جامعة الوادي
كلية العلوم الدقيقة

ثالثة إعلام آلي
ساعة ونصف

امتحان هندسة البرمجيات (2026)

التمرين الأول : اشرح مستعينا بمخطط كل من الآتي:
النموذج الحلزوني (Le modèle en spirale), SCRUM , DevOps .

التمرين الثاني :

في إطار ضمان نزاهة وشفافية العملية الانتخابية، ترغب الهيئة الوطنية للانتخابات لتطوير نظام معلوماتي لمراقبة الانتخابات. تمنح الهيئة لكل حزب من الأحزاب السياسية المشاركة برنامج software مستقل بقاعدة معطيات مستقلة، مُدرج بها قائمة مراكز ومكاتب التصويت (لكل ولاية، دائرة وبلدية) بأسماء الأشخاص الإداريين المسؤولون عليها، وقائمة المترشحين. من خلال البرنامج يقوم الحزب برفع وتسجيل كل الملاحظات ونتائج التصويت، ليتم بعدها الحزب بإرسال النتائج النهائية برفع قاعدة معطياته للهيئة.

نريد تصميم البرنامج المعطى للحزب حيث عمله يكون كالآتي :

قبل يوم الانتخاب، يعيّن كل حزب مراقبين تابعين له في مختلف مكاتب التصويت، حيث يتم ادراجهم بحسابات على مستوى المكتب الولائي للحزب. يوم الانتخاب، كل مراقب يقوم بمراقبة عملية التصويت داخل مكتبه, ينقل من خلال البرنامج الملاحظات، نسبة المشاركة وفي الأخير النتيجة النهائية.

كل مركز تصويت يتكون من مكاتب، مرتبط بدائرة انتخابية، يستقبل مراقبين من عدة أحزاب ويحتوي على نتائج انتخابية متعددة (نتيجة لكل حزب). يعتبر النظام أن النتيجة العامة للانتخابات صحيحة إذا كانت النتائج المدخلة من طرف جميع الأحزاب متطابقة، في حالة وجود اختلاف بين نتائج الأحزاب، يقوم النظام بالإشارة إلى وجود خلل و يطلب مراجعة

العمل المطلوب:

1) تصميم الجانب العملياتي (les besoins des acteurs du système) لهذا البرنامج.
2) تصميم لقاعدة بيانات هذا البرنامج, اضف ما يجب اضافته.
3) لو ان كل الاحزاب تتشارك نفس قاعدة البيانات، كيف يصبح شكلها؟
4) نلاحظ ان نسبة المشاركة تكون نفسها عند كل المستعملين، اقترح كيف نبرمجها(جافا)؟
5) نلاحظ ان بنية الادارة يعني ولاية، دائرة، بلدية، مركز ومكتب لها شكل خاص، كيف نحقق هذا(جافا)؟

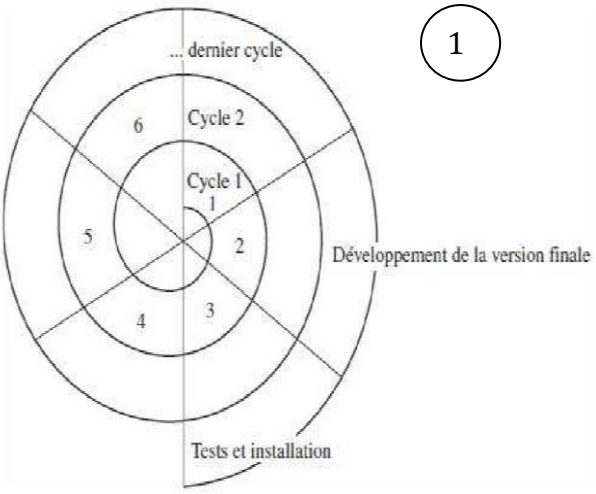
بالتوفيـــق

الحل

التمرين الأول (4):

النموذج الحلزوني **(Le modèle en spirale)** هو أحد نماذج دورة حياة تطوير البرمجيات، يتم تطوير النظام عبر دورات متكررة على شكل حلزوني كالمخطط الآتي

