- story basic (just a picture) — قصة بسيطة (صورة فقط)
- story with music — قصة مع موسيقى
- story video with special effects — قصة بها فيديو و مؤثرات خاصة

```
class StoryDirector:
    """
```

```
    Construct an object using the Builder interface.
    """
    def __init__(self):
        self._builder = None
    def construct(self, builder):
        self._builder = builder
    def basic_story(self,):
        print("*** create basic story ***")
        self._builder._build_image()
    def photo_music_story(self,):
        print("*** create photo + music story ***")
        self._builder._build_image()
        self._builder._build_music()
        self._builder._build_effet()
    def video_story(self,):
        print("*** create video story ***")
        self._builder._build_video()
        self._builder._build_effet()
```

3 We want to create a class that allows to create special components for FaceBook and Instagram

نريد إنشاء صنف يسمح لك بإنشاء مكونات خاصة لـ FaceBook وInstagram

```
class FacebookConcreteBuilder(Builder):
    """
    Construct and assemble parts of the product by implementing the
    Builder interface.
    Define and keep track of the representation it creates.
    Provide an interface for retrieving the product.
    """
    def _build_image(self):
        self.story.add("image facebook")
    def _build_music(self):
        self.story.add("music facebook")
    def _build_effet(self):
        self.story.add("effect facebook")
    def _build_video(self):
        self.story.add("video facebook")

class InstagramConcreteBuilder(Builder):
    """
    Construct and assemble parts of the product by implementing the
    Builder interface.
    Define and keep track of the representation it creates.
    Provide an interface for retrieving the product.
    """
    def _build_image(self):
        self.story.add("image Instagram")
    def _build_music(self):
        self.story.add("music Instagram")
    def _build_effet(self):
        self.story.add("effect Instagram")
    def _build_video(self):
```

```
        self.story.add("video Instagram")
```

4 Test main function call

```
def main():
    concrete_builder = FacebookConcreteBuilder()
    director = StoryDirector()
    director.construct(concrete_builder)
    director.basic_story()
    story = concrete_builder.story
    story.show()
    concrete_builder2 = InstagramConcreteBuilder()
    director = StoryDirector()
    director.construct(concrete_builder2)
    director.video_story()
    story = concrete_builder2.story
    story.show()


if __name__ == "__main__":
    main()
```

2.7.5.1 Questions

1 Given the Sandwich object which consists of the varieties of ingredients, we want to give names to the different configurations (choices) of the sandwiches in order to facilitate the task of creating the sandwiches:

نفرض الكائن شطيرة (سندويش) الذي فيه مكونات عديدة، نريد تسمية الإعدادات (الاختيارات) المختلفة للشطائر لتسهيل مهمة إنشاء الشطائر:

- Shawarma Sandwich — شطيرة شاروما
- Ground meat — شطيرة لحم مرحي
- Fried with omelet. — بطاطا مقلية بالبيض
- Etc.

Implement product creation using Builder.

برمج إنشاء المنتجات باستخدام نمط البنّاء Builder.

2 A new car is sold in several options (the basic, the whole, with or without some options. Implement the creation of car objects according to orders.

تُباع السيارات الجديدة بعدة خيارات (الأساسية، الكاملة، مع بعض الخيارات أو دونها).

برمج إنشاء كائنات السيارة حسب الطلب.

2.7.6 Home work

واجب منزلي

لا واجب منزلي هذه المرة.

2.7.7 References

مراجع

- أنيس حكمت، دورة Pattern Design باللغة العربية[5](Hikmat, 2021).

2.7.8 Additional exercises

تمارين للتعمق

يُتبع...

[5] https://2nees.com/courses-and-articles/design-pattern

2.8 Practical works n°7 العمل التطبيقي رقم 7

- **Chapter 3**: Design patterns الفصل 3
- **Objectives** أهداف: Structural Design patterns
- **Data** بيانات:
- **Tools** أدوات : Python

2.8.1 Part TP

2.8.2 Adapter المحوّل

Run the following program نفّذ البرنامج الآتي

```
class Person:
    def __init__(self, nom, prenom):
        self.nom = nom
        self.prenom = prenom

class personne:
    def __init__ (self, nomcomplet):
        self.nom = nomcomplet

class adapter(personne):
    def __init__(self, person):
        self.nom    = person.nom +"-"+ person.prenom

def main():
    # personne
    personOne = Person("Akli", "Mohand")
    per = adapter(personOne)
    print(per.nom)
    return 0

if __name__ == '__main__':
    main()
```

Questions :

1 Analyze the pattern and cite its advantages.

حلل النمط واذكر مزاياه.

2 The date is stored on the machine in YYYY/MM/DD format, we want to create adapters that display dates in several styles (MM/DD/YYYY), (DD/MM/YYYY).

التاريخ مخزّن على الجهاز بتنسيق YYYY/MM/DD، ونريد إنشاء محولات تعرض التواريخ في عدة أنماط (MM/DD/YYYY)، (DD/MM/YYYY).

2.8.3 Facade الواجهة

```
class Facade:
    """
    Know which subsystem classes are responsible for a request.
    Delegate client requests to appropriate subsystem objects.
    """

    def __init__(self):
        self._subsystem_1 = Subsystem1()
        self._subsystem_2 = Subsystem2()
        self._subsystem_3 = Subsystem3()

    def operation_retait_ccp(self):
        self._subsystem_1.operation1()
    def operation_virment_ccp(self):
        self._subsystem_1.operation2()
    def operation_fact_eau(self):
        self._subsystem_2.operation1()
    def operation_fact_elec_gaz(self):
        self._subsystem_3.operation1()


class Subsystem1:
    """
    Implement subsystem functionality.
    Handle work assigned by the Facade object.
    Have no knowledge of the facade; that is, they keep no references to
    it.
    """

    def operation1(self):
        print("operation retrait ccp")

    def operation2(self):
        print("operation vierment ccp")


class Subsystem2:
    """
    Implement subsystem functionality.
    Handle work assigned by the Facade object.
    Have no knowledge of the facade; that is, they keep no references to
    it.
    """

    def operation1(self):
        print("operation payment facture  de l'eau ADE")
class Subsystem3:
    """
    Implement subsystem functionality.
```

```
    Handle work assigned by the Facade object.
    Have no knowledge of the facade; that is, they keep no references to
    it.
    """

    def operation1(self):
        print("operation payment facture  de Eletcricité/gaz Sonelgaz")


def main():
    facade = Facade()
    facade.operation_retait_ccp()
    facade.operation_virment_ccp()
    facade.operation_fact_eau()
    facade.operation_fact_elec_gaz()


if __name__ == "__main__":
    main()
```

Questions

1 There are several information resources for computer science students; لطلبة الإعلام الآلي مصادر عديدة للمعلومات

- Site de département — موقع قسم الإعلام الآلي
- e-learning-Info Plateform — منصة التعليم الإلكتروني للقسم
- University elearning platform — منصة التعليم الإلكتروني للجامعة
- Department, Faculty and university fFaceBook pages — صفحات فيسبوك للقسم والكلية والجامعة
- Student groups — مجموعات الطلبة

Propose a facade to group displays in one place. اقترح واجهة لتجميع المعلومات في مكان واحد.

2 Algérie Poste uses two strategies: تستعمل مؤسسة بريد الجزائر استراتيجيتين هما:

- Unique counter for all operations — الشبّاك الموحّد لكل العمليات
- Specialized counter for each operation — شبابيك متخصصة لكل عملية
- Criticize the two strategies — انتقد كلتا الاستراتيجيتين

2.8.4 Proxy الوكيل

```python
class RealSubject:
    """
    The RealSubject contains some core business logic. Usually, RealSubjects are
    capable of doing some useful work which may also be very slow or sensitive -
    e.g. correcting input data. A Proxy can solve these issues without any
    changes to the RealSubject's code.
    """

    def request(self):
        print("RealSubject: Handling request.")


class Proxy(RealSubject):
    """
    The Proxy has an interface identical to the RealSubject.
    """

    def __init__(self, RealSubject):
        self._real_subject = real_subject

    def request(self):
        """
        The most common applications of the Proxy pattern are lazy loading,
        caching, controlling the access, logging, etc. A Proxy can perform one
        of these things and then, depending on the result, pass the execution to
        the same method in a linked RealSubject object.
        """

        if self.check_access():
            self._real_subject.request()
            self.log_access()

    def check_access(self):
        print("Proxy: Checking access prior to firing a real request.")
        return True

    def log_access(self):
        print("Proxy: Logging the time of request.", end="")


def client_code(subject: RealSubject):
    """
    The client code is supposed to work with all objects (both subjects and
    proxies) via the Subject interface in order to support both real subjects
    and proxies. In real life, however, clients mostly work with their real
    subjects directly. In this case, to implement the pattern more easily, you
    can extend your proxy from the real subject's class.
    """
```

```
    # ...

    subject.request()

    # ...


if __name__ == "__main__":
    print("Client: Executing the client code with a real subject:")
    real_subject = RealSubject()
    client_code(real_subject)

    print("")

    print("Client: Executing the same client code with a proxy:")
    proxy = Proxy(real_subject)
    client_code(proxy)
```

Questions

1 Consider an access service to a sports hall or a stadium.
Create the proxy that allows you to check the health pass (covid19) before entering the place.
نريد إنشاء خدمة مراقبة الدخول إلى قاعة رياضية أو ملعب.
أنشئ صنف الوكيل proxy للتحقق من الجواز الصحي لكورونا قبل دخول المكان.

2 To submit the visa file, embassies make contracts with specialized companies.
Realize the proxy diagram for the visa application filing operation.
لمعالجة ملفات التأشيرات تتعاقد السفارات مع شركات متخصصة.
أنشئ مخطط الوكيل لعملية طلب التأشيرة.

2.8.5 Behavioral design patterns: Strategy

```
#Define a family of algorithms, encapsulate each one, and make them
#interchangeable. Strategy lets the algorithm vary independently from
#clients that use it.

class Context:
    """
    Define the interface of interest to clients.
    Maintain a reference to a Strategy object.
    """
    def __init__(self, strategy):
        self._strategy = strategy

    def context_interface(self):
        self._strategy.algorithm_interface()

class Strategy():
    """
    Declare an interface common to all supported algorithms. Context
    uses this interface to call the algorithm defined by a
    ConcreteStrategy.
    """
    def algorithm_interface(self):
        pass

class ConcreteStrategyA(Strategy):
    """
    Implement the algorithm using the Strategy interface.
    """
    def algorithm_interface(self):
        pass

class ConcreteStrategyB(Strategy):
    """
    Implement the algorithm using the Strategy interface.
    """

    def algorithm_interface(self):
        pass

def main():
    concrete_strategy_a = ConcreteStrategyA()
    context = Context(concrete_strategy_a)
    context.context_interface()

if __name__ == "__main__":
    main()
```

Questions

1 We want to implement the transition between the years of study in bachelor's and master's

degrees (the average and the necessary credits, as well as the required units.)

نريد برمجة طريقة الانتقال بين سنوات الدراسة في الليسانس والماستر (المعدل والأرصدة الضرورية وكذلك الوحدات المطلوبة).

2 We want to implement a bank credit procedure with or without interest.

نريد برمجة القروض البنكية بالربا ودون ربا.

2.8.6 Behavioral Design patterns :Observer

```
"""
Define a one-to-many dependency between objects so that when one object
changes state, all its dependents are notified and updatedautomatically.
"""
class Subject:
    """
    Know its observers. Any number of Observer objects may observe a
    subject.
    Send a notification to its observers when its state changes.
    """
    def __init__(self):
        self._observers = set()
        self._subject_state = None

    def attach(self, observer):
        observer._subject = self
        self._observers.add(observer)

    def detach(self, observer):
        observer._subject = None
        self._observers.discard(observer)

    def _notify(self):
        for observer in self._observers:
            observer.update(self._subject_state)

    def subject_state(self, arg):
        self._subject_state = arg
        self._notify()


class Observer():
    """
    Define an updating interface for objects that should be notified of
    changes in a subject.
    """

    def __init__(self):
        self._subject = None
        self._observer_state = None

    def update(self, arg):
        pass
    def show(self,):
        print(self._observer_state)


class ConcreteObserver(Observer):
    """
    Implement the Observer updating interface to keep its state
```

```
        consistent with the subject's.
        Store state that should stay consistent with the subject's.
        """

        def update(self, arg):
            self._observer_state = arg
            # ...


def main():
    subject = Subject()
    ahmed_observer = ConcreteObserver()
    salem_observer = ConcreteObserver()
    subject.attach(ahmed_observer)
    subject.attach(salem_observer)

    subject.subject_state(123)

    ahmed_observer.show()
    salem_observer.show()

    subject.subject_state(99)

    ahmed_observer.show()
    salem_observer.show()



if __name__ == "__main__":
    main()
```

Questions

1. We want to implement a notification class from the department site, where the student can register to receive notifications of new posts.
نريد تنفيذ فصل إعلامي من موقع القسم، حيث يمكن للطالب التسجيل لتلقي إشعارات المنشورات الجديدة.

2. We want to implement a calendar where users can sign up to receive reminders.
نريد برمجة رزنامة (تقويم) يشترك فيه المستخدمون لتلقي التنبيهات.

2.8.7 Homework واجب منزلي

Choose 3 design patterns (one per category), then implement the questions for each pattern.
اختر ثلاثة أنماط تصميم (واحد من كل صنف) ثم أنجز أسئلة كل نمط.

2.8.8 Addtional questions (optional) أسئلة إضافية (اختيارية)

For the management of a tourist agency, we want to develop:

- A facade for outsourced services: transportation, accommodation and catering.
- Multiple strategies for each service, to satisfy different customer budgets.
- Adapters for paying in dinars for services abroad.
- An access control to resources to avoid overlapping booking (use singleton patterns).
- Possibility of canceling booking (use the memento pattern)

- واجهة Facade للخدمات المتعاقد عليها: النقل والإقامة والتموين.
- استراتيجيات متعددة لكل خدمة، لتلبية ميزانيات الزبائن المختلفة.
- محولات Adapter للدفع بالدينار مقابل الخدمات في الخارج.
- التحكم في الوصول إلى الموارد لتجنب الحجوزات المتداخلة (استخدم نمط الفريد Singleton).
- إمكانية إلغاء الحجوزات (استخدم نمط memento)

Questions :

- Draw class diagrams. ارسم مخططات الأصناف
- Explain patterns use. اشرح استعمال أنماط التصميم
- Simulate class implementation. حاكي برمجة الأصناف

2.8.9 References مراجع

- أنيس حكمت، دورة Pattern Design باللغة العربية[6](Hikmat, 2021).

