Université de Skikda
Faculté de science
Département d'informatique
Module : POO

Année : 2023-2024
Niveau : 2année licence informatique
Durée : 01h 30m

CORRIGE-TYPE EXAMEN

Exo 01 : (04 pt)

I - 1- Corriger les erreurs dans la classe suivante en respectant les conventions et les principes de la programmation orientée objet. **(02 pt)**

```
public class Module {
 private String  nom ;
private  int coefficient  ;
public Module (String nom,  int coefficient)  {
   this.nom = nom ;
   this.coefficient = coefficient ;
 }
public  void  setNom(String nom) {this.nom = nom  ;  }
}
```

II- (01 pt)

3- (01 pt)

```
int n = Integer.parseInt("25") ;
```

III –

4- (01pt)

```
public interface Comparable
{
boolean estEgale(Oject autre) ;
boolean estPlusGrand(Oject autre) ;
boolean estpetit(Oject autre) ;

}
```

Exo 02 : (16 pt)

```
// I-
// 1- (02 pts)
public class Tache {
      protected int numero  ;
      protected String nom  ;
      protected int duree  ;
      protected String description  ;
      public Tache(int numero, String nom, int duree) {
            this.numero = numero;
            this.nom = nom;
            this.duree = duree;
      }

      public Tache(int numero, String nom) {
```

```java
            this.numero = numero;
            this.nom = nom;
      }

      //2- (01 pt)
      public String getDescription() {
            return description;
      }
      public void setDescription(String description) {
            this.description = description;
      }

// 3- (01 pt)
      public void afficher ()
      {
            System.out.println("la tâche numero " + numero  + "intitulé " + nom  );
      }

      // 4-    (01 pt)
      public boolean estLongue ()
      {
            if (duree > 10 )
                  return true ;
            else
                  return false ;
            // ou bien :  return duree > 10 ;
      }
      //5-   (01 pt)
      public void ettendre (int val )
      {
            duree += val  ;
      }

      //6- (01 pt)
      public void ettendre ()
      {
            duree += 1  ;
      }
      //7- (01 pt)
      public boolean estPlusCourte (Tache t)
      {
            if (duree < t.duree )
                  return true ;
            else
                  return false ;
            // ou bien   : return  duree < t.duree ;
      }

}
```

```java
// II-
  //8-  (01 pt)
public class TacheCollective extends Tache {
     protected int nbrTravailleurs  ;

     public TacheCollective(int numero, String nom, int duree, int
nbrTravailleurs) {
          super(numero, nom, duree);
          this.nbrTravailleurs = nbrTravailleurs;
     }
     public TacheCollective(int numero, String nom, int nbrTravailleurs) {
          super(numero, nom);
          this.nbrTravailleurs = nbrTravailleurs;
     }
     }

//III-
//9- (01 pt)
 public class Projet {
     protected String intitule  ;
     ArrayList< Tache > taches  = new ArrayList<Tache>() ;
     public Projet(String intitule) {
          this.intitule = intitule;
     }
     // - 10- (01 pt )
     public int nbrTaches ()
     {
          return taches.size()  ;
     }
     //- 11- (01 pt)
     public void ajouter (Tache t)
     {
          taches.add(t) ;
     }
 }

//   ou bien
//- III-

//9- (01 pt)
 public class Projet {
     protected String intitule  ;
     protected Tache [] taches  = new Tache [100] ;
     protected int nbrTaches = 0 ;

     public Projet(String intitule) {
          this.intitule = intitule;
     }

     //- 10 (01 pt)

     public int nbrTaches ()
     {
          return nbrTaches ;
```

```
	}

	//- 11- (01 pt)
	public void ajouter (Tache t)
	{
		if (nbrTaches < 100)
		{
			taches[nbrTaches] = t  ;
			nbrTaches++ ;
		}
		else
		{
		  System.out.println("erreur: on peut pas avoir plus de 100 taches" );
		}
	}
   }


//-IV

public class Prog {

	public static void main(String[] args) {

		//- 12 (01 pt)
		Tache t1 = new Tache(01, "planification") ;

		//- 13 (0.5 pt)
		TacheCollective t2 = new TacheCollective(03, "construction", 10);

		//- 14 (0.5 pt)
		Projet p1 = new Projet("construction de 100 logements")  ;

		// 15- (01 pt)
		if (t1.estPlusCourte(t2))
		{
			t1.afficher ();
		}
		else
		{
			t2.afficher();
		}


	//-16	(01 pt)
		p1.ajouter(t1);
		p1.ajouter(t2);
	}
}
```

I- Raliser la classe **Tache** permet de représenter une Tâche. Chaque tâche se caractérise par : son numéro **numero** , son nom ***nom*** , sa durée duree **,** et sa description **description** .

1- Lors de la création de cette Tâche : **(02 pt)**
- Soit on indique son numéro, son nom et sa durée.
- Soit on indique **que** le numéro et le nom**.**

2- Ajouter un **getter** et un **setter** pour l'attribut **description**. **(01 pt)**

3- Ajouter la méthode **detail()** permet d'afficher le détail de cette tâche . **(01 pt)**

4- Ajouter la méthode **estLongue ()** permet d'indiquer est ce que cette tâche est une tache de longue durée ou non. Sachant qu'une tâche de longue durée c'est une tâche qui dépasse 10 jours. **(01 pt)**

5- Ecrire la méthode **etendre ()** permet d'étendre (augmenter) la durée de cette tâche par un nombre précis de jours. **(01 pt)**

6- surcharger la méthode **etendre()** afin de préciser la nouvelle durée. **(01 pt)**

7- Ecrire la méthode **estPlusCourte ()** permet d'indiquer est ce que cette tâche est plus courte qu'une autre tâche. **(01 pt)**

II-

8- Réaliser la classe ***TacheCollective*** une sous classe de la classe Tache qui se caractérise en plus par le nombre de travailleurs ***nbrTravailleur*** . **(01 pt)**

III- Réaliser la classe ***Projet*** qui représente un projet chaque projet se caractérise par son intitulé **intitule**. Il est constitué d'un ensemble de tâches.

9- Lors de la création d'un projet ; on doit indiquer son intitulé**. (01 pt)**

10- Ecrire la méthode ***nbrTaches()*** permet de calculer le nombre de ses tâches. **(01 pt)**

11- Ecrire la méthode **ajouterTache()** permet d'ajouter une tâche à ce projet. **(01 pt)**

III- dans un programme :

12- créer la tâche numéro **01** intitulé « **planification** ». **(01 pt)**

13- créer la tâche numero **03** intituler « **construction** » réaliser par **10** personne. **(0.5 pt)**

14- créer un projet intitulé « **construction de 100 logement** ». **(0.5 pt)**

15- tester et afficher la tâche la plus courte entre ces deux tâches**. (01 pt)**

16- ajouter ces deux tâches au projet. **(01 pt)**

Bon courage