1) Analyse du risque
2) Développement d'un prototype
3) Simulation et essais du prototype ;
4) Détermination des besoins
5) Validation des besoins
6) Planification du cycle suivant



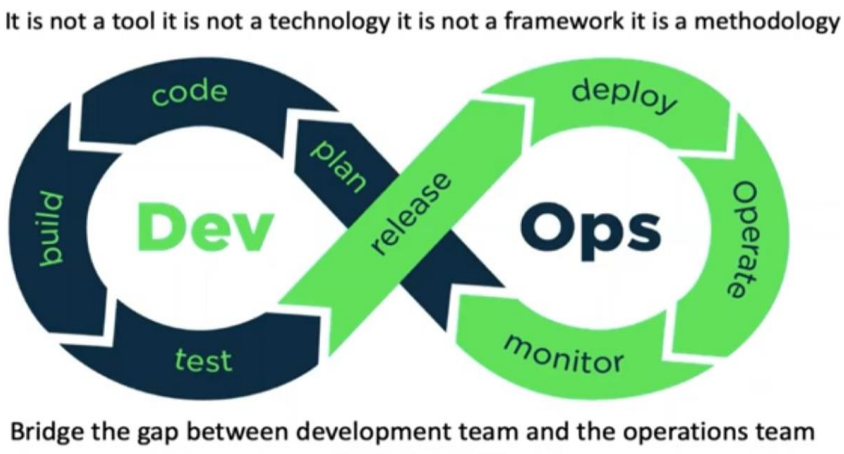
1

SCRUM هو إطار عمل لتطوير البرمجيات وفق المنهجيات الرشيقة(Agile) .

يعتمد في العمل على دورات قصيرة تسمىSprints ، وكل Sprint ينتج جزءًا جاهزًا من المنتج يمكن تسليمه للعميل, يمكن تلخيص طريقة العمل في المخطط الآتي

2

DevOps هو منهجية Methodology وطريقة عمل تهدف إلى ربط فريق التطوير (Development) بفريق التشغيل (Operations)لجعل بناء البرامج وتشغيلها أسرع، أكثر جودة، وأكثر استقرارًا. طريقة العمل ملخصة في المخطط الآتي