2.8.10 Additional exercises تمارين للتعمق

يُتبع...

[6] https://2nees.com/courses-and-articles/design-pattern

2.9 Practical works n°8 العمل التطبيقي رقم 8

- **Chapter 3**: Design patterns الفصل 3
- **Objectives** أهداف: Reflectionالانعكاسية
- **Data** بيانات:
- **Tools** أدوات : Python

2.9.1 Part TP

2.9.2 Fontion type()

Run the following scripts and note the results نفّذ البرامج الآتية وسجّل النتائج

```
a = 12
b = 13.57
c = True
d = 12 + 3j

print(type(a))
print(type(b))
print(type(c))
print(type(d))
```

```
mystr = 'Salam Alykom!'
mylist = [1, 2, 3, 4, 5, 6, 7]
mytuple = (80, 'TD', 15, 'Cours')

print(type(mystr))
print(type(mylist))
print(type(mytuple))
```

```
my_tuple = (10, 'Hello', 45, 'Hi')
my_dict = {1: 'One', 2: 'Two', 3: 'Three'}

if type(my_tuple) is not type(my_dict):
      print("Both variables have different object types.")
else:
      print("Same Object type")
```

The type() function allows you to create a type dynamically, execute the script.
Then explore the documentation for the type() and vars() functions.

تتيح لك دالة type() إنشاء نوع ديناميكيًا.
نفّذ البرنامج ثم استكشف الوثائق الخاصة بالدوال type() وvars()

```
a = type('Test', (object, ), dict(x = 'Hello', y = 10, z = 'World'))

print(type(a))
```

```
print(vars(a))
```

Exemple d'application : calculer en contrôlant le type des arguments

Application example : calculate by checking the type of the arguments

مثال تطبيقي: الحساب مع التحكّم في نوع الوسائط

```
def calculate(x, y, op='sum'):
    if not(isinstance(x, int) and isinstance(y, int)):
        print(f'Invalid Types of Arguments - x:{type(x)}, y:{type(y)}')
        raise TypeError('Incompatible types of arguments, must be integers')
    if op == 'diff':
        return x - y
    if op == 'mult':
        return x * y
    # default is sum
    return x + y
```

Improve the calculate function to allow calculations with integers and real numbers.

حسّن الدالة للحساب بالأعداد الصحيحة والأعداد الحقيقية.

2.9.3 isinstance() function

Test if variables are instances of standard types.

تحقق من أنّ المتغيرات تنتمي إلى الأنواع الأساسية.

```
int_1 = 7
str_1 = "Learn Python"
list_3 = [2, 4, 6]

print ("Is int_1 an integer?" + str (isinstance (int_1, int)))
print ("Is int_1 a string?" + str (isinstance (int_1, str)))
print ("Is str_1 a string?" + str (isinstance (str_1, str)))
print ("Is list_1 an integer?" + str (isinstance (list_1, int)))
print ("Is list_1 a list?" + str (isinstance (list_1, list)))

# test if its types is in a list
print ("Is int_1 integer or list or string?" + str (isinstance (int_1, (list, str, int))))
print ("Is list_1 string or tuple?" + str (isinstance (list_1, (str, tuple))))
```

Test if a variable is an instance of a class

تحقق أن المتغير ينتمي إلى صنف معين

```
class Test1:
      a = 6

testInstance = Test1 ()

print (isinstance (testInstance, Test1))
print (isinstance (testInstance, (list, tuple)))
print (isinstance (testInstance, (list, tuple, Test1)))
```

2.9.4 vars() function

```
class Student:
    def __init__(self, name = 'Leo', age = 22, course = 'MBA', city = 'Mumbai'):
        self.Name = name
        self.Age = age
        self.Course = course
        self.City = city

obj = Student()
print('Dictionary output is:', vars(obj))

obj2 =obj = Student("Samir", 22, "MBB", "Bouira")
print('Dictionary output is:', vars(obj2))
```

2.9.5 dir() function

Show methods and attributes of a module or class اعرض دوال وخصائص الوحدة أو الصنف

```
import random
print("The random library's contents are as follows::")
print(dir(random))
```

Show attributes/methods of a variable of given type اعرض سمات ودوال متغير من نوع معيّن

```
number = [1, 2, 3]
print(dir(number))
```

Re-test the previous script with a variable of type list, dict, float, and without parameters.
أعد تجربة البرنامج السابق باستخدام متغير من نوع list، dict، float وبدون وسائط.

Show methods and attributes of a class اعرض دوال وخصائص الصنف

```
class Student():
     def __init__(self,x):
          return self.x
# Calling function
att = dir(Student)
print(att)
```

You can customize the dir() method for a given class. يمكن تخصيص الدالة dir() لصنف معيّن

```
class Student():
     def __init__(self,x):
          return self.x
     def __dir__(self):
        return [10,20,30]
s = Student()
att = dir(s)
print(att)
```

2.9.6 Extracting Python class details استخراج تفاصيل صنف بايثون

Analyze the following script execution result: حلّل نتيجة تنفيذ البرنامج التالي:

```
class Data:
    """Data Class"""
    d_id = 10


class SubData(Data):
    """SubData Class"""
    sd_id = 20

print(Data.__class__)
print(Data.__bases__)
print(Data.__dict__)
print(Data.__doc__)

print(SubData.__class__)
print(SubData.__bases__)
print(SubData.__dict__)
print(SubData.__doc__)
```

2.9.7 Dynamic modification of attributes التعديل الديناميكي للخصائص

```
class A:
    pass
# Dynamic adding of  attribute to class A
A.x = 1
a = A()

# Dynamic adding of  attribute to object a
a.y = 2

print("Attributes of class A: ",vars(A))
print("Attributes of instance a: ",vars(a))
```

Since methods are just a special type of attribute, this means we can also add methods at runtime. Let's modify our class by dynamically adding a ___init___ method.

نظرًا لأن الدوال هي مجرد نوع خاص من الخصائص، لذا يمكننا أيضًا إضافة دوال أثناء التنفيذ. فلنعدّل الصنف بإضافة دالة __init__ إليه ديناميكيًا.

```
def init(self):    # the function and argument can have any name
    self.x = 1
class A:
    pass
setattr(A, '__init__', init)
a = A()
print(a.x)
```

We can take this concept a step further by modifying the ___code___ attribute of a function. This time by simple assignment:

يمكننا أن نتوسّع في هذا المفهوم، ونعدّل خاصية __code__ للدالة. هذه المرة نغيّر إسنادًا بسيطًا:

```
def test():
    return "Test old code"
print(test())
# ~ test.__code__ = (lambda x : print("Hello", x)).__code__
test.__code__ = (lambda: "Test new code").__code__
print(test())
```

2.9.8 ___call___() Method

Python has a set of built-in methods and ___call___ is one of them. The ___call___ method allows Python programmers to write classes where instances behave like functions and can be called like a function. When the instance is called as a function; if this method is defined, x(arg1, arg2, ...) is a shortcut for x.___call___(arg1, arg2, ...).

لدى بايثون مجموعة من الدوال القياسية و__call__واحدة منها. تسمح الدالة __call__ لمبرمجي Python بكتابة أصناف تتصرف فيها الكائنات مثل الدوال ويمكن استدعاؤها كدالة، عندما يستدعي الفرد كدالة؛ إذا تم تعريف هذه الدالة، فإن x(arg1, arg2, ...) هو اختصار لـ x.__call__(arg1, arg2, ...).

```
class Product:
    def __init__(self):
        print("Instance Created")

    # Defining __call__ method
    def __call__(self, a, b):
        print(a * b)

# Instance created
ans = Product()

# __call__ method will be called
ans(10, 20)
```

The callable() function : tests whether a class or variable can be called as a function

تختبر الدالة callable(): ما إذا كان من الممكن استدعاء صنف أو متغير كدالة

```
print("Is str callable? ", callable(str)) # str class
print("Is len callable? ", callable(len)) # len function
print("Is list callable? ", callable(list)) # list class

num=10
print("Is variable callable? ", callable(num))
```

Test the following script: جرّب البرنامج الآتي:

```
class student:
    def greet(self):
        print("Hello there")

std = student()
print("Is student class callable? ",callable(student))
```

```
print("Is student.greet() callable? ",callable(std.greet))
print("Is student instance callable? ",callable(std))
```

Test the following script: جرّب البرنامج الآتي:

```
class student:
    def greet(self):
        print("Hello there")

    def __call__(self):
        print("Hello, I am a student.")

std = student()
print("Is student instance callable? ", callable(std))
print(std())
```

2.9.9 Application تطبيق

We want to apply these notions to an example of a class for the management of complex numbers.
نريد تطبيق هذه المفاهيم على مثال لصنف التعامل مع الأعداد المركبة.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
class ComplexIt(object):
    def __init__(self, real, imag=0.0):
        self.real = real
        self.imag = imag

    def __add__(self, other):
        return ComplexIt(self.real + other.real,
                         self.imag + other.imag)

    def __sub__(self, other):
        return ComplexIt(self.real - other.real,
                         self.imag - other.imag)

    def __mul__(self, other):
        return ComplexIt(self.real*other.real - self.imag*other.imag,
                         self.imag*other.real + self.real*other.imag)

    def __truediv__(self, other):
        sr, si, orl, oi = self.real, self.imag, other.real, other.imag # short forms
        r = float(orl**2 + oi**2)
        return ComplexIt((sr*orl+si*oi)/r, (si*orl-sr*oi)/r)

    def __abs__(self):
        return sqrt(self.real**2 + self.imag**2)

    def __neg__(self):   # defines -c (c is ComplexIt)
        return ComplexIt(-self.real, -self.imag)

    def __eq__(self, other):
```

```
        return self.real == other.real and self.imag == other.imag

    def __ne__(self, other):
        return not self.__eq__(other)

    def __str__(self):
        return '(%g, %g)' % (self.real, self.imag)

    def __repr__(self):
        return 'ComplexIt' + str(self)

    def __pow__(self, power):
        raise NotImplementedError ('self**power is not yet impl. for ComplexIt')
    def _illegal(self, op):
        print('illegal operation "%s" for ComplexIt numbers' % op)
    def __gt__(self, other):  self._illegal('>')
    def __ge__(self, other):  self._illegal('>=')
    def __lt__(self, other):  self._illegal('<')
    def __le__(self, other):  self._illegal('<=')



if __name__ == '__main__':

    a = ComplexIt(1,5)
    b = ComplexIt(3,2)

    # usual operations
    print(a)
    print(b)
    print(a+b)
    print(a-b)
    print(a*b)
    print(a/b)
    print(a<=b)

    # addition with integer
    x=  a + 4.5
    y=  a + ComplexIt(4.5, 0)
    print(x, y, x==y)
```

2.9.10 Homework

واجب منزلي

لا واجب منزلي

2.9.11 References

مراجع

- Reflection in Python https://www.geeksforgeeks.org/reflection-in-python/
- Python Reflection and Introspection https://betterprogramming.pub/python-reflection-and-introspe

- Python Type() https://www.toppr.com/guides/python-guide/references/methods-and-functions/methods/built-in/type/python-type/
- Python type() Function [With Easy Examples] https://www.digitalocean.com/community/tutorials/python-type
- Hans Petter Langtangen, Introduction to classes, 2016 http://hplgit.github.io/primer.html/doc/pub/class/
- Hans Petter Langtangen, A Primer on Scientific Programming with Python, Springer, 2014 (Langtangen, 2014)

2.10 Practical works n°9 العمل التطبيقي رقم 9

- **Chapter 3**: Meta programming: Aspect oriented Programming الفصل 3
- **Objectives** أهداف: Module Creation, Aspect Oriented Programming
- **Data** بيانات:
- **Tools** أدوات : Python, unittest, aspectlib

2.10.1 Module Creation إنشاء وحدة برمجية

Returning to the example of the ComplexIt class, we want to create a complex number manipulation module.
Save the script as "complexit.py", then run the script.
نعود إلى مثال صنف الأعداد المركبة ComplexIt، نريد إنشاء وحدة التعامل مع الأعداد المركبة.
احفظ البرنامج باسم "complexit.py"، ثم نفّذه.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
class ComplexIt(object):
    def __init__(self, real, imag=0.0):
        self.real = real
        self.imag = imag

    def __add__(self, other):
        return ComplexIt(self.real + other.real,
                         self.imag + other.imag)

    def __sub__(self, other):
        return ComplexIt(self.real - other.real,
                         self.imag - other.imag)

    def __mul__(self, other):
        return ComplexIt(self.real*other.real - self.imag*other.imag,
                         self.imag*other.real + self.real*other.imag)

    def __truediv__(self, other):
        sr, si, orl, oi = self.real, self.imag, other.real, other.imag # short forms
        r = float(orl**2 + oi**2)
        return ComplexIt((sr*orl+si*oi)/r, (si*orl-sr*oi)/r)

    def __abs__(self):
        return sqrt(self.real**2 + self.imag**2)

    def __neg__(self):   # defines -c (c is ComplexIt)
        return ComplexIt(-self.real, -self.imag)

    def __eq__(self, other):
        return self.real == other.real and self.imag == other.imag

    def __ne__(self, other):
```

```
        return not self.__eq__(other)

    def __str__(self):
        return '(%g, %g)' % (self.real, self.imag)

    def __repr__(self):
        return 'ComplexIt' + str(self)

    def __pow__(self, power):
        raise NotImplementedError ('self**power is not yet impl. for ComplexIt')
    def _illegal(self, op):
        print('illegal operation "%s" for ComplexIt numbers' % op)
    def __gt__(self, other):  self._illegal('>')
    def __ge__(self, other):  self._illegal('>=')
    def __lt__(self, other):  self._illegal('<')
    def __le__(self, other):  self._illegal('<=')



if __name__ == '__main__':

    a = ComplexIt(1,5)
    b = ComplexIt(3,2)

    # usual operations
    print(a)
    print(b)
    print(a+b)
    print(a-b)
    print(a*b)
    print(a/b)
    print(a<=b)

    # addition with integer
    x=  a + 4.5
    y=  a + ComplexIt(4.5, 0)
    print(x, y, x==y)
```

Call ComplexIt class from another file : ”trycomplex.py” استدعاء صنف ComplexIt من ملف آخر

```
from complexit import ComplexIt
# ~ import complexit


def main(args):
    a = ComplexIt(1,5)
    b = ComplexIt(3,2)

    # usual operations
    print(a)
    print(b)
    print(a+b)
```

```
    print(a-b)
    print(a*b)
    print(a/b)
    print(a<=b)

    # addition with integer
    x=  a + 4.5
    y=  a + ComplexIt(4.5, 0)
    print(x, y, x==y)

    return 0

if __name__ == '__main__':
    import sys
    sys.exit(main(sys.argv))
```

Edit the ”complexit.py” file to avoid duplicate output. عدّل الملف "complexit.py" لتجنب تكرار الإخراج.

