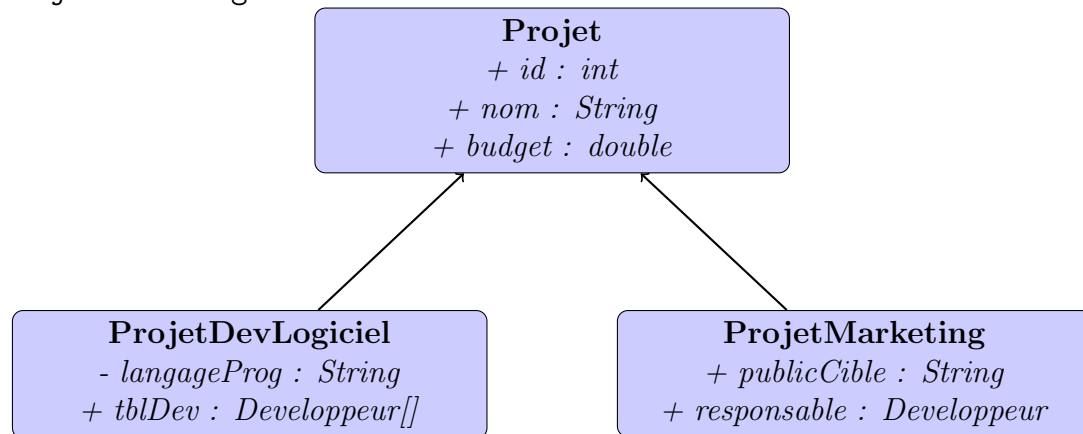


CC 23

Exercice 2

1 Schéma illustrant la relation d'héritage entre les classes

Le diagramme suivant montre la relation d'héritage entre les classes `Projet`, `ProjetDevLogiciel`, et `ProjetMarketing`.



Remarque

Dans le CC, seul l'attribut `langageProg` est demandé de le déclarer comme privé. Dans ce cas, il faut ajouter le getter et le setter pour cet attribut. Le CC n'oblige pas de spécifier un niveau de visibilité pour les autres attributs, donc on peut simplifier et les mettre en public. Cela signifie qu'il n'y a pas besoin de getters et setters pour ces attributs.

2 Classe Projet

```
package entity;

public class Projet {
    public int id;
    public String nom;
    public double budget;

    public static int compteur = 1;

    public Projet(String nom, double budget) {
        this.id = compteur++;
        this.nom = nom;
    }
}
```

```

        this.budget = budget;
    }

    @Override
    public String toString() {
        return "Projet [id=" + id + ", nom=" + nom + ", budget=" +
            budget + "]";
    }
}

```

3 Classe ProjetDevLogiciel

```

package entity;

import java.util.Arrays;

public class ProjetDevLogiciel extends Projet {
    private String langageProg;
    public Developpeur[] tblDev;

    public ProjetDevLogiciel(String nom, double budget, String
        langageProg, Developpeur[] tblDev) {
        super(nom, budget);
        this.langageProg = langageProg;
        this.tblDev = tblDev;
    }

    public String getLangageProg() {
        return langageProg;
    }

    public void setLangageProg(String langageProg) {
        this.langageProg = langageProg;
    }

    @Override
    public String toString() {
        return super.toString() + ", langageProg=" + langageProg + ", d
            éveloppeurs=" + Arrays.toString(tblDev);
    }
}

```

Explications : La classe `ProjetDevLogiciel` hérite de `Projet` et ajoute des informations spécifiques au développement logiciel, telles que le langage de programmation et une liste de développeurs associés.

4 Méthode main pour la création d'un ProjetDevLogiciel

```

package test;

import entity.*; // importer toutes les classes dans le package entity

```

```

public class Test {
    public static void main(String[] args) {
        // Création de développeurs
        Developpeur dev1 = new Developpeur(1, "Ahmed");
        Developpeur dev2 = new Developpeur(2, "Ayoub");

        // Initialisation du tableau des développeurs pour le premier
        projet
        Developpeur[] tblDev1 = new Developpeur[4];
        tblDev1[0] = dev1;
        tblDev1[1] = dev2;

        // Création d'un projet de développement logiciel avec des dé
        veloppeurs
        ProjetDevLogiciel projetDev1 = new ProjetDevLogiciel("
            Application Web", 50000, "Java", tblDev1);

        // Affichage du projet
        System.out.println(projetDev1);
    }
}

```

Explications : La méthode `main` crée deux développeurs et un projet de développement logiciel avec une liste de développeurs pré-initialisée.

Exercice 3

1 Méthode `trierProjets` : Tri des projets par ordre croissant du budget

```

public ProjetDevLogiciel[] trierProjets(ProjetDevLogiciel[] tProjDev) {
    for (int i = 0; i < tProjDev.length - 1; i++) {
        for (int j = i + 1; j < tProjDev.length; j++) {
            if (tProjDev[i] != null && tProjDev[j] != null && tProjDev[
                i].budget > tProjDev[j].budget) {
                ProjetDevLogiciel temp = tProjDev[i];
                tProjDev[i] = tProjDev[j];
                tProjDev[j] = temp;
            }
        }
    }
    return tProjDev;
}

```

2 Méthode countDev : Compter le nombre de développeurs affectés

```
public int countDev() {  
    int count = 0;  
    for (int i = 0; i < tblDev.length; i++) {  
        if (tblDev[i] != null) count++;  
    }  
    return count;  
}
```

3 Méthode suppProjetVide : Supprimer les projets sans développeurs

```
public ProjetDevLogiciel[] suppProjetVide(ProjetDevLogiciel[] tProjDev)  
{  
    for (int i = 0; i < tProjDev.length; i++) {  
        if (tProjDev[i] != null && tProjDev[i].countDev() == 0) {  
            tProjDev[i] = null;  
        }  
    }  
    return tProjDev;  
}
```

Lien vers le code

Vous pouvez consulter le code sur GitHub en suivant ce lien : [Code source sur GitHub](#).