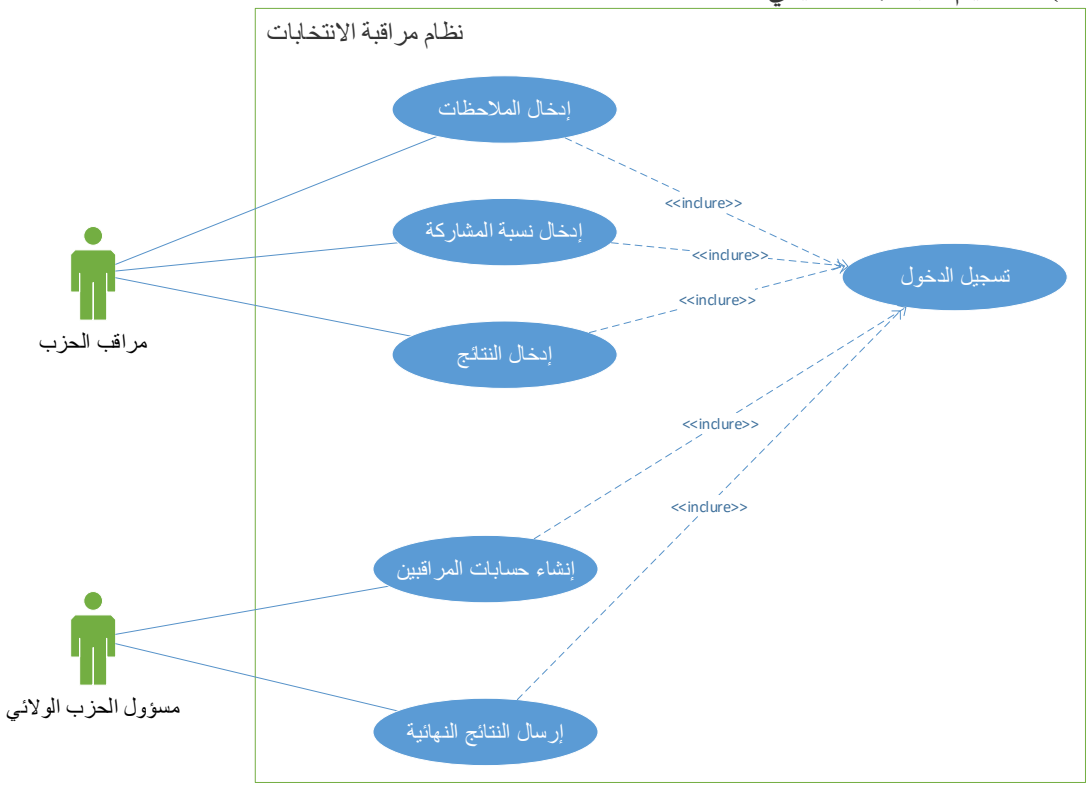


1

التمرين الثاني (16):