2.10.2 Test unit creation إنشاء وحدة فحص

Save the script as ”testcomplexit.py”, then run the script. احفظ البرنامج باسم "testcomplexit.py"، ثم نفّذه.

```
import unittest
from complexit import ComplexIt


class ComplexTestCase(unittest.TestCase):
    """Tests for `complexit.py`."""

    def test_addition(self):
        """Test complex addition"""
        a = ComplexIt(1,2)
        b = ComplexIt(2,2)
        c= ComplexIt(3,4)
        self.assertEqual(a+b, c)
        self.assertNotEqual(a+b, 0)
        self.assertTrue(a+b==c)


if __name__ == '__main__':
    unittest.main()
```

Add other methods to test other operations أضف دوالا أخرى لاختبار بقية العمليات.

2.10.3 Decorators Use استعمال المُزخرفات

Run the following script نفذ البرنامج الآتي

```
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_salam():
    print("Salam!")


if __name__ == '__main__':
    say_salam()
```

Run the following script

نفذ البرنامج الآتي

```
# ~ def my_decorator(func):
    # ~ def wrapper(*args, **kwargs):
        # ~ print("Something is happening before the function is called.")
        # ~ func(*args, **kwargs)
        # ~ print("Something is happening after the function is called.")
    # ~ return wrapper

def my_decorator(func):
    def wrapper(*args, **kwargs):
        print("Something is happening before the function is called.")
        print("Function called", func.__name__, " with arguments ", kwargs,args )
        func(*args, **kwargs)
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_salam(name="Ahmed"):
    print("Salam!"+" ya "+name)

@my_decorator
def say_hello(name="Ahmed"):
    print("Hello!"+" ya "+name)


if __name__ == '__main__':
    say_salam(name="Ali")
    say_hello(name="Ali")
```

Use decorators to parse the ___add___ method in the "ComplexIt" class.
Use the functions seen in reflexivity practicals to analyze the ComplexIt module.

استخدم المزخرفات لتحليل الدالة __add__ في صنف "ComplexIt".
استخدم الدوال التي تعلمناها في تطبيق الانعكاسية لتحليل وحدة ComplexIt.

2.10.4 Aspects oriented programming البرمجة جانبية التوجه

Install aspectlib library ثبّت مكتبة aspectlib

```
pip install aspectlib
```

Test the following code on ComplexIt module جرب الكود الموالي على وحدة ComplexIt

```
import aspectlib
import aspectlib.debug
import sys
from complexit import ComplexIt


if __name__ == '__main__':

    with aspectlib.weave(ComplexIt.__add__,
        aspectlib.debug.log(print_to=sys.stdout, stacktrace=None,),
       lazy=True,):
        a = ComplexIt(1,3)
        b = ComplexIt(2,4)
        print(a+ b)
```

Test the following code on ComplexIt module جرب الكود الموالي على وحدة ComplexIt

```
import aspectlib
import unittest
import sys
from complexit import ComplexIt

class MyTestCase(unittest.TestCase):

    def test_stuff(self):

        @aspectlib.Aspect
        def mock_add(self, value):
            if value == 0:
                yield aspectlib.Return(0)
            else:
                yield aspectlib.Proceed

        with aspectlib.weave(ComplexIt.__add__, mock_add):
            a = ComplexIt(1,2)
            b = ComplexIt(2,2)
            c= ComplexIt(3,4)
            self.assertEqual(a.__add__(b), c)

if __name__ == '__main__':
    unittest.main()
```

2.10.5 Homework

واجب منزلي

لا واجب منزلي

2.10.6 References

مراجع

- Python-aspectlib's documentation https://python-aspectlib.readthedocs.io/en/latest/
- Python-pytilities's documentation https://pytilities.sourceforge.net/doc/1.2.0/
- Aspect Oriented Programming by Spring https://docs.spring.io/spring-python
- Joseph D. Gradecki, Mastering AspectJ: Aspect-Oriented Programming in Java 1st Edition, Wiley. (Gradecki and Lesiecki, 2003)
- Kiczales, Gregor, et al. "Aspect-oriented programming." European conference on object-oriented programming. Springer, Berlin, Heidelberg, 1997 (Kiczales et al., 1997). [7].
- Pahlsson, Niklas. "Aspect-oriented programming." Topic Report for Software Engineering (2002): 11-03 (Påhlsson, 2002) [8].
- Lee, Ken Wing Kuen. "An introduction to aspect-oriented programming." COMP610E: Course of Software Development of E-Business Applications (Spring 2002), Hong Kong University of Science and Technology (2002) (Lee, 2002)[9].
- شرح مفصل لـ Spring-AOP https://arabicprogrammer.com/article/8817216166/

[7] https://apps.dtic.mil/sti/pdfs/ADA417906.pdf
[8] http://oberon2005.oberoncore.ru/paper/np2002.pdf
[9] https://www.inf.utfsm.cl/~mcloud/iwi-253/lecturas/Introduction_to_Aspect_Oriented_Programming.pdf

Chapter 3

Tests فحوص

3.1 Tests n°1

Test n°1 : for chapter 1

الفحوص رقم 1 للفصل الاول

3.1.1 Topic n°1

1. List the differences between DTD and XSchema
2. Classify computer languages according to use in Front-End and Back-End
3. List the differences between using XSL/T and the DOM API
4. What are the basic rules for writing an XML document?
5. What is the difference between CDATA and PCDATA?
6. What is XML namespace?
7. Are spaces ignored by XML?
8. Can we use graphics in XML?
9. What is the contribution of MVC for the Front-End and Back-End.
10. Give the formula in Excel that calculates the attendance score based on the number of attendances and the number of sessions.

1. اذكر الفروق بين DTD وXschema
2. صنّف لغات البرمجة حسب استخدامها في الواجهة الأمامية والخلفية Front-End و Back-End
3. اذكر الفروق بين استخدام XSL/T وAPI DOM
4. ما هي القواعد الأساسية لكتابة مستند XML؟
5. ما هو الفرق بين CDATA وPCDATA؟
6. ما هي مساحة اسم space name في XML؟
7. هل يتم تجاهل المسافات (الفراغات) بواسطة XML؟
8. هل يمكن استخدام رسومات في XML؟
9. ما هي مساهمة MVC للواجهة الأمامية والخلفية؟
10. أعط الصيغة في Excel التي تسمح لك بحساب علامة (نقطة) الحضور للطالب بناءً على عدد مرات الحضور وعدد الحصص.

3.1.2 Topic n°2

Project : We want to develop a server dedicated to prayer timesمواقيت الصلاة around the world, it offers an API for various applications. We assume to have an XML database of cities and their schedules during the solar year.

تحتسب **مواقيت الصلاة** للمسلمين لمدينة ما حسب خطوط الطول والعرض والارتفاع عن سطح الأرض وتاريخ اليوم الشمسي، وعادة ما تكون ثابتة بالنسبة ليوم ما في السنة الشمسية. تحتسب مواقيت الصلاة بطرائق مختلفة لاسيما صلاتي الفجر والعشاء. يختار كل مستخدم طريقة حسب المذهب الذي يتبعه أو الدولة التي ينتمي إليها.

للمشروع أعلاه

1. أنشئ ملف تعريف DTD لمواقيت الصلاة.
2. استخدم XSLT لعرض أوقات الصلاة لمدينة معينة في تاريخ معين.

3 اكتب دالة بايثون تعرض أوقات الصلاة لمدينة معينة في تاريخ معين، من ملف XML باستخدام DOM

For the project :

1 Define a DTD for prayer times.

2 Use XSLT to display prayer times for a given city on a given date.

3 Write a python function that returns the prayer times of a given city on a given date, from an XML file using DOM.

3.1.3 Topic n°3

التمرين 1 (نقطتان [0.5، 0.25*4؛ 0.5)

1 اذكر مزايا وعيوب DOM API؟

2 أعط الاسم الكامل لاختصارات الآتية ووصفا مختصرًا لها: SGML DOM DTD XSD

3 ما الفرق بين HTML وXHTML؟

التمرين 2 (3 نقاط: 1، 1، 1)

نعتبر مستندًا قرآنيًا فيه آيات، في كل آية على مجموعة من الترجمات والروابط الصوتية.

- س1. اقترح DTD لهذا المستند.

اختر سؤالين:

- س2- اكتب قطعة الكود XSLT (القالب) الذي يسمح لك بعرض الآيات باللغة الإنجليزية.
- س3- اكتب قطعة الكود DOM في لغة بايثون يعرض قائمة السور ذات أكثر من 100 آية.
- س4- اكتب قطعة الكود XSLT الذي بعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س5- اكتب جزءًا من كود DOM بلغة بايثون والذي يعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س6- حوّل DTD إلى XSD.

Exercise 1 (2pts [0.5, 0.25*4; 0.5)

1 List the advantages and disadvantages of DOM API?

2 Give the full name of each abbreviation and a brief definition: XSD DTD DOM SGML

3 What is the difference between HTML and XHTML?

Exercise 2 (3pts: 1, 1, 1)

We have a Quranic document where each ayat contains a set of translations and audio links.

- Q1. Propose a DTD for this document.

Choose two questions:

- Q2- Write the XSLT code fragment (template) which allows you to display the ayates in English.
- Q3- Write the DOM code fragment in python which allows you to display the list of surates which contains more than 100 ayates.
- Q4- Write the XSLT code fragment (template) which allows you to display an index ordered according to the first word of the ayates in Arabic.
- Q5- Write the fragment of the DOM code in python which allows you to display an index ordered according to the first word of the ayates in Arabic.
- Q6- Convert DTD to XSD.

3.2 Tests n°2

Test n°2 :

الفحوص رقم 2

3.2.1 Topic n°1

السؤال 1: أنماط التصميم (نقطتان: 4*0.5) في كل حالة مما يلي ، اذكر نمط التصميم المناسب، مبررًا إجابتك.

1. في العيادات الطبية تخصصات مختلفة حسب توفر الأطباء.
2. لحساب علامة التقويم المستمر تُقترح عدة طرق: البحوث، الواجبات المنزلية، الفحوص.
3. واجهة رسومية تدعم اللغات ذات الاتجاه من اليمين إلى اليسار.
4. مطعم للوجبات السريعة يقدم ساندويتشات مخصصة بالكامل حسب الطلب.

السؤال الثاني: DOM: (3نقاط: 2*1.5)

1. بالنسبة لمشروع القرآن: اكتب دالة بيثون باستخدام DOM للعثور على أطول آية مع رقمها والسورة.
2. بالنسبة لمشروع تتبع الطلاب: اكتب دالة بيثون باستخدام DOM للعثور على الطلاب غير المقصيين الذين يمكنهم اجتياز الامتحان الاستدراكي.

Question 1: Design patterns (2pts: 0.5*4) Choose the appropriate design pattern for each case, justifying your answer.

1. Medical clinics contain different specialties depending on the availability of doctors.
2. For the calculation of the continuous assessment mark, several methods are proposed: presentations, homework, tests.
3. A graphical interface that supports right-to-left languages.
4. A fast-food restaurant that offers completely personalized sandwiches on demand.

Question 2: DOM: (3pts: 2*1.5)

1. For Quran project: Write python function using DOM to find longest aya with number and sura.
2. For student tracking project: Write Python function using DOM to find non-excluded students who can pass the catch-up exam الامتحان الاستدراكي.

3.2.2 Topic n°2

التمرين 1 (نقطتان: 4*0.5) أسئلة :

1. عرّف أنماط التصميم.
2. منهجية MVC هي نمط تصميم هيكلي، اشرح.
3. اذكر الفرق بين المحول Adapter ولوحة الواجهة Facade.
4. اذكر الفرق بين المصنع المجرد Factory Abstract وطريقة المصنع Factory method.

اختر تمرينًا من 2 و 3
التمرين الثاني (3*1 نقاط):

نريد إنشاء مُولّد للأسئلة لمادة ما، ونريد توليد اختبارات (فحص 1، فحص 2، امتحان، استدراك). الفحص الأول فيه أسئلة من الفصول الأولى، والفحص الثاني فيه أسئلة الفصول الأخيرة، في حين أن مواضيع الامتحان والاستدراك تشمل جميع الفصول.
نريد إنشاء الاختبارات من:

- قاعدة من الأسئلة الجاهزة: على سبيل المثال: "س1- عرّف ،XML س2- اذكر مزايا "DOM
- أو الأسئلة القياسية المُولّدة تلقائيًا باستخدام بيانات عشوائية: x+ ؟y => 4+3؟

أسئلة :

1. اقترح نمط التصميم، وتبرير اختيارك.
2. ارسم مخطط الصنف diagram Class للحل.
3. برمج الأصناف من خلال محاكاة الدوال (برمجة سطحية)

التمرين 3 (3*1 نقاط):

في أحد مواقع التجارة الإلكترونية الجزائرية، طريقة الدفع إما بالحساب الجاري CCP أو عند الاستلام. وبالدينار فقط.
نريد تصميم نظامنا للسماح بتوسيع النظام لطرق دفع جديدة وعملات أخرى (اليورو والدولار).

أسئلة :

1. اقترح نمطًا تصميميًّا واحدًا أو أكثر، وبرّر اختيارك.
2. ارسم مخطط الصنف diagram Class للحل.
3. برمج الأصناف من خلال محاكاة الدوال (برمجة سطحية)

Exercise 1 (2 pts: 0.5*4)

Questions :

1. Give a definition of design patterns.
2. MVC is an architectural design; explain.
3. State the difference between the Adapter and the Facade.
4. State the difference between an Abstarct Factory and a Factory Method.

Choose an exercise from 2 and 3.

