Cours M3105 :
IUT Villetaneuse
Conception et programmation objet avancées

Département Informatique
S3, 2020-2021

QCM UML

Q1. Que veut dire UML?

	Union mondiale de la lecture.
	Unified modeling language.
	Unité mesure libre

Q2. L'agrégration est-elle un type d'association? :

	Oui
	Non

Q3. Une composition est-elle un type d'agrégration?

	Oui
	Non

Q4. Que signifie la multiplicité 1..*? :

	Plusieurs incluant la possibilité d'aucun
	Exactement 1
	Au plus un
	Au moins un

Q5. Une action qu'un objet peut réaliser s'appelle :

	Une opération
	Une classe
	Un attribut
	Une formule

Q6. Qu'est ce qu'une relation dite include (A --> B) ?

	Le cas d'utilisation A est réalisé avant le cas d'utilisation B.
	Le cas d'utilisation A a toujours besoin du cas d'utilisation B.
	Le cas d'utilisation A peut éventuellement avoir besoin du cas d'utilisation B.
	Le cas d'utilisation B peut éventuellement avoir besoin du cas d'utilisation A

Q7. Qu'est ce qu'une relation dite extend (B-->A) ?

	Le cas d'utilisation A est réalisé avant le cas d'utilisation B.
	Le cas d'utilisation A a toujours besoin du cas d'utilisation B.
	Le cas d'utilisation A peut éventuellement avoir besoin du cas d'utilisation B.
	Le cas d'utilisation B peut éventuellement avoir besoin du cas d'utilisation A

Q8. Que doit-on faire lors de l'analyse des besoins principaux d'un projet logiciel ?

	Définir toutes les informations nécessaires du futur logiciel.
	Découvrir les acteurs et les fonctionnalités du futur logiciel.
	Décrire une fonctionnalité du futur logiciel.
	Définir les packages ou les grandes parties du logiciel à créer.

Q9. Dans un diagramme de classes en langage UML, la généralisation :

	Est une relation transitive : si C dérive d'une classe B qui dérive elle-même d'une classe A, alors C dérive également de A
	Est une relation réflexive : une classe peut dériver d'elle-même
	Est une relation symétrique : si une classe B dérive d'une classe A, alors la classe A peut dériver de la classe B
	Représente une association non symétrique dans laquelle une des extrémités joue un rôle prédominant par rapport à l'autre extrémité

Q10. En Java, une association (dans un diagramme de classes UML) est-elle implémentée par ?

	une variable d'instance
	une opération
	une variable de classe
	un constructeur

Q11. Un rôle (dans un diagramme de classes UML) se traduit-il en Java par ?

	un nom de variable
	une association
	un constructeur
	une opération

Q12. Des véhicules sont dotés de châssis et de propulsions. Différents types de châssis permettent aux véhicules de rouler, voler ou bien flotter. Différents types de propulsions leur permettent d'avancer grâce au vent ou bien grâce à un moteur. Toutes les combinaisons de véhicules sont possibles : une voiture roule avec un moteur ; un planeur vole avec du vent etc. Au minimum, combien de classes sont nécessaires pour représenter les véhicules avec toutes leurs déclinaisons, si on n'utilise **que l'héritage** ? (en incluant la classe Véhicule)

	7
	8
	9
	10

Q13. Même problème que la question précédente, mais **en utilisant des interfaces** combien de classes/interfaces sont nécessaires pour représenter les véhicules avec toutes leurs déclinaisons (en incluant la classe Véhicule) ?

	7
	8
	9
	10

Q14. Quel(s) type(s) de relations sont permises entre des cas d'utilisation ? (plusieurs choix possibles)

	Association
	Généralisation / héritage
	Dépendance de type "include"
	Dépendance de type "extend"
	Aucune de ces réponses n'est correcte.

Q15. Quel est le modificateur d'accès le plus contraignant ?

	private
	public
	protected

Q16. Entre une classe Véhicule et une classe Roue, quel type de relation est adéquat ?

	Composition
	Association
	Héritage
	Agrégation

Q17. Entre une classe Véhicule et une classe Conducteur, quel type de relation est adéquat ?