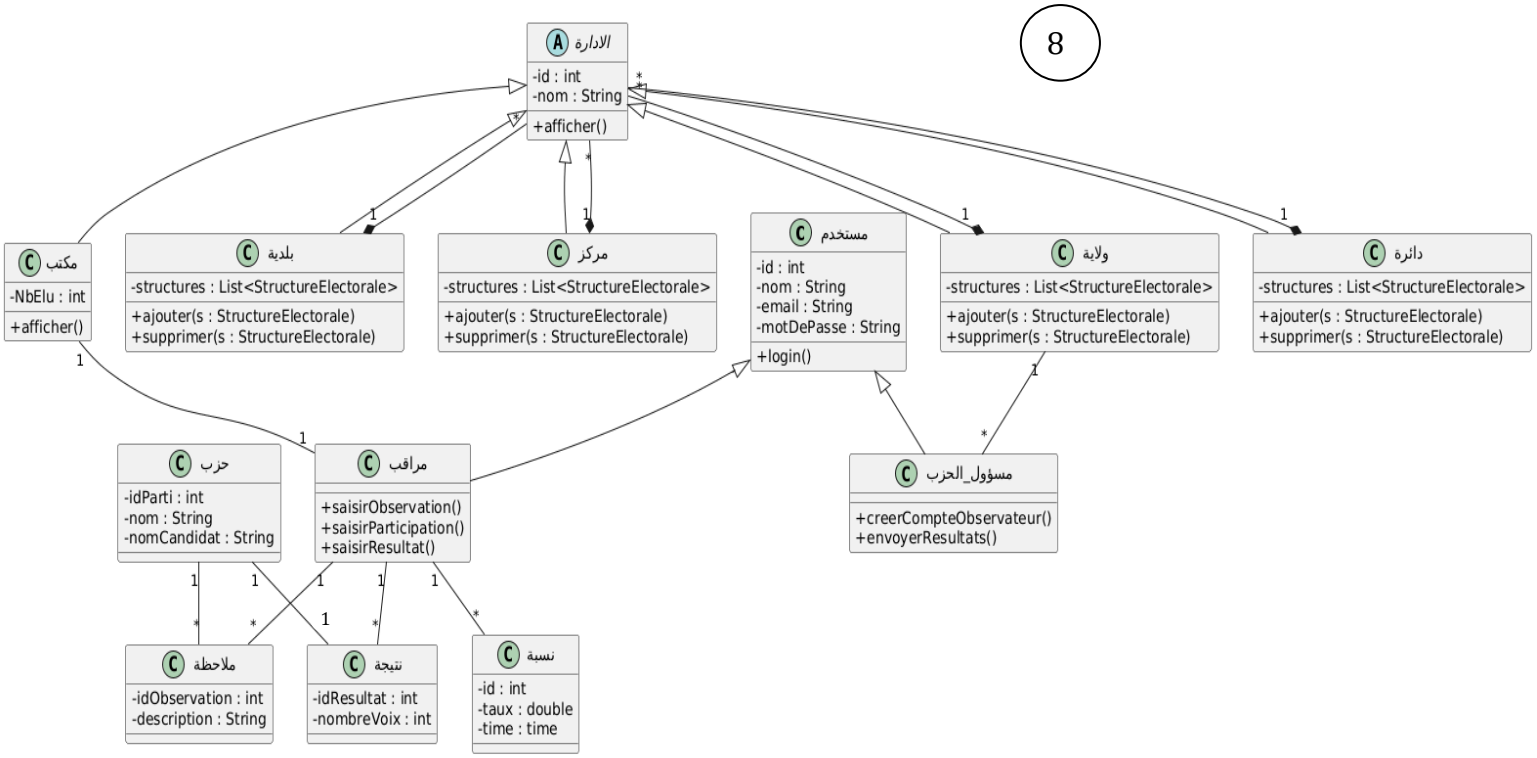
4

1) تصميم الجانب العملياتي

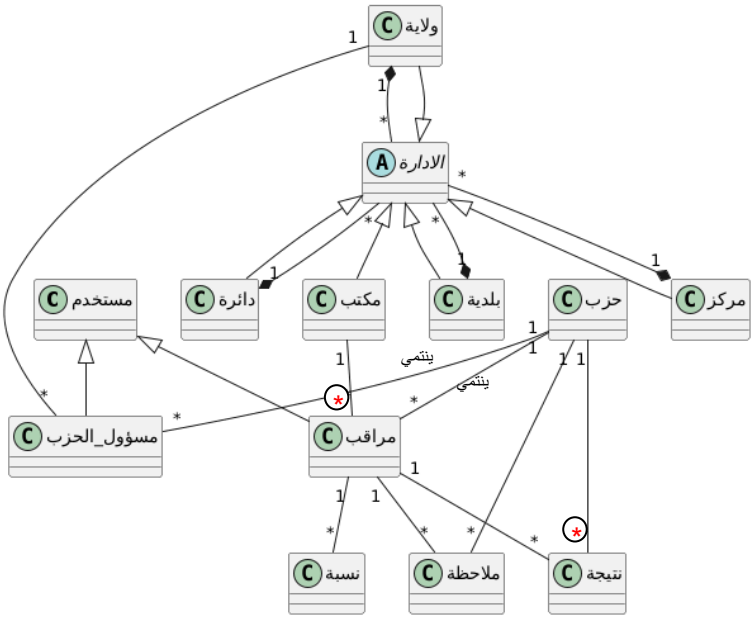


2) تصميم قاعدة بيانات

8



3) قاعدة بيانات يصبح شكلها



1

1) نبرمج نسبة المشاركة باستعمال نمط Singleton

1.5

جافا :

```
public class Participation {
   private static Participation instance;
   private double taux;
   private Participation() {}
   public static Participation getInstance() {
      if (instance == null)  instance = new Participation();
      return instance;
   }
   public double getTaux() { return taux;  }
   public void setTaux(double taux) { this.taux = taux; }
}
```