Exercise 2 (3*1 pts):

We want to create a test generator for a module. We want to generate several tests (test1 and test2, exam, catch-up استدراك). Test 1 contains questions from the first chapters, the second from the last chapters, while the exams and the catch-up cover all chapters.
We want to generate the tests from

- a base of static questions: for example, "Q1- define XML, Q2- cite the advantages of DOM"
- or standard questions automatically generated with random data: x+ y? => 3+4?

Questions :

1. Propose a design pattern and justify your choice.
2. Draw the class diagram for the solution.

Implement classes by simulating methods.

Exercise 3 (3*1 pts):

On an Algerian e-commerce site, the payment method is either currant account CCP or payment on delivery. On the other hand, payment is only made in dinars.
We want to design our system to allow the extension of the system for new payment methods and new currencies (Euro, Dollar).

Questions :

1. Propose one or more design patterns; justify your choice.
2. Draw the class diagram for the solution.
3. Implement classes by simulating methods.

Chapter 4

Test solutions حلول الفحوص

4.1 Chapter one test solutions

4.1.1 Solution n°1

1 List the differences between DTD and XSchema

DTD	Xschema
Special syntax.	XML syntax.
Special language.	XML dialect.
Simple data.	Allows you to define complex data structures.
Simple poor typing	Allows you to control the different complex types.

2 Classify computer languages according to use in Front-End and Back-End

Backend	Front end
PHP	
Javascript (node.js)	Javascript
HTML	CSS
XML	JSON

3 List the differences between using XSL/T and the DOM API

XSL/T	DOM
Operating system independent	Operating system independent
Programming language independent	Programming language independent
Requires browser support	Programmable
Complex syntax	Flexible

4 What are the basic rules for writing an XML document? (closing tags, root element, case sensitive, ignoring space, nesting, attribute value in quotes)

5 What is the difference between CDATA and PCDATA?

PCDATA (parsed data) (CDATA not parsee data).

6 What is XML namespace?

Qualified name associated with DTD/Schema location.

7 Are spaces ignored by XML? Yes.

8 Can we use graphics in XML? Yes, inserted by link or represented in SVG.

9 What is the contribution of MVC for the Front-End and Back-End. The MVC brings the separation of profession between the Back and the front end.

10 Give the formula in Excel that calculates the attendance score based on the number of attendances and the number of sessions.

Let A2 be the number of sessions, B2: the number of attendances

$= IF(A2 - B2 >= 3; 0; 3 - (A2 - B2))$

1 اذكر الفروق بين DTD وXschema

Xschema	DTD
صياغة بلغة XML.	صياغة خاصة
لهجة XML	لغة خاصة
يمكن تعريف هياكل بيانات معقّدة.	بيانات بسيطة
يمكن تعريف أنواع معقّدة.	أنواع بسيطة مفتقرة

2 صنّف لغات البرمجة حسب استخدامها في الواجهة الأمامية والخلفية Front-End و Back-End

end Front	Backend
	PHP
Javascript	(node.js) Javascript
CSS	HTML
JSON	XML

3 اذكر الفروق بين استخدام XSL/T وAPI DOM

DOM	XSL/T
مستقل عن نظام التشغيل	مستقل عن نظام التشغيل
مستقل عن لغة البرمجة	مستقل عن لغة البرمجة
قابل للبرمجة	يتطلب دعم المتصفح
مرونة	صياغة معقّدة

4 ما هي القواعد الأساسية لكتابة مستند XML؟ (إغلاق الوسوم، عنصر الجذر، حساس لحالة الأحرف، تجاهل المسافة، التراكب، قيمة الخاصية بين علامتي الاقتباس)

5 ما هو الفرق بين CDATA وPCDATA؟ PCDATA (بيانات قابلة للمعالجة) CDATA (بيانات غير قابلة للمعالجة).

6 ما هي مساحة اسم space name في XML؟ اسم مؤهّل مرتبط بمكان أو موقع ملف تعريف البيانات. DTD/Schema

7 هل يتم تجاهل المسافات (الفراغات) بواسطة XML؟
نعم

8 هل يمكن استخدام رسومات في XML؟ نعم، بواسطة رابط أو بصيغة الرسومات الشعاعية SVG.

9 ما هي مساهمة MVC للواجهة الأمامية والخلفية؟ يقوم MVC بفصل بين مهن الواجهة الخلفية و مهن الواجهة الأمامية.

10 أعط الصيغة في Excel التي تسمح لك بحساب علامة (نقطة) الحضور للطالب بناءً على عدد مرات الحضور وعدد الحصص. ليكن A2 عدد الحصص، وB2 عدد مرات الحضور، صيغة العلامة هي:
$= IF(A2 - B2 >= 3; 0; 3 - (A2 - B2))$

4.1.2 Solution n°2

Project : We want to develop a server dedicated to prayer timesمواقيت الصلاة around the world, it offers an API for various applications. We assume to have an XML database of cities and their schedules during the solar year.
تحتسب **مواقيت الصلاة** للمسلمين لمدينة ما حسب خطوط الطول والعرض والارتفاع عن سطح الأرض وتاريخ اليوم الشمسي، وعادة ما تكون ثابتة بالنسبة ليوم ما في السنة الشمسية. تحتسب مواقيت الصلاة بطرائق مختلفة لاسيما صلاتي الفجر والعشاء. يختار كل مستخدم طريقة حسب المذهب الذي يتبعه أو الدولة التي ينتمي إليها.

للمشروع أعلاه

1 أنشئ ملف تعريف DTD لمواقيت الصلاة.

2 استخدم XSLT لعرض أوقات الصلاة لمدينة معينة في تاريخ معين.

3 اكتب دالة بايثون تعرض أوقات الصلاة لمدينة معينة في تاريخ معين، من ملف XML باستخدام DOM

For the project :

1 Define a DTD for prayer times.

```
<!ELEMENT PrayerTimes (city)+>
<!ATTLIST PrayerTimes method #CDATA>
<!ELEMENT city (day)+>
<!ATTLIST city name #CDATA
 latitude #CDATA
```

```
 longitude #CDATA>
<!ELEMENT day (date, fadjr, dohr, asr, magrib, ishaa)>
<!ELEMENT date #PCDATA>
<!ELEMENT fadjr #PCDATA>
<!ELEMENT dohr#PCDATA>
<!ELEMENT asr #PCDATA>
<!ELEMENT magrib #PCDATA>
<!ELEMENT ishaa #PCDATA>
```

2 Use XSLT to display prayer times for a given city on a given date.

```
<xsl:template match="/city">

<xsl:for-each select="city[@name='Bouira']">
<h1 align="center"></h1>

<table border=1 align="center">
<xsl:for-each select="day">
    <xsl:test select="day/date == '15/12/2021'">
<tr>
    <td> Fadjr </td> <td> <xsl:value-of select="fadjr"/> </td>
</tr>
<tr>
    <td> Dohr </td> <td> <xsl:value-of select="dohr"/> </td>
</tr>
<tr>
    <td> Asr </td> <td> <xsl:value-of select="asr"/> </td>
</tr>
<tr>
    <td> Maghrib </td> <td> <xsl:value-of select="maghrib"/> </td>
</tr>
<tr>
    <td> Ishaa </td> <td> <xsl:value-of select="ishaa"/> </td>
</tr>

</xsl:for-each>
</xsl:test>


</table>
</xsl:for-each>
</xsl:template>
```

3 Write a python function that returns the prayer times of a given city on a given date, from an XML file using DOM.

```
<xsl:template match="/city">

<xsl:for-each select="city[@name='Bouira']">
<h1 align="center"></h1>

<table border=1 align="center">
```

```
<xsl:for-each select="day">
    <xsl:test select="day/date == '15/12/2021'">
<tr>
    <td> Fadjr </td> <td> <xsl:value-of select="fadjr"/> </td>
</tr>
<tr>
    <td> Dohr </td> <td> <xsl:value-of select="dohr"/> </td>
</tr>
<tr>
    <td> Asr </td> <td> <xsl:value-of select="asr"/> </td>
</tr>
<tr>
    <td> Maghrib </td> <td> <xsl:value-of select="maghrib"/> </td>
</tr>
<tr>
    <td> Ishaa </td> <td> <xsl:value-of select="ishaa"/> </td>
</tr>

</xsl:for-each>
</xsl:test>


</table>
</xsl:for-each>
</xsl:template>
```

4.1.3 Solution n°3

التمرين 1 (نقطتان [0.5، 4*0.25، 0.5)

1 اذكر مزايا وعيوب DOM API؟

2 أعط الاسم الكامل لاختصارات الآتية ووصفا مختصرًا لها: SGML DOM DTD XSD

3 ما الفرق بين HTML وXHTML؟

التمرين 2 (3 نقاط: 1، 1، 1)

نعتبر مستندًا قرآنيًا فيه آيات، في كل آية على مجموعة من الترجمات والروابط الصوتية.

- س1. اقترح DTD لهذا المستند.

اختر سؤالين:

- س2- اكتب قطعة الكود XSLT (القالب) الذي يسمح لك بعرض الآيات باللغة الإنجليزية.
- س3- اكتب قطعة الكود DOM في لغة بايثون يعرض قائمة السور ذات أكثر من 100 آية.
- س4- اكتب قطعة الكود XSLT الذي بعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س5- اكتب جزءًا من كود DOM بلغة بايثون والذي يعرض فهرسًا مرتبًا حسب الكلمة الأولى من آيات القرآن الكريم باللغة العربية.
- س6- حوّل DTD إلى XSD.

Exercise 1 (2pts [0.5, 0.25*4; 0.5)

1 List the advantages and disadvantages of DOM API? advantages: مزايا

- DOM XML is language and platform independent مستقل عن لغات البرمجة ومنصات التشغيل

- XML DOM is browsed — قابل للتنقل عبره
- XML DOM is editable — قابل للتعديل

Disadvantages: — مساوئ

- Consumes memory: if the XML structure is large) as a program written once remains in memory all the time until and unless explicitly removed.

يستهلك كثيرًا من الذاكرة لاسيما إن كان حجم ملف البيانات كبيرًا.

2 Give the full name of each abbreviation and a brief definition: XSD DTD DOM SGML

- **XSD** (XML schema Definition) : is an XML document format description language for defining the structure and content type of an XML document.

لغة توصيف أنواع البيانات المحتواة في مستند XML، تستعمل نفس صياغة XML في تعريف البيانات

- **DTD** (Document Type definition): describes the grammar of the document — list of elements (or tags), attributes, their content and their arrangement.

لغة توصيف أنواع البيانات المحتواة في مستند XML

- **DOM** (Document Object model): is an application programming interface (API) for well-formed XML documents. .

واجهة برمجية للتطبيقات، لمعالجة ملفات XML

- **SGML** (standard generalized Markup language) is a markup description language, ISO standard.

لغة توصيف بيانات عامّة، معيار تابع لمنظمة المعايير الدولية ISO

3 What is the difference between HTML and XHTML?

- Flexible framework requiring lenient HTML specific parser.
- Restrictive subset of XML and needs to be parsed with standard XML parsers.

XHTML هو إطار عمل مرن يتطلب محلل HTML صارمًا. مجموعة فرعية مقيدة من XML ويجب تحليلها باستخدام محللات XML القياسية.

Exercise 2 (3pts: 1, 1, 1)

We have a Quranic document where each ayat contains a set of translations and audio links.

- Q1. Propose a DTD for this document.

```
<!ELEMENT mushaf (sura)+>
<!ELEMENT sura (aya)+>
<!ATTLIST sura
id CDATA #REQUIRED
name CDATA #REQUIRED
nb_aya CDATA #REQUIRED>
<!ELEMENT aya (arabic, translation+, audio+)>
<!ATTLIST aya id CDATA #REQUIRED>
<!ELEMENT arabic PCDATA>
<!ELEMENT translation PCDATA>
<!ATTLIST translation lang CDATA #REQUIRED>
<!ATTLIST audio reciter CDATA #REQUIRED>
```

Choose two questions:

- Q2- Write the XSLT code fragment (template) which allows you to display the ayates in English.

```
<xsl:for-each select="surat/aya/translation[@lang='english']">
    …
</xsl:for-each>
```

- Q3- Write the DOM code fragment in python which allows you to display the list of surates which contains more than 100 ayates.

```
result =[]
for srt in xmldoc.getElementsByTagName('sura'):
     nb_aya = srt.getAttribute('nb_aya')
     name = srt.getAttribute('name')
     if  nb_aya >100 :
       result.append({"aya":aya_id, "sura":name, "aya_text":aya_text})
print(result)
```

- Q4- Write the XSLT code fragment (template) which allows you to display an index ordered according to the first word of the ayates in Arabic.

```
<xsl:apply-templates select="mushaf">
    <xsl:sort select="*/*/aya/arabic"/>
</xsl:apply-templates>
```

- Q5- Write the fragment of the DOM code in python which allows you to display an index ordered according to the first word of the ayates in Arabic.

```
result =[]
for aya in xmldoc.getElementsByTagName('arabic'):
     result =  aya.firstChild.data
result = sorted(result)
print(result)
```

- Q6- Convert DTD to XSD.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="translation">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="lang"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="audio">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute type="xs:string" name="qari"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="aya">
    <xs:complexType>
      <xs:sequence>
```

```
        <xs:element ref="translation" maxOccurs="unbounded" minOccurs="1"/>
        <xs:element ref="audio" maxOccurs="unbounded" minOccurs="1"/>
      </xs:sequence>
      <xs:attribute type="xs:interger" name="id"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="sura">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="aya" maxOccurs="unbounded" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute type="xs:integer" name="nb_aya" use="optional"/>
      <xs:attribute type="xs:integer" name="id" use="optional"/>
      <xs:attribute type="xs:string" name="name" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="mushaf">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="sura" maxOccurs="114" minOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4.2 Chapter one test solutions

4.2.1 Solution n°1

السؤال 1: أنماط التصميم (نقطتان: 0.5*4) في كل حالة مما يلي ، اذكر نمط التصميم المناسب، مبررًا إجابتك.

1 في العيادات الطبية تخصصات مختلفة حسب توفر الأطباء.

2 لحساب علامة التقويم المستمر تُقترح عدة طرق: البحوث، الواجبات المنزلية، الفحوص.

3 واجهة رسومية تدعم اللغات ذات الاتجاه من اليمين إلى اليسار.

4 مطعم للوجبات السريعة يقدم ساندويتشات مخصصة بالكامل حسب الطلب.

