ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR

Membre de HONORIS UNITED UNIVERSITIES

CLASSES ABSTRAITES

3IIR - POO2

CLASSE ABSTRAITE

- Une méthode abstraite est une méthode déclarée avec le mot clé **abstract**. C'est **une méthode sans code.**
  - ***L'implémentation d'une telle méthode est la responsabilité des sous-classes.***
- ***Toute classe qui contient au moins une méthode abstraite est une classe abstraite***. Elle doit être déclarée avec le mot clé **abstract.**

```
public abstract class Oiseau
  {
    public abstract void parler();
    public abstract void voler();
  }
```

CLASSE ABSTRAITE

- **On utilise les classes abstraites:**
  - Lorsqu'on souhaite créer une superclasse qui fournit une implémentation généralisée partagée par toutes ses sous-classes tout en garantissant que seules les sous-classes seront instanciées.
    - **La classe abstraite** définit **un type général** qui structure l'héritage, mais qui n' a pas de sens en tant qu'objet concret.
  - Lorsqu'une superclasse ne peut pas fournir une implémentation significative pour une méthode et que celle-ci doit impérativement être implémentée par la sous-classe pour que cette dernière ait un sens.

CLASSE ABSTRAITE

- **On ne peut pas créer d'objets d'une classe abstraite**. De tels objets seraient inutiles, car une classe abstraite n'est que partiellement définie.

```
public static void main(String[] args)
   {
      Oiseau O = new Oiseau(); // Erreur classe abstraite
   }
```

- Une classe abstraite est utilisée pour fournir une implémentation partielle ou commune pour ses sous classes.

CLASSE ABSTRAITE

- Toute sous-classe d'une classe abstraite doit implémenter toutes les méthodes abstraites de sa superclasse. Sinon, la sous classe doit être déclarée aussi abstract.

```
public class Pigeon  extends  Oiseau
{
   @Override
   public void parler()
         {
           System.out.println("coo! coo!");
         }
   @Override
   public void voler()
         {
           System.out.println("je suis un voyageur");
         }
}
```

CLASSE ABSTRAITE

- Bien que les classes abstraites ne puissent pas être utilisées pour instancier des objets, elles peuvent être utilisées pour déclarer des références d'objets.

```
public class Main
{
   public static void main(String[] args)
   {
      Oiseau P = new Pigeon(); //up-casting
      P.parler();
      P.voler();

   }
}
```

En Java, le polymorphisme dynamique est implémenté par l'utilisation de références de superclasses.

CLASSE ABSTRAITE : EXEMPLE

- Etant donné qu'on ne peut pas calculer le périmètre ni l'aire d'une forme bidimensionnelle non définie, la classe **Forme** suivante déclare **perimetre()** et **aire()** comme méthodes abstraites.

```
public abstract class Forme2D
{
protected double dim1;
protected double dim2;
public Forme2D(double a, double b) {
dim1 = a; dim2 = b;
}
public abstract double perimetre();
public abstract double aire();
}
```

CLASSE ABSTRAITE : EXEMPLE

- Toutes les sous classes de **Forme** doivent redéfinir **perimetre() et aire()**

```
//Triangle Rectangle
public class Triangle extends Forme2D
{
  public Triangle(double b, double h )
  {
    super(b,h);
  }
  @Override
  public double perimetre()
  {
   double h= Math.sqrt(dim1 * dim1 + dim2 * dim2);
   return dim1 + dim2 + h;
  }
 @Override
  public double aire()
  {
    return dim1 * dim2 / 2;
  }
}
```

```
public class Rectangle extends Forme2D
{
 public Rectangle(double l, double h)
  {
     super(l,h);
  }
 @Override
  public double perimetre()
  {
    return 2 *  (dim1  + dim2 );
  }
 @Override
  public double aire()
  {
    return dim1 * dim2;
  }
}
```

CLASSE ABSTRAITE : EXEMPLE

```
public class EssaiFormes
{
  public static void main(String[] argv)
  {
      Rectangle R = new Rectangle(9, 5);

      Triangle T = new Triangle(10, 8);

      Forme2D F ; // OK, on peut déclarer une reference de type Forme

      F = R;

      System.out.println("Aire = " + F.aire());

  }
}
```

EXERCICE

- **Vrai ou Faux ?**
  - Vous ne pouvez pas étendre une classe concrète par une classe abstraite.
  - Vous ne pouvez pas étendre une classe abstraite par une autre classe abstraite.
  - Vous ne pouvez pas instancier une classe abstraite, mais vous pouvez instancier une sous-classe concrète d'une classe abstraite.
  - Une méthode abstraite peut avoir une implémentation dans une classe abstraite.
  - Toutes les méthodes d'une classe abstraite doivent être abstraites.
  - Une classe abstraite peut avoir à la fois des méthodes abstraites et des méthodes concrètes.

EXERCICE

- Choisissez la bonne réponse en fonction de la définition de classe suivante :

```
public abstract final class Forme { }
```

A. Erreur : une classe abstraite ne doit pas être vide.

B. Erreur : une classe abstraite ne peut pas être déclarée final.

C. Erreur : une classe abstraite doit comporter au moins une méthode abstraite.

D. Cette définition de classe est correcte et sera compilée avec succès.

EXERCICE

- Choisissez la bonne réponse en fonction de la définition de classe suivante :

```
public abstract class Forme
{
  public abstract final void draw();
  private abstract void rouler();
}
```

A. Erreur : une méthode abstraite ne peut pas être déclarée **private**.

B. Erreur : une méthode abstract ne peut pas être déclarée **final**.

C. Cette définition de classe est correcte et sera compilée avec succès.