2) بنية الادارة نحققها باستعمال نمط Composite

1.5

جافا :

```
abstract class الادارة{ protected  int id ; protected String nom;    public abstract void afficher(); }
class ولاية extends الادارة{ List<Structure> dairas;}
class دائرة extends الادارة{ List<Structure> communes;}
class بلدية extends الادارة{List<Structure> centres;}
class مركز extends الادارة{ List<Structure> bureaux;}
class مكتب extends الادارة{
   protected int NbElu ;
   public void afficher() { System.out.println("Bureau: " + nom); }
}
```

```
@startuml
skinparam classAttributeIconSize 0
'====================
' المستخدمون
'====================
class مستخدم{
  -id : int
  -nom : String
  -email : String
  -motDePasse : String
  +login()
}
class مراقب{
  +saisirObservation()
  +saisirParticipation()
  +saisirResultat()
}
class مسؤول_الحزب{
  +creerCompteObservateur()
  +envoyerResultats()
}
مستخدم <|-- مراقب
مستخدم <|-- مسؤول_الحزب
class حزب{
  -idParti : int
  -nom : String
  -nomCandidat : String
}
'====================
' الهيكل الإداري (Composite)
'====================
'========================
' Composite Pattern
'========================
'----- Component -----
abstract class الادارة{
  -id : int
  -nom : String
  +afficher()
}
'----- Composite -----
class ولاية{
  -structures : List<StructureElectorale>
  +ajouter(s : StructureElectorale)
  +supprimer(s : StructureElectorale)
}
class دائرة{
  -structures : List<StructureElectorale>
  +ajouter(s : StructureElectorale)
  +supprimer(s : StructureElectorale)
}
class بلدية{
  -structures : List<StructureElectorale>
  +ajouter(s : StructureElectorale)
  +supprimer(s : StructureElectorale)
}
class مركز{
  -structures : List<StructureElectorale>
  +ajouter(s : StructureElectorale)
  +supprimer(s : StructureElectorale)
}
'----- Leaf -----
class مكتب{
  -NbElu : int
  +afficher()
}
ولاية "1" --* "*" الادارة
دائرة "1" --* "*" الادارة
```

```
بلدية " 1" *-- "*" الادارة
مركز " 1" *-- "*" الادارة
الادارة <|-- ولاية
الادارة <|-- دائرة
الادارة <|-- بلدية
الادارة <|-- مركز
الادارة <|-- مكتب
ولاية " 1" -- "*" مسؤول_الحزب
'====================
الانتخابات '
'====================
class نتيجة {
 -idResultat : int
 -nombreVoix : int
}
class نسبة {
 -id : int
 -taux : double
 -time : time
}
class ملاحظة{
 -idObservation : int
 -description : String
}
حزب " 1" -- "*" نتيجة
حزب " 1" -- "*" ملاحظة
مكتب " 1" -- " 1" مراقب
مراقب " 1" -- "*" نسبة
مراقب " 1" -- "*" ملاحظة
مراقب " 1" -- "*" نتيجة
@enduml

@startuml skinparam classAttributeIconSize
class مستخدم
class مراقب
class مسؤول_الحزب
مستخدم <|-- مراقب
مستخدم <|-- مسؤول_الحزب
class حزب
حزب " 1" -- "*" مراقب
حزب " 1" -- "*" مسؤول_الحزب
abstract class الادارة
class ولاية
class دائرة
class بلدية
class مركز
class مكتب
ولاية " 1" *-- "*" الادارة
دائرة " 1" *-- "*" الادارة
بلدية " 1" *-- "*" الادارة
مركز " 1" *-- "*" الادارة
الادارة <|-- ولاية
الادارة <|-- دائرة
الادارة <|-- بلدية
الادارة <|-- مركز
الادارة <|-- مكتب
ولاية " 1" -- "*" مسؤول_الحزب
class نتيجة
class نسبة
class ملاحظة
مكتب " 1" -- "*" مراقب
حزب " 1" -- "*" نتيجة
حزب " 1" -- "*" ملاحظة
مراقب " 1" -- "*" نسبة
مراقب " 1" -- "*" ملاحظة
مراقب " 1" -- "*" نتيجة
@enduml
```

```
@startuml
left to right direction
actor "مسؤول الحزب الولائي" as Admin
actor "مراقب الحزب" as Observer
rectangle "نظام مراقبة الانتخابات (برنامج الحزب)" {
  (إنشاء حسابات المراقبين) as CreateAccounts
  (إرسال النتائج النهائية) as SendResults
 (إدخال الملاحظات) as Notes
 (إدخال نسبة المشاركة) as Participation
(إدخال النتائج) as Results
Observer -- Notes
Observer -- Participation
Observer -- Results
  Admin -- CreateAccounts
  Admin -- SendResults
  (تسجيل الدخول) as Login
  CreateAccounts .> Login : <<include>>
  SendResults .> Login : <<include>>
  Results .> Login : <<include>>
  Participation .> Login : <<include>>
  Notes .> Login : <<include>>
}
@enduml

https://www.plantuml.com/plantuml/uml
```