السؤال الثاني: DOM: (3نقاط: 2*1.5)

1 بالنسبة لمشروع القرآن: اكتب دالة بيثون باستخدام DOM للعثور على أطول آية مع رقمها والسورة.

2 بالنسبة لمشروع تتبع الطلاب: اكتب دالة بيثون باستخدام DOM للعثور على الطلاب غير المقصيين الذين يمكنهم اجتياز الامتحان الاستدراكي.

Question 1: Design patterns (2pts: 0.5*4) Choose the appropriate design pattern for each case, justifying your answer.

1 Medical clinics contain different specialties depending on the availability of doctors.

Factory Method:

- an abstract clinic contains all possible specialties,
- a concrete clinic contains a subset of specialties.

توفر العيادة افتراضيا كل التخصصات، بينما في الواقع توفر العيادة ما أمكنها.

2 For the calculation of the continuous assessment mark, several methods are proposed: presentations, homework, tests.

Strategy :

- The evaluation class delegates the calculation method to another class "calculation method", so we can change the calculation strategy without changing the evaluation class.

يمكن لنصف التقييم أن يفوّض طريقة الحساب لكائن آخر، أي تغيير استراتيجية الحساب، دون تعديل كائن التقييم.

3 A graphical interface that supports right-to-left languages.

Abstract Factory :

- Interfaces are created in two ways for each object, LTR by default or RTL.

تُبنى الواجهات بشكلين لكل كائن، من اليسار إلى اليمين أو من اليمين إلى اليسار.

Another solution: **Adapter :**

- - adapt the original GUIs to languages from right to left.

حل آخر باستعمال المحول: أي تحويل الواجهة الأصلية إلى واجهة من اليمين إلى اليسار

4 A fast-food restaurant that offers completely personalized sandwiches on demand.

Builder :

- The sandwiches class can have complements added on demand, so this class is constructed with added parts.

يمكن إضافة مكملات لصنف السندويشات حسب الطلب، لذلك تم إنشاء هذا الصنف بأجزاء إضافية.

Question 2: DOM: (3pts: 2*1.5)

1 For Quran project: Write python function using DOM to find longest aya with number and sura.

```
DATA_FILE = 'data/mushaf-amma1.xml'
import xml.dom.minidom as minidom
import sys
def main():
    try:
        # o
        xmldoc = minidom.parse(DATA_FILE)
    except:
        print("Can't Open the file", DATA_FILE)
        sys.exit()
    suraname_max = ""
    aya_id_max = ""
    aya_text_max = ""
    suras = xmldoc.getElementsByTagName('sura')
    for sura in suras:
        suraname = sura.getAttribute("name")
        # get aya
        ayas = xmldoc.getElementsByTagName('aya')
        for aya in ayas:
            aya_text = aya.firstChild.data
            if len(aya_text) >len(aya_text_max):
                suraname_max = suraname
                aya_id_max = aya.getAttribute("id"),
                aya_text_max = aya_text

    print(suraname_max, aya_id_max, aya_text_max)
    return 0
if __name__ == '__main__':
    main()
```

2 For student tracking project: Write Python function using DOM to find non-excluded students who can pass the catch-up exam الامتحان الاستدراكي.

```
DATA_FILE = 'data/suividata.xml'
import xml.dom.minidom as minidom
import sys
def main():
  try:
    # o
    xmldoc = minidom.parse(DATA_FILE)
  except:
    print("Can't Open the file", DATA_FILE)
    sys.exit()
  students = xmldoc.getElementsByTagName('student')
  for std in students:
    exclusion = std.getElementsByTagName("exclusion")[0].firstChild.data
    print(exclusion)
    if exclusion == "non":
      avg_before_catchup = std.getElementsByTagName("Avg")[0].firstChild.data
      if float(avg_before_catchup) < 10:
        name = std.getElementsByTagName("name")[0].firstChild.data
```

```
        firstname = std.getElementsByTagName("firstname")[0].firstChild.data
        print(name, firstname, avg_before_catchup)
  return 0
if __name__ == '__main__':
  main()
```

4.2.2 Solution n°2

التمرين 1 (نقطتان: 4*0.5) أسئلة :

1. عرّف أنماط التصميم.
2. منهجية MVC هي نمط تصميم هيكلي، اشرح.
3. اذكر الفرق بين المحول Adapter ولوحة الواجهة Facade.
4. اذكر الفرق بين المصنع المجرد Factory Abstract وطريقة المصنع Factory method.

اختر تمرينًا من 2 و 3

التمرين الثاني (3*1 نقاط):

نريد إنشاء مُولّد للأسئلة لمادة ما، ونريد توليد اختبارات (فحص 1، فحص 2، امتحان، استدراك). الفحص الأول فيه أسئلة من الفصول الأولى، والفحص الثاني فيه أسئلة الفصول الأخيرة، في حين أن مواضيع الامتحان والاستدراك تشمل جميع الفصول.
نريد إنشاء الاختبارات من:

- قاعدة من الأسئلة الجاهزة: على سبيل المثال: "س1- عرّف XML، س2- اذكر مزايا DOM"
- أو الأسئلة القياسية المُولّدة تلقائيًا باستخدام بيانات عشوائية: ؟x+ y؟ <= 4+3؟

أسئلة :

1. اقترح نمط التصميم، وتبرير اختيارك.
2. ارسم مخطط الصنف diagram Class للحل.
3. برمج الأصناف من خلال محاكاة الدوال (برمجة سطحية)

التمرين 3 (3*1 نقاط):

في أحد مواقع التجارة الإلكترونية الجزائرية، طريقة الدفع إما بالحساب الجاري CCP أو عند الاستلام. وبالدينار فقط.
نريد تصميم نظامنا للسماح بتوسيع النظام لطرق دفع جديدة وعملات أخرى (اليورو والدولار).

أسئلة :

1. اقترح نمطًا تصميميًّا واحدًا أو أكثر، وبرّر اختيارك.
2. ارسم مخطط الصنف diagram Class للحل.
3. برمج الأصناف من خلال محاكاة الدوال (برمجة سطحية)

Exercise 1 (2 pts: 0.5*4)

Questions :

1. Give a definition of design patterns.

   In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. (Shvets, 2018)

   في هندسة البرمجيات، نمط التصميم هو حل عام قابل للتكرار لمشكلة شائعة في تصميم البرمجيات. نمط التصميم ليس تصميمًا نهائيًا يمكن تحويله مباشرة إلى كود. إنه وصف أو قالب لكيفية حل مشكلة يمكن استخدامه في العديد من المواقف المختلفة.

2 MVC is an architectural design; explain.

MVC is an architectural pattern which means it rules the whole architecture of the applications. Even though often it is known as design pattern but we may be wrong if we refer it only as a design pattern because design patterns are used to solve a specific technical problem, whereas architecture pattern is used for solving architectural problems, so it affects the entire architecture of our application. (Svirca, 2020)

الMVC هو نمط هيكلي (معماري) مما يعني أنه يحدد البنية الكاملة للتطبيقات. على الرغم من أنه يُعرف في كثير من الأحيان بنمط التصميم إلا أننا قد نكون مخطئين إذا أشرنا إليه كنمط تصميم فقط لأن أنماط التصميم تستخدم لحل مشكلة فنية معينة، في حين يستخدم النمط المعماري لحل المشكلات المعمارية، لذلك فهو يؤثر على كامل بنية تطبيقنا.

3 State the difference between the Adapter and the Facade.

An Adapter is used when the wrapper must respect a particular interface and must support a polymorphic behavior. On the other hand, a facade is used when one wants an easier or simpler interface to work with (Freeman and Robson, 2020).

يستخدم المحول عندما يجب أن نحافظ على واجهة معينة وندعم سلوكًا متعدد الأشكال. من ناحية أخرى، يتم استخدام الواجهة عندما يريد المرء واجهة أسهل أو أبسط للعمل معها.

4 State the difference between an Abstarct Factory and a Factory Method.

- **Method Factory:** Provides an abstract interface to create ONE product.
- **Abstract Factory:** Provides an abstract interface to create a product family
- **Method Factory:** Each subclass chooses which concrete class to instantiate
- **Abstract Factory:** Each concrete subclass creates a product family

- طريقة المصنع: يوفر واجهة مجردة لإنشاء منتج واحد.
- المصنع المجرّد: يوفر واجهة مجردة لإنشاء مجموعة منتجات
- مصنع الطريقة: يختار كل صنف فرعي الصنف الواقعي الذي سينشئ الأفراد.
- المصنع المجرّد: ينشئ كل صنف فرعي واقعي عائلة منتجات

Choose an exercise from 2 and 3.

Exercise 2 (3*1 pts):

We want to create a test generator for a module. We want to generate several tests (test1 and test2, exam, catch-up استدراك). Test 1 contains questions from the first chapters, the second from the last chapters, while the exams and the catch-up cover all chapters.

We want to generate the tests from

- a base of static questions: for example, "Q1- define XML, Q2- cite the advantages of DOM"
- or standard questions automatically generated with random data: x+ y? => 3+4?

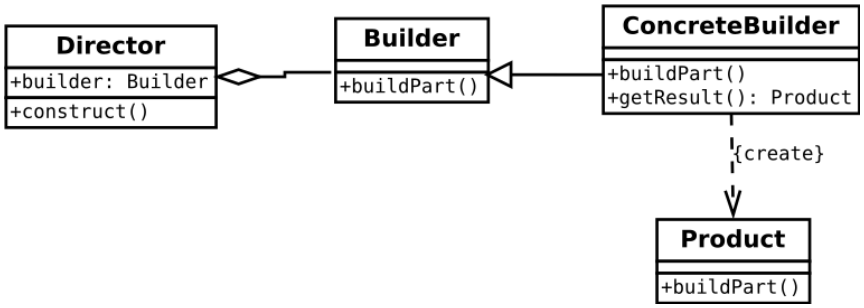
Questions :

1 Propose a design pattern and justify your choice.

The builder, because the tests and exams have the same form composed of questions, the choice of questions is made on the fly during the creation of the object.

البناء نظرًا لأن الاختبارات والامتحانات لها نفس الشكل أي مكونة من أسئلة. يتم اختيار الأسئلة أثناء إنشاء الكائن.

2 Draw the class diagram for the solution.



Implement classes by simulating methods.

```
       class Director:
    """
    Construct an object using the Builder interface.
    """
    def __init__(self):
        self._builder = None
    def construct(self, builder):
        self._builder = builder
    def test_one(self,):
        self._builder._build_questions_chapitre_1()
        self._builder._build_questions_chapitre_2()
        self._builder._build_questions_chapitre_2_dynamique()
    def test_two(self,):
        self._builder._build_questions_chapitre_3()
        self._builder._build_questions_chapitre_3_dynamique()


    def examen(self,):
        self._builder._build_questions_chapitre_1()
        self._builder._build_questions_chapitre_2()
        self._builder._build_questions_chapitre_3_dynamique()

class Builder():
    """
    Specify an abstract interface for creating parts of a Product
    object.
    """
    def __init__(self):
        self.product = Product()
    def _build_questions_chapitre_1(self):
        pass
    def _build_questions_chapitre_2(self):
        pass
    def _build_questions_chapitre_3(self):
        pass
    def _build_questions_chapitre_1_dynamique(self):
        pass
    def _build_questions_chapitre_2_dynamique(self):
        pass
    def _build_questions_chapitre_3_dynamique(self):
        pass


class TestBuilder(Builder):
    def _build_questions_chapitre_1(self):
        self.product.add("Questions sur le chapitre n° 1")

    def _build_questions_chapitre_2(self):
        self.product.add("Questions sur le chapitre n° 2")

    def _build_questions_chapitre_3(self):
```

```
        self.product.add("Questions sur le chapitre n° 3")
    def _build_questions_chapitre_1_dynamique(self):
        self.product.add("Questions sur le chapitre n° 1")

    def _build_questions_chapitre_2_dynamique(self):
        self.product.add("Questions sur le chapitre n° 2")

    def _build_questions_chapitre_3_dynamique(self):
        self.product.add("Questions sur le chapitre n° 3")

class Product:
    """
    Represent the complex object under construction.
    """
    def __init__(self, ):
        self.question_list = []
    def add(self, qst):
        self.question_list.append(qst)
    def __repr__(self,):
        return u"\n".join(self.question_list)
def main():
    concrete_builder = TestBuilder()
    director = Director()
    director.construct(concrete_builder)
    director.test_one()
    product = concrete_builder.product
    print(product)
if __name__ == "__main__":
    main()
```

Exercise 3 (3*1 pts):

On an Algerian e-commerce site, the payment method is either currant account CCP or payment on delivery. On the other hand, payment is only made in dinars.

We want to design our system to allow the extension of the system for new payment methods and new currencies (Euro, Dollar).

Questions :

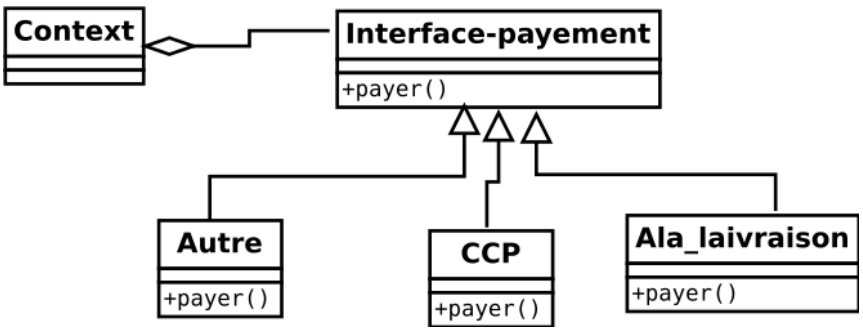
1 Propose one or more design patterns; justify your choice.

- For **electronic payment** we propose the **Strategy** design pattern
- For the **currency**, we propose an **adapter** for the conversion.