	Composition
	Association
	Héritage
	Agrégation

Q18. Entre une Classe Véhicule et une classe Bateau, quel type de relation est adéquat ?

	Composition
	Association
	Héritage
	Agrégation

Q19. Considérons une association entre une classe Client et une classe Commande. Quelle multiplicité mettriez-vous du côté de Commande ?

	0..1
	0..*
	1..*
	1..1

Q20. Considérons une association entre une classe Client et une classe Commande. Quelle multiplicité mettriez-vous du côté du Client ?

	0..1
	0..*
	1..*
	1..1

Q21. Grâce à une relation d'héritage, de quoi hérite la classe enfant ? (plusieurs choix possibles)

	des opérations
	des propriétés
	des associations
	des relations d'héritage
	des attributs

QCM JAVA

Q1. Combien d'instances de la classe A crée le code suivant ?

```
A x,u,v;
x=new A();
A y=x;
A z=new A();
```

	Aucune
	Cinq
	Trois
	Deux

Q2. Pour la classe B définie comme suit:

```
class B{
        public B(){
                System.out.print(”Ciao”);
        }

        public B(int i){
                this(); System.out.println(”Bonjour ”+i);
        }

}
```

qu'affichera l'instruction suivante?

```
B monB=new B(2003);
```

	erreur de compilation
	erreur d'exécution
	CiaoBonjour 2003
	Bonjour 2003

Q3. Cochez la bonne assertion.

	Une classe peut implémenter plusieurs interfaces mais doit étendre une seule classe
	Une classe peut étendre plusieurs classes mais ne peut étendre qu'une seule interface
	Une classe peut implémenter plusieurs classes et peut étendre plusieurs interfaces
	Une classe peut implémenter une seule interface et ne peut étendre qu'une seule classe

Q4. Etant donnée que la classe Grande étend la classe Petite, trouvez une ligne correcte parmi les suivantes :

	Petite y =new Petite(); Grande x= (Grande)y; Petite z=x;
	Grande x= new Grande(); Petite y = x; Grande z=(Grande)y;
	Grande x= new Grande(); Petite y = x; Grande z=y;
	Petite y =new Petite(); Grande x= (Grande)y; Petite z=(Petite)x;

Q5. Pour la classe C définie comme suit :

```
class C{
        public static int i=0;
        private int j;
        public C(){
                i++; j=i;
        }
}
```

qu'affichera le code suivant ?

```
public static void main(String[] args){
        C x=new C(); C y=new C(); C z= x;
        System.out.println(z.i + " et " + z.j);
}
```

	2 et 2
	1 et 1
	2 et 1
	1 et 3

Q6. On se donne les classes et interfaces suivantes :

```
interface Propulsion{
        public void bouger() ;
}
```

Q6.1 Quel est ou quels sont les codes qui provoquent une ou plusieurs erreurs :

	public class Moteur implements Propulsion {
	public void bouger(){
	// faire quelque chose...
	}
	}

	public class Moteur implements Propulsion{
	public void ronronner(){
	// faire quelque chose...
	}
	}

Q6.2 on suppose la classe Moteur correcte et on ajoute la classe suivante (supposée correcte) :

```
class Vent implements Propulsion{
        //....
}
```

Quel est ou quels sont les codes qui provoquent une ou plusieurs erreurs (0, 1 ou plusieurs réponses possibles) :

a)	public class Véhicule{
	private Propulsion p ;
	public void setPropulsion(){p = new Propulsion() ;}
	}

b)	public class Véhicule{
	private Propulsion p ;
	public void setPropulsion(){p = new Moteur() ;}
	}

c)	public class Avion extends Véhicule{
	public Avion(){setPropulsion(new Moteur()) ;}
	}

d)	public class Avion extends Véhicule{
	public Avion(){p=new Vent() ;}
	}

e)	public class Avion extends Véhicule{
	public Avion(){setPropulsion(new Propulsion()) ;}
	}

g)	public class Avion extends Véhicule{
	public Avion(){p=new Moteur() ;}
	}

h)	public class Avion extends Véhicule{
	public Avion(){setPropulsion(new Vent()) ;}
	}