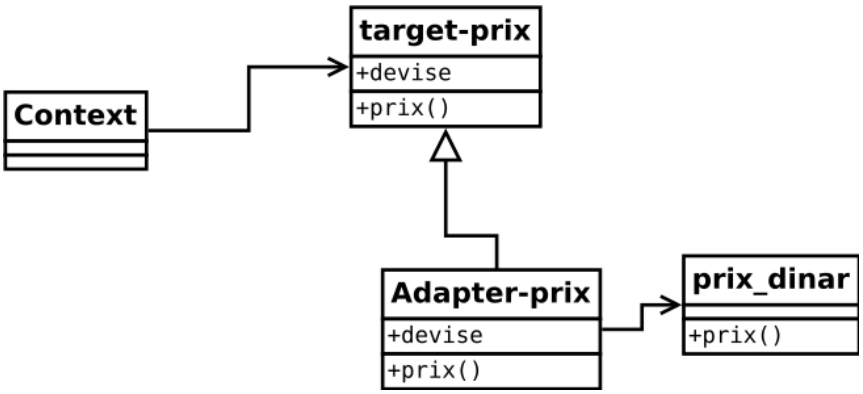
للدفع الالكتروني نختار نمط تصميم "الاستراتيجية". أما للعملة فنختار نمط "المحوّل".

2 Draw the class diagram for the solution.

For payment للدفع



For currency للعملة



3 Implement classes by simulating methods.

```
class Context:
    """
    Define the interface of interest to clients.
    Maintain a reference to a Strategy object.
    """
    def __init__(self, strategy):
        self._strategy = strategy

    def context_interface(self):
        self._strategy.payer()

class Strategy_payement():
    """
    Declare an interface common to all supported algorithms. Context
    uses this interface to call the algorithm defined by a
    ConcreteStrategy.
    """
    def payer(self):
        pass

class strategy_ccp(Strategy_payement):
    def payer(self):
        print("payer par CCP")
class strategy_laivraison(Strategy_payement):
    def payer(self):
        print("payer par CCP")
```

```
class Target_prix():
    """
    Define the domain-specific interface that Client uses.
    """
    def __init__(self):
        self.devise = devise
        self.price = price
        self._adaptee = Adaptee()
    def prix(self):
        return self.price

class Adapter_prix(Target_prix):
    """
    Adapt the interface of Adaptee to the Target interface.
    """
    def __init__(self, prix_dinar, devise):
        self.devise = devise
        self.price = self.convert(prix_dinar.prix())
    def prix(self):
        return self.price
    def convert(self, price):
        return price // 190
        pass

class Prix_dinar:
    """
    Define an existing interface that needs adapting.
    """
    def __init__(self, price):
        self.price = price
    def prix(self):
        return self.price

def main():
    mode_payement = strategy_ccp()
    context = Context(mode_payement)
    context.context_interface()

    prix_local = Prix_dinar(1200)
    prix_en_devise = Adapter_prix(prix_local, "euro")
    print("prix en dinar", prix_local.prix())
    print("prix en", prix_en_devise.devise, prix_en_devise.prix() )

if __name__ == '__main__':
    main()
```

Chapter 5

Exams امتحانات

5.1 Exams امتحانات

5.1.1 Topic n°1

مشروع :
بمناسبة كأس إفريقيا للأمم للمحليين المنظمة في الجزائر، يمكن حجز تذاكر المباريات من موقع "تذكرتي"، الدفع بالبطاقة الذهبية أو البنكية، تستلم وصل دفع، ثم تذهب إلى أقرب مركز لطباعة التذكرة،
هذه صورة لحساب مستخدم في موقع "تذكرتي" :

National identification number :	160345206800000007		
Username :	أحمد بن أحمد	Address :	Bouira
First name :	أحمد	Gender :	Male
Last name :	بن أحمد	Phone :	(+213) 066-666-666
Email :	wefawe6521@prolug.com		
Date of Birth :	October 15, 2002		

تمرين 1 (8 نقاط : 2، 1×4، 1×2) : منهجية

1. اذكر لغات برمجة لهذا المشروع، مع التوضيح
2. اذكر أربعة أنماط تصميم Patterns Design في المشروع مع التحليل والتعليل
3. اذكر دور تقنيات الانعكاسية Reflection والبرمجة جانبية المنحى Programming Oriented Aspect في المشروع

تمرين 2 (10 نقاط : 2، 2 ، 2*3) : تطبيقي
نريد حفظ بيانات المستخدم في مشروع "تذكرتي" في ملف XML، ثم معالجته.

1. 1. أنشئ تعريفا لهيكل البياناتXchema (وحدّد قيود أنواع البريد الإلكتروني، والجنس ورقم الهاتف)
2. استعمل xslt أو DOM Python لعرض البيانات كما في الصورة أعلاه

تمرين 3 (2 نقطتان) : البرمجية الوظيفية
تتكون علامة التقويم المستمر في مادة MTI من علامة الحضور 2، علامة الفحصين 2×5، علامة الواجبات المنزلية 2×6، بعلامة قصوى 20/20.
اكتب صيغة الحساب بلغة وظيفية langage fonctionnel.

5.1.2 Topic n°2

التمرين 1 (7 نقاط: 2، 1، 1، 1، [4*0.5]) أسئلة المنهجية

1. اذكر بعض أنماط التصميم وتصنيفها(0.25 لكل إجابة صحيحة، -0.25 لكل إجابة غير صحيحة)، بحد أقصى نقطتين
2. ما نمط التصميم الذي يحدد تبعية واحد إلى متعدد بين الكائنات؟
3. ما نمط التصميم الذي يوفر فئة واحدة توفر الدوال المبسطة التي يطلبها العميل ويفوّض الاستدعاء لهذه الدوال؟
4. ما القواعد الأساسية لكتابة مستند XML؟
5. ما الفرق بين CDATA و PCDATA؟
6. ما هو نطاق الاسم space name للغة XML؟
7. هل يتم تجاهل المسافات بواسطة XML؟
8. هل يمكننا استخدام الرسوميات في XML؟

المشروع: (التمرين 2 و 3)

نريد تصميم نظام لإدارة الامتحانات ذات الإجابات المتعددة MCQ) أسئلة متعددة الخيارات) عن الويب.

بنك الأسئلة فيه عدد كبير من الأسئلة ذات الخيارات. كل سؤال فيه الحل الصحيح، ويمكن أن يحوي على الإجابة التي اختارها المستخدم.

التمرين 2 (10 نقاط [1، 2، 2، 2، 1.5، 1.5]) :XML

أسئلة :

1. عرّف DTD للأسئلة ذات الخيارات .
2. أعط مخطط Xschema للأسئلة ذات الخيارات.
3. ا عرض الأسئلة باستخدام XSLT.
4. اعرض نتيجة الاختبار باستخدام XSLT.
5. اعرض الأسئلة باستخدام Python/DOM.
6. اعرض نتيجة الاختبار والعلامة باستخدام Python/DOM.

التمرين 3 (3 نقاط: 2*0.75):

نريد توسيع نظام إدارة الاختيار المتعدد الخاص بنا، ونريد:

1. إنشاء أسئلة MCQ بإجابة واحدة صحيحة، أو عدة إجابات صحيحة
2. عرض الاختبار باللغة العربية أو الأجنبية حسب إعدادات المستخدم.
3. متابعة تقدّم النتيجة أثناء الاختبار.
4. حفظ نسخة من الاختبارات لكل طالب.

أعط نمط التصميم المناسب لكل خيار. برّر جوابك.

Exercise 1 (7 pts: 2, 1, 1, 1, [0 5*4])) Methodology questions

1. Cite some design-patterns and their categories **(0 25 for each correct answer, -0 25 for each incorrect answer), max 2 points**
2. Which design pattern defines the one-to-many dependency between objects?
3. What is the design pattern that provides a single class that provides the simplified methods required by the client and delegates the call to those methods?
4. What are the basic rules for writing an XML document?
5. What is the difference between CDATA and PCDATA?
6. What is XML namespace?
7. Are spaces ignored by XML?
8. Can we use graphics in XML?

Project: (exercises 2 and 3)

We want to design a system for managing multiple choice questions (MCQ (for the Web.

A question bank contains a multitude of questions. Each question contains the correct solution, it can contain the answer chosen by the user.

Question Bank

1. What is XML
   - A. ○ eXentible Markup Language
   - B. ○ eXpandable Style Language
   - C. ○ eXtensible Stylesheet Language
   - D. ○ eXtra Style Language

Exercise 2 (10 pts [1, 2, 2, 2, 1.5, 1.5]) XML:
Questions :

1. Define a DTD for MCQ.
2. Define an XSchema for MCQ.
3. Show questions with XSLT.
4. View test result with XSLT.
5. Show questions with Python/DOM.
6. Show test result and grade with Python/DOM.

Exercise 3 (3 pts: 0.75*2):
We want to extend our multiple choice management system, we want to:

1. Generate MCQ questions with a single correct answer, or several answers.
2. Display the test in Arabic or in English depending on the user's configuration.
3. Follow the progress of the score during the test.
4. Save a copy of the tests for each student.

Give the appropriate design pattern for each option. Justify your answer.

Chapter 6

Exams Solutions حلول الامتحانات

6.1 Exam solutions حلول امتحانات

6.1.1 Solution n°1

examS12023

مشروع :

بمناسبة كأس إفريقيا للأمم للمحليين المنظمة في الجزائر، يمكن حجز تذاكر المباريات من موقع "تذكرتي"، الدفع بالبطاقة الذهبية أو البنكية، تستلم وصل دفع، ثم تذهب إلى أقرب مركز لطباعة التذكرة،

هذه صورة لحساب مستخدم في موقع "تذكرتي" :

National identification number :	160345206800000007		
Username :	أحمد بن أحمد	Address :	Bouira
First name :	أحمد	Gender :	Male
Last name :	بن أحمد	Phone :	(+213) 066-666-666
Email :	wefawe6521@prolug.com		
Date of Birth :	October 15, 2002		

تمرين 1 (8 نقاط : 2، 1×4، 1×2) : منهجية

1 اذكر لغات برمجة لهذا المشروع، مع التوضيح

- جافا سكريبت : لغة برمجة لواجهة المستخدم، وواجهة الإدارة backend frontend,
- لغة واجهة تفاعلية جافاسكريبت، html، CSS
- لغة الإدارة يمكن استخدام PHP, بيثون، جافاسكريبت.

اذكر أربعة أنماط تصميم Patterns Design في المشروع مع التحليل والتعليل

2
a. نمط الملاحظ observer لإرسال الإشعارات للمستخدم والإدارة عند الحجز أو نفاد الكمية
b. نمط الوكيل proxy يسمح للمراكز المختلفة طباعة التذاكر
c. نمط الواجهة facade يسمح بشراء التذاكر مركزيا من الموقع لمختلف الأحداث الرياضية
d. نمط الاستراتيجية strategy لاختيار طريقة الدفع الالكتروني بين بطاقة الذهبية أو بطاقة بنكية
e. نمط الوكيل في الاتصال بوسيلة الدفع الالكتروني الذهبية أو البنكية
f. نمط prototype التذكرة لها نفس التصميم، مع تعديلات لكل متفرج
g. نمط Factory method : إنتاج أنواع مختلفة من التذاكر (صفوف أولى، منصة شرفية، ...)
h. الإجابة ليست حصرية

3 اذكر دور تقنيات الانعكاسية Reflection والبرمجة جانبية المنحى Programming Oriented Aspect في المشروع

- تستعمل تقنيات الانعكاسية في تنقيح برمجة الموقع، وحل مشاكله، debugging.
- تستعمل تقنيات البرمجة جانبية المنحى في ضبط سجلات الاستخدامlogging ، تسيير قواعد البيانات، databases ربط الموقع بخوادم الدفع الالكتروني.

تمرين 2 (10 نقاط : 2، 2 ، 3*2) : تطبيقي

نريد حفظ بيانات المستخدم في مشروع "تذكرتي" في ملف XML، ثم معالجته.

1 1. أنشئ تعريفا لهيكل البياناتXchema (وحدّد قيود أنواع البريد الإلكتروني، والجنس ورقم الهاتف)

```
encoding="utf-8"?> version="1.0" <?xml
 1.0 Tools Online Technologies Liquid with Created <!--
```

```
--> (https://www.liquid-technologies.com)
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="users">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="user" type="userType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="userType">
    <xs:sequence>
      <xs:element name="nid" type="xs:string"/>
      <xs:element name="username" type="xs:string" />
      <xs:element name="firstname" type="xs:string" />
      <xs:element name="lastname" type="xs:string" />
      <xs:element name="gender" type="genderType" />
      <xs:element name="email" type="emailType" />
      <xs:element name="birthdate" type="xs:date" />
      <xs:element name="address" type="xs:string" />
      <xs:element name="phone" type="phoneType" />
    </xs:sequence>
  </xs:complexType>


    <xs:simpleType name="genderType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="male"/>
        <xs:enumeration value="female"/>
      </xs:restriction>

    </xs:simpleType>
  <xs:simpleType name="emailType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[^@]+@[^\.]+\..+"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="phoneType">
  <xs:restriction base="xs:string">
   <xs:pattern value="0[0-9]{9}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

2 استعمل xslt أو DOM Python لعرض البيانات كما في الصورة أعلاه

n script

3 استعمل dom python لعرض بعض الإحصائيات :

a. توزيع نسبة الذكور والإناث

b. توزيع الأعمار كل 10 سنوات

c. توزيع المتفرجين حسب الولايات.

```
def get_rate_gender(xmldoc):
    """
distribution gender
"""
    gender_table={"male":0,
    "female":0}
    genders = xmldoc.getElementsByTagName('gender')
    for gender in genders:
        value = gender.firstChild.data
        if value in gender_table:
            gender_table[value] +=1
    total = len(genders)
    rates print #
    for key in gender_table:
        print(key, gender_table[key], gender_table[key]/total)

def get_rate_age(xmldoc):
    """
distribution age
"""
    age_table={0:0, years [1-10[ #
    1:0,years [10-20[ #
    2:0,years [20-30[ #
    3:0,years [30-40[ #
    4:0,years [40-50[ #
    5:0,years [50-60[ #
    }
    births = xmldoc.getElementsByTagName('birthdate')
    for birth in births:
        value = birth.firstChild.data
        year = value.split("-")[0]
        age = 2023 - int(year)
        decade = age // 10  10 div #
        if decade in age_table:
            age_table[decade] +=1
    total = len(births)
    rates print #
    for key in age_table:
        print("[%d  - 0%d0["%(key, key+1), age_table[key], age_table[key]/total)
def get_rate_address(xmldoc):
    """
distribution address
"""
    address_table={
    }
    addresses = xmldoc.getElementsByTagName('address')
    for address in addresses:
        value = address.firstChild.data
        if value in address_table:
            address_table[value] +=1
```

```
        else:
            address_table[value] =1
    total = len(addresses)
    rates print #
    for key in address_table:
        print(key, address_table[key], address_table[key]/total)
DATA_FILE = 'data/tadkiratidata.xml'
import xml.dom.minidom as minidom
import sys
def main():
    try:
        o #
        xmldoc = minidom.parse(DATA_FILE)
    except:
        print(file" the Open t'"Can, DATA_FILE)
        sys.exit()

    get_rate_gender(xmldoc)
    get_rate_age(xmldoc)
    get_rate_address(xmldoc)

    return 0
if __name__ == '__main__':
    main()
```

تمرين 3 (2 نقطتان) : البرمجية الوظيفية

تتكون علامة التقويم المستمر في مادة MTI من علامة الحضور 2، علامة الفحصين 5×2، علامة الواجبات المنزلية 2×6، بعلامة قصوى 20/20.

اكتب صيغة الحساب بلغة وظيفية langage fonctionnel.

1 **الطريقة الأولى :**

$$note = min(20, sum(test1, test2, tp1, tp2, tp3, tp4, tp5, tp6, presence))$$

2 **الطريقة الثانية**

$$(min\ 20(+\ test1\ test2\ tp1\ tp2\ tp3\ tp4\ tp5\ tp6\ presence))$$

3 **الطريقة الثالثة :** هذه الطريقة لا تحسب علامة المادة، بل تشير فقط إلى الصيغة.

$$(min\ 20\ (+(*5\ test)\ (*2\ tp)\ presence))$$

6.1.2 Solution n°2

examS12022

التمرين 1 (7 نقاط: 2، 1، 1، 1، [4*0.5]) أسئلة المنهجية

1 اذكر بعض أنماط التصميم وتصنيفها(0.25 لكل إجابة صحيحة، -0.25 لكل إجابة غير صحيحة)، بحد أقصى نقطتين

2 ما نمط التصميم الذي يحدد تبعية واحد إلى متعدد بين الكائنات؟

3 ما نمط التصميم الذي يوفر فئة واحدة توفر الدوال المبسطة التي يطلبها العميل ويفوّض الاستدعاء لهذه الدوال؟

4 ما القواعد الأساسية لكتابة مستند XML؟

5 ما الفرق بين CDATA و PCDATA؟

6 ما هو نطاق الاسم space name للغة XML؟

7 هل يتم تجاهل المسافات بواسطة XML؟

8 هل يمكننا استخدام الرسوميات في XML؟

المشروع: (التمرين 2 و 3)

نريد تصميم نظام لإدارة الامتحانات ذات الإجابات المتعددة MCQ) أسئلة متعددة الخيارات) عن الويب.

بنك الأسئلة فيه عدد كبير من الأسئلة ذات الخيارات. كل سؤال فيه الحل الصحيح، ويمكن أن يحوي على الإجابة التي اختارها المستخدم.

التمرين 2 (10 نقاط [1، 2، 2، 2، 1.5، 1.5]) :XML

أسئلة :

1 عرّف DTD للأسئلة ذات الخيارات .

2 أعط مخطط Xschema للأسئلة ذات الخيارات.

3 ا عرض الأسئلة باستخدام .XSLT

4 اعرض نتيجة الاختبار باستخدام .XSLT

5 اعرض الأسئلة باستخدام .Python/DOM

6 اعرض نتيجة الاختبار والعلامة باستخدام .Python/DOM

التمرين 3 (3 نقاط: 0.75*2):

نريد توسيع نظام إدارة الاختيار المتعدد الخاص بنا، ونريد:

1 إنشاء أسئلة MCQ بإجابة واحدة صحيحة، أو عدة إجابات صحيحة

2 عرض الاختبار باللغة العربية أو الأجنبية حسب إعدادات المستخدم.

3 متابعة تقدّم النتيجة أثناء الاختبار.

4 حفظ نسخة من الاختبارات لكل طالب.

أعط نمط التصميم المناسب لكل خيار. برّر جوابك.

Exercise 1 (7 pts: 2, 1, 1, 1, [0 5*4])) Methodology questions

1 Cite some design-patterns and their categories **(0 25 for each correct answer, -0 25 for each incorrect answer), max 2 points**

2 Which design pattern defines the one-to-many dependency between objects?

Observer

Explanation: The observer pattern defines a one-to-many dependency between objects such that when an object changes state, all of its dependent objects are notified.

المراقب.

شرح: يقوم نمط المراقب بتعريف تبعية واحد إلى متعدد بين الكائنات بحيث أنه عندما يغيّر كائن حالته، يتم إعلام كافة الكائنات التابعة له.

3 What is the design pattern that provides a single class that provides the simplified methods required by the client and delegates the call to those methods? Facade

4 What are the basic rules for writing an XML document?

(closing tags, root element, case sensitive, ignoring space, nesting, attribute value in quotes)

5 What is the difference between CDATA and PCDATA?

PCDATA (parsed data) (CDATA not parsee data).

6 What is XML namespace?

Qualified name associated with DTD/Schema location.

7 Are spaces ignored by XML?
Yes.

8 Can we use graphics in XML?
Yes, inserted by link or represented in SVG.

Project: (exercises 2 and 3)
We want to design a system for managing multiple choice questions (MCQ (for the Web.
A question bank contains a multitude of questions. Each question contains the correct solution, it can contain the answer chosen by the user.

Question Bank

1. What is XML
   - A. ○ eXentible Markup Language
   - B. ○ eXpandable Style Language
   - C. ○ eXtensible Stylesheet Language
   - D. ○ eXtra Style Language

Exercise 2 (10 pts [1, 2, 2, 2, 1.5, 1.5]) XML:
Questions :

1 Define a DTD for MCQ.
Example of XML file للتوضيح فقط، غير مطلوب

```
<questionbank>
<module>Méthodes et technologies d'implémentation'</module>
 <question id="1">
 <label>What is XML</label>
    <choice correct="true" chosen="true">
       eXentible Markup Language
    </choice>
    <choice correct="false" chosen="false">
       eXpandable Style Language
    </choice>
    <choice correct="false" chosen="false">
       eXtensible Stylesheet Language
    </choice>
    <choice correct="false" chosen="false">
       eXtra Style Language
    </choice>
 </question>
<question id="2">
 <label>Which option is true about session scope?</label>
    <choice correct="false" chosen="false">
     Objects are accessible only from the page in which they are created
    </choice>
    <choice correct="true" chosen="false">
    Objects are accessible only from the pages which are in same session
    </choice>
    <choice correct="false" chosen="false">
    Objects are accessible only from the pages which are processing
```

```
    the same request
    </choice>
    <choice correct="false" chosen="true">
    Objects are accessible only from the pages which reside
    in same application
    </choice>
 </question>
</questionbank>
```

DTD file

```
<!ELEMENT questionbank (module, question+)>
<!ELEMENT question (label,choice+)>
<!ATTLIST question id CDATA #REQUIRED>
<!ELEMENT label (#PCDATA)>
<!ELEMENT choice (#PCDATA)>
<!ELEMENT module (#PCDATA)>
<!ATTLIST choice correct CDATA #IMPLIED chosen CDATA #IMPLIED>
```

2 Define an XSchema for MCQ.

```
<xs:element name="questionbank">
  <xs:complexType name="QuestionBankType">
     <xs:sequence>
        <xs:element name="module" type="String" />
        <xs:element maxOccurs="unbounded" name="question" type="QuestionType" />
     </xs:sequence>
  </xs:complexType>
   </xs:element>
   <xs:complexType name="QuestionType">
     <xs:sequence>
      <xs:element name="label" type="String" />
      <xs:element maxOccurs="5" name="choice" type="ChoiceType" />
     </xs:sequence>
      <xs:attribute name="id" type="Integer" use="required" />
   </xs:complexType>
   <xs:complexType name="ChoiceType">
      <xs:attribute default="false" name="correct" type="String" />
      <xs:attribute default="false" name="chosen" type="String" />
   </xs:complexType>
</xs:schema>
```

3 Show questions with XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
   <xsl:output method="html" encoding="utf-8"
   doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
   doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" />
   <xsl:template match="/questionbank">
      <h1 align="center">
         Question Bank of
```

```
         <xsl:value-of select="module" />
      </h1>
      <ol>
         <xsl:for-each select="question">
            <li>
               <xsl:value-of select="label" />
               <ol type="A">
                  <xsl:for-each select="choice">
                     <li>
                        <input type="radio" />
                        <xsl:value-of select="." />
                     </li>
                  </xsl:for-each>
               </ol>
            </li>
            <br />
         </xsl:for-each>
      </ol>
   </xsl:template>
</xsl:stylesheet>
```

Example of display للتوضيح فقط، غير مطلوب

Question Bank of Méthodes et tech

1. What is XML
   - A. ○ eXentible Markup Language
   - B. ○ eXpandable Style Language
   - C. ○ eXtensible Stylesheet Language
   - D. ○ eXtra Style Language

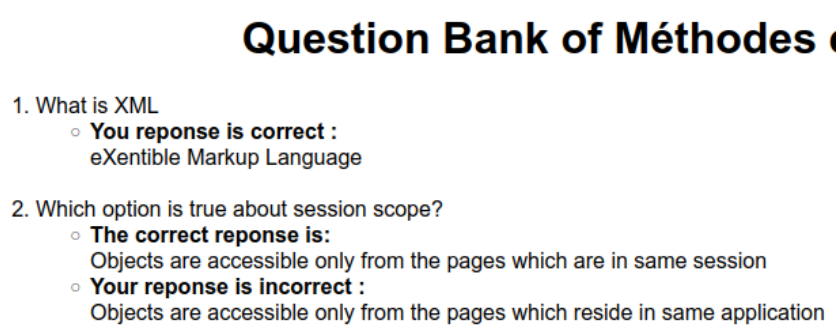
2. Which option is true about session scope?
   - A. ○ Objects are accessible only from the page in which they are created
   - B. ○ Objects are accessible only from the pages which are in same session
   - C. ○ Objects are accessible only from the pages which are processing the same request
   - D. ○ Objects are accessible only from the pages which reside in same application

4 View test result with XSLT.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" encoding="utf-8"
doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" />
<xsl:template match="/questionbank">
<h1 align="center">
 Question Bank of
 <xsl:value-of select="module" />
</h1>
<ol>
 <xsl:for-each select="question">
    <li>
       <xsl:value-of select="label" />
       <ul>
          <xsl:for-each select="choice">
             <xsl:choose>
                <xsl:when test="(@correct='true') and (@chosen='true')">
```

```
                    <li>
                        <strong>You reponse is correct :</strong>
                        <br />
                        <xsl:value-of select="." />
                    </li>
                </xsl:when>
                <xsl:when test="@correct='true'">
                    <li>
                        <strong>The correct reponse is:</strong>
                        <br />
                        <xsl:value-of select="." />
                    </li>
                </xsl:when>
                <xsl:when test="@chosen='true'">
                    <li>
                        <strong>Your reponse is incorrect :</strong>
                        <br />
                        <xsl:value-of select="." />
                    </li>
                </xsl:when>
            </xsl:choose>
        </xsl:for-each>
    </ul>
  </li>
  <br />
 </xsl:for-each>
</ol>
</xsl:template>
</xsl:stylesheet>
```

Example of score display للتوضيح فقط، غير مطلوب



5 Show questions with Python/DOM.

6 Show test result and grade with Python/DOM.

```
mport xml.dom.minidom as minidom
import sys

def show_question(xmldoc):
    """ Show questions """
    module = xmldoc.getElementsByTagName("module")[0].firstChild.data
    questions = xmldoc.getElementsByTagName("question")
    for question in questions:
```

```
        qid = question.getAttribute("id")
        label = question.getElementsByTagName("label")[0].firstChild.data
        choices = question.getElementsByTagName("choice")
        print(qid,"-",label)
        for i,choice in enumerate(choices):
            reponse = choice.firstChild.data
            print("\t", "%d-"%i, reponse)
def show_responses(xmldoc):
    """ Show responses with score """
    score = 0
    module = xmldoc.getElementsByTagName("module")[0].firstChild.data
    questions = xmldoc.getElementsByTagName("question")
    for question in questions:
        qid = question.getAttribute("id")
        label = question.getElementsByTagName("label")[0].firstChild.data
        choices = question.getElementsByTagName("choice")
        print(qid,"-",label)
        # show responses
        for i,choice in enumerate(choices):
            reponse = choice.firstChild.data
            correct = choice.getAttribute("correct")
            chosen = choice.getAttribute("chosen")
            if correct=="true" and chosen=="true":
                print('Your response is correct')
                print("\t", "%d-"%i, reponse)
                score +=1
            elif correct=="true":
                print('The correct answer is ')
                print("\t", i, "-", reponse)
            elif chosen=="true":
                print('Your response is ')
                print("\t", "%d-"%i, reponse)
        print("Score is ", score)
if __name__ == "__main__":
    datafile = "qcm.xml"
    try:
        xmldoc = minidom.parse(datafile)
    except:
        print("Can't open file", datafile)
        sys.exit()
    show_question(xmldoc)
    show_responses(xmldoc)
```

Example of display للتوضيح فقط، غير مطلوب

```
1- What is XML
   0- eXentible Markup Language
   1- eXpandable Style Language
   2- eXtensible Stylesheet Language
   3- eXtra Style Language
2- Which option is true about session scope?
   0- Objects are accessible only from the page in which they are created
```

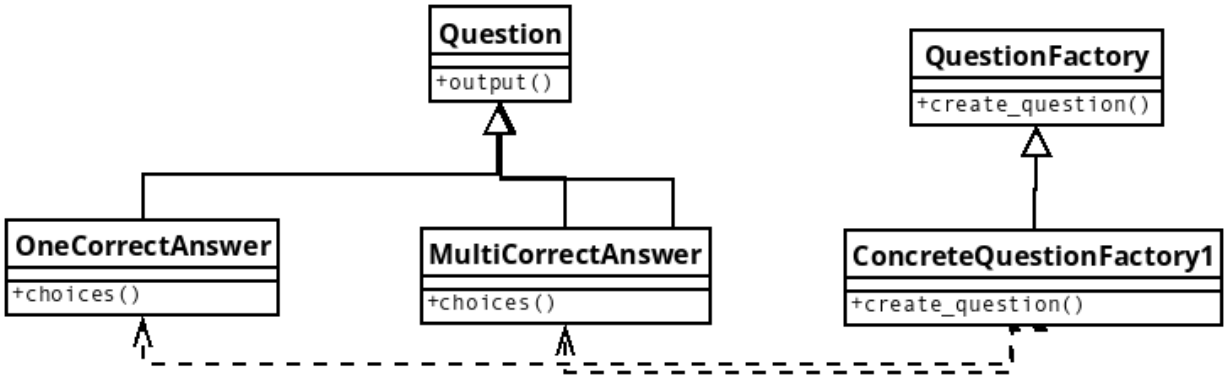
```
    1- Objects are accessible only from the pages which are in same session
    2- Objects are accessible only from the pages which are processing the same request
    3- Objects are accessible only from the pages which reside in same application
```

Example of score display للتوضيح فقط، غير مطلوب

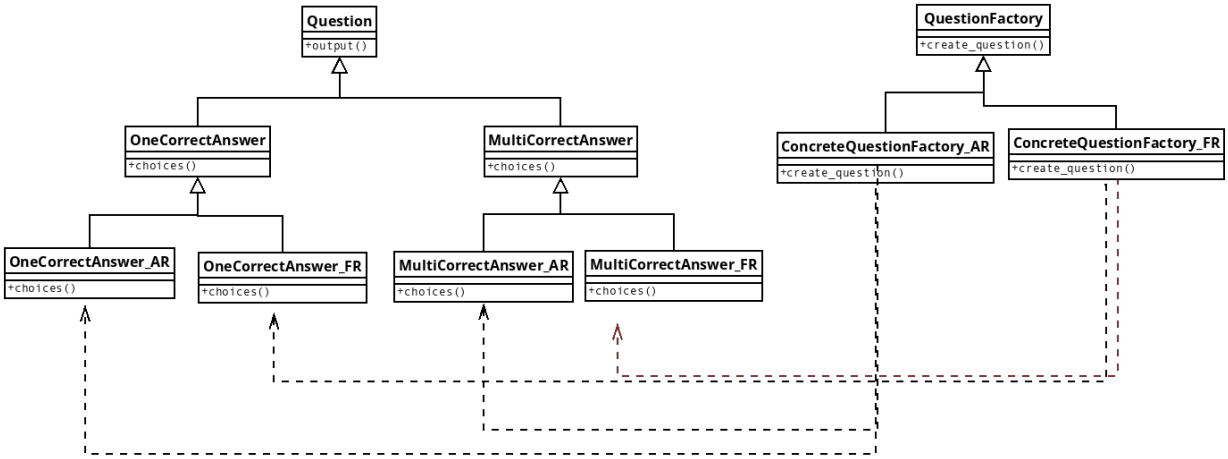
```
1- What is XML
   0- eXentible Markup Language
   1- eXpandable Style Language
   2- eXtensible Stylesheet Language
   3- eXtra Style Language
2- Which option is true about session scope?
   0- Objects are accessible only from the page in which they are created
   1- Objects are accessible only from the pages which are in same session
   2- Objects are accessible only from the pages which are processing the same request
   3- Objects are accessible only from the pages which reside in same application
```

Exercise 3 (3 pts: 0.75*2):
We want to extend our multiple choice management system, we want to:

1 Generate MCQ questions with a single correct answer, or several answers.
Factory method للتوضيح فقط، غير مطلوب



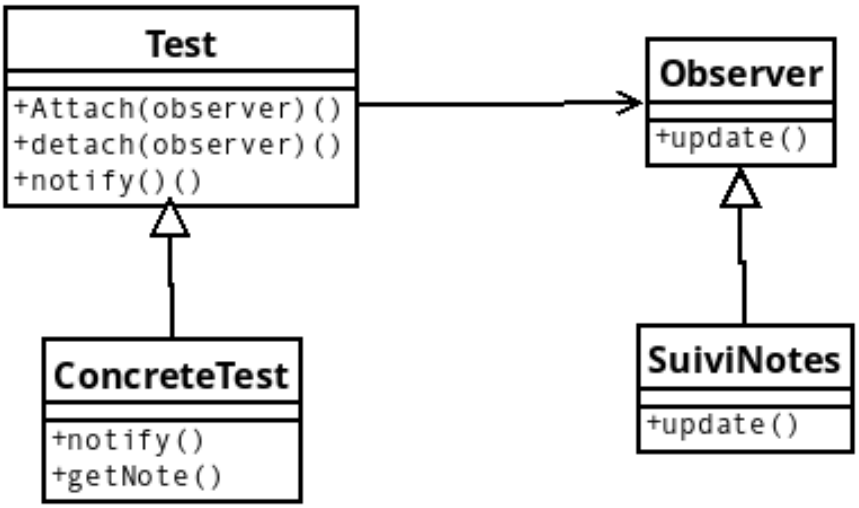
2 Display the test in Arabic or in English depending on the user's configuration.
Abstract Factory
للتوضيح فقط، غير مطلوب



3 Follow the progress of the score during the test.

Observer

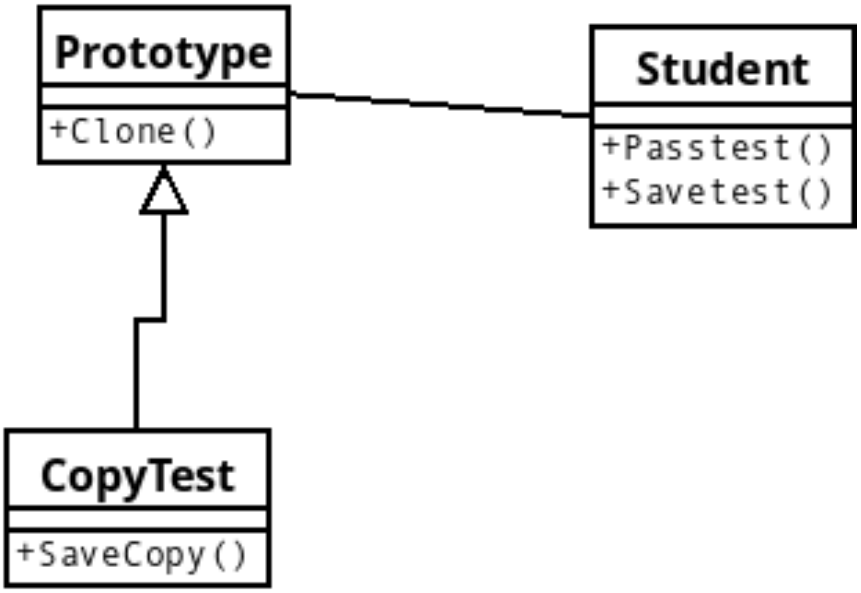
للتوضيح فقط، غير مطلوب



4 Save a copy of the tests for each student.

Prototype

للتوضيح فقط، غير مطلوب



Give the appropriate design pattern for each option. Justify your answer.

Chapter 7

Appendices ملحقات

Appendice A

ملحق أ

هذه قائمة من المراجع والموارد المفيدة لمادة "منهجيات البرمجة وتقنياتها":

7.0.1 Books

كتب

- كتاب نبراس: دليل المصطلحات للشعب التقنية http://nibras.sf.net (Zerrouki, 2012).
- علي آل ياسين، مختصر دليل لغات البرمجة، 2014 (علي آل ياسين, 2016citepalyassen2016)
https://alyassen.github.io/Brief-guide-to-programming-languages-v1.2.4.pdf
- Joseph D. Gradecki, Mastering AspectJ: Aspect-Oriented Programming in Java 1st Edition, Wiley. (Gradecki and Lesiecki, 2003)

7.0.2 Courses online

دروس أونلاين

- أنيس حكمت، دورة Pattern Design باللغة العربية[1](Hikmat, 2021).
- Tutorial points [2] (point, 2022).
- W3Schools Tutorials (**w3schools**): https://www.w3schools.com
- Design Patterns (Arabic) by Passionate Coders محمد المهدي [3]
- Design Patterns بالعربى by DevGeeks Academy [4]
- Design Patterns by Mohammed Reda [5]
- Software Design Patterns by محمد يحيى [6]
- Design Patterns in Object Oriented Programming (English) by Christopher Okhravi [7]

7.0.3 Software

برامج وتطبيقات

[1] https://2nees.com/courses-and-articles/design-pattern
[2] https://www.tutorialspoint.com/
[3] https://www.youtube.com/playlist?list=PLsV97AQt78NTrqUAZM562JbR3ljX19JFR
[4] https://www.youtube.com/playlist?list=PLd-dOEgzBpGnt3GuEszo_piQq52XSqAmj
[5] https://www.youtube.com/@m_reda/
[6] https://www.youtube.com/playlist?list=PLrwRNJX9gLs3oQyBoXtYimY7M5aSFO_oC
[7] https://www.youtube.com/playlist?list=PLrhzvIcii6GNjpARdnO4ueTUAVR9eMBpc

Appendice B

ملحق ب

البرنامج الرسمي المعتمد لمادة منهجيات البرمجة وتقنياتها، المعتمد في قسم الإعلام الآلي، كلية العلوم والعلوم التطبيقية، جامعة البويرة، حسب التعديل الجديد لسنة 2023-2024.

- **MASTER TITLE:** Computer Systems Engineering (GSI)
- **MATERIAL TITLE:** Implementation Method and Technology
- **SEMESTRE:** 1
- **TEACHING UNITY:** Methodology unit
- **CREDIT:** 5
- **COFFICIENT:** 3
- **Evaluation mode :** Exams, practical and continuous monitoring
- **Recommended prior knowledge:** : A programming language (preferably Java)

7.0.4 Educational objectives

Implementation technologies, the web-centric ones, are undergoing a very active evolution. The so-called ”rich” applications are one of the current achievements of this evolution: multimodal interfaces, adaptive, nomadism, 3D, etc. Many technologies exist (Microsoft Silverlight, Sun JavaFX, Adobe Flex/AIR, Curl Platform, AJAX, Javascript, and other frameworks).The objective of this course is to enable the student to appreciate, understand, and master this development in order to use it effectively when designing or creating software.

7.0.5 Content of the material:

- Reminder: Evolution of programming techniques (procedural, modular, object, component)
- The design/implementation bosses:
  - → General Introduction
  - → The most common patterns (factory, adaptation, reverse control, etc.)
- Reflectivity
- The basic concepts of the AOP
- Framework for web development (Spring AOP, JBOSS AOP, Struts 2)
- Rich customer-oriented frameworks (richFaces etc..)

7.0.6 References

1 Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma et al, Addison-Wesley professional computing series, ISBN-13: 978- 0201633610 (Gamma et al., 1995).

2 Coders at Work: Reflections on the Craft of Programming, Peter Seibel, APress 2009,ISBN-13: 978- (Seibel, 2009)

3 Aspect-Oriented Programming with AspectJ, Ivan Kiselev, Sams, 2003, ISBN- 13: 978- (Kiselev, 2003)

7.1 Glossary

مسرد

Accès	بلوغ، وصول، دخول
Acquérir	اكتسب

Bibliography

Freeman, Eric and Elisabeth Robson (2020). *Head First Design Patterns*. O'Reilly Media (cit. on p. 90).

Gamma, E. et al. (1995). *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley professional computing series. Addison-Wesley. ISBN: 9789332555402. URL: https://books.google.dz/books?id=PK7-zgEACAAJ (cit. on p. 114).

Gradecki, Joseph D and Nicholas Lesiecki (2003). *Mastering AspectJ: aspect-oriented programming in Java*. John Wiley & Sons (cit. on pp. 73, 113).

Hikmat, Anis (2021). *Design patterns in Arabic*. URL: https://2nees.com/courses-and-articles/design-pattern (cit. on pp. 49, 59, 113).

Issam, Wajdi (2013). *Model-View-Controller* نمط باستخدام البرمجيات تصميم. https://informatic-ar.com/mvc_java/ (cit. on p. 41).

Kiczales, Gregor et al. (1997). "Aspect-oriented programming." In: *ECOOP'97—Object-Oriented Programming: 11th European Conference Jyväskylä, Finland, June 9–13, 1997 Proceedings 11*. Springer, pp. 220–242. URL: https://apps.dtic.mil/sti/pdfs/ADA417906.pdf (cit. on p. 73).

Kiselev, I. (2003). *Aspect-oriented Programming with AspectJ*. Sams. ISBN: 9780672324109. URL: https://books.google.dz/books?id=K5NQAAAAMAAJ (cit. on p. 114).

Langtangen, Hans Petter (2014). *A primer on scientific programming with Python*. Vol. 6. Springer (cit. on p. 67).

Lee, Ken Wing Kuen (2002). "An introduction to aspect-oriented programming." In: *COMP610E: Course of Software Development of E-Business Applications (Spring 2002), Hong Kong University of Science and Technology*. URL: https://www.inf.utfsm.cl/~mcloud/iwi-253/lecturas/Introduction_to_Aspect_Oriented_Programming.pdf (cit. on p. 73).

Påhlsson, Niklas (2002). "Aspect-oriented programming." In: *Topic Report for Software Engineering*, pp. 11–03. URL: http://oberon2005.oberoncore.ru/paper/np2002.pdf (cit. on p. 73).

Philippe Rigaux, Raphaël Fournier-S'niehotta et (2021). *Applications orientées données (Cours)*. Ed. by Conservatoire National des Arts et Métiers (Cnam) Paris. URL: http://orm.bdpedia.fr/mvc.html (cit. on p. 41).

point, Tutorials (2022). *MVC Framework*. URL: https://www.tutorialspoint.com/mvc_framework/ (cit. on pp. 41, 113).

Seibel, P. (2009). *Coders at Work: Reflections on the Craft of Programming*. IT Pro. Apress. ISBN: 9781430219491. URL: https://books.google.dz/books?id=2kMIqdfyT8kC (cit. on p. 114).

Shvets, Alexander (2018). *Dive Into Design Patterns* (cit. on p. 89).

Svirca, Zanfina (May 2020). *Everything you need to know about MVC architecture*. https://towardsdatascience.com/you-need-to-know-about-mvc-architecture-3c827930b4c1 (cit. on p. 90).

Zerrouki, Taha (2012). *Nibras: Guide des terminologies pour les branches techniques*. Université de Bouira (cit. on pp. 3, 113).

المبرمج العربي (2022). *MVC* نمط التصميم. https://arabicprogrammer.com/article/6434275856/ (cit. on p. 41).

علي آل ياسين (2016). مختصر دليل لغات البرمجة. URL: https://alyassen.github.io/Brief-guide-to-programming-languages-v1.2.4.pdf (cit. on pp. 14, 113).