

Règles examen Java

Voici la correction complète et commentée des exercices du fichier TP_Generals.pdf :



Exercice 1 : Interface et polymorphisme

♦ Étape 1 : Interface `Animal`

java

```
public interface Animal {  
    void crier();  
    String regimeAlimentaire();  
}
```

♦ Étape 2 : Classes implémentant `Animal`

✓ Classe `Chien`

java

```
public class Chien implements Animal {  
    public void crier() {  
        System.out.println("Aboyer");  
    }  
  
    public String regimeAlimentaire() {  
        return "Omnivore";  
    }  
}
```

✓ Classe `Chat`

java

```
public class Chat implements Animal {  
    public void crier() {  
        System.out.println("Miauler");  
    }  
  
    public String regimeAlimentaire() {  
        return "Carnivore";  
    }  
}
```

✓ Classe Vache

java

```
public class Vache implements Animal {  
    public void crier() {  
        System.out.println("Meugler");  
    }  
  
    public String regimeAlimentaire() {  
        return "Herbivore";  
    }  
}
```

♦ Étape 3 : Classe principale Zoo

java

```
public class Zoo {  
    public static void main(String[] args) {  
        Animal[] animaux = { new Chien(), new Chat(), new Vache() };  
  
        for (Animal a : animaux) {  
            a.crier();  
            System.out.println("Régime : " + a.regimeAlimentaire());  
        }  
}
```

```
}  
}
```

Exercice 2 : Exceptions personnalisées

♦ Étape 1 : Définir les exceptions

✓ Exception ErrAge

```
java  
  
public class ErrAge extends Exception {  
    public ErrAge(String message) {  
        super(message);  
    }  
}
```

✓ Exception ErrVieillessement

```
java  
  
public class ErrVieillessement extends Exception {  
    public ErrVieillessement(String message) {  
        super(message);  
    }  
}
```

♦ Étape 2 : Classe Animal

```
java  
  
public class Animal {  
    private String nom;  
    private int age;  
  
    public Animal(String nom, int age) throws ErrAge {
```

```

        if (age < 0) {
            throw new ErrAge("Âge négatif non autorisé.");
        }
        this.nom = nom;
        this.age = age;
    }

    public void vieillir(int annees) throws ErrVieillissement {
        if (age + annees > 150) {
            throw new ErrVieillissement("Âge dépassé (150 ans max).");
        }
        age += annees;
    }

    public int getAge() {
        return age;
    }
}

```

♦ Étape 3 : Classe TestAnimal

java

```

public class TestAnimal {
    public static void main(String[] args) {
        try {
            Animal a1 = new Animal("Tigre", 10);
            a1.vieillir(20);
            System.out.println("Nouvel âge : " + a1.getAge());

            Animal a2 = new Animal("Éléphant", 100);
            a2.vieillir(60); // Provoque une exception
        } catch (ErrAge e) {
            System.out.println("Erreur d'âge : " + e.getMessage());
        } catch (ErrVieillissement e) {
            System.out.println("Erreur de vieillissement : " + e.getMessage());
        }
    }
}

```

```
}  
}
```

Exercice 3 : Interfaces & packages

Structure recommandée du projet :

markdown

```
interfaces/  
└─ animaux/  
    ├── Animal.java  
    ├── mammiferes/  
    │   ├── Chien.java  
    │   └─ Chat.java  
    └─ herbivores/  
        └─ Vache.java
```

♦ interfaces/animaux/Animal.java

java

```
package interfaces.animaux;  
  
public interface Animal {  
    void crier();  
    String regimeAlimentaire();  
}
```

♦ interfaces/animaux/mammiferes/Chien.java

java

```
package interfaces.animaux.mammiferes;  
import interfaces.animaux.Animal;  
  
public class Chien implements Animal {  
    public void crier() {
```

```
        System.out.println("Aboyer");
    }

    public String regimeAlimentaire() {
        return "Omnivore";
    }
}
```

◆ interfaces/animaux/mammiferes/Chat.java

java

```
package interfaces.animaux.mammiferes;
import interfaces.animaux.Animal;

public class Chat implements Animal {
    public void crier() {
        System.out.println("Miauler");
    }

    public String regimeAlimentaire() {
        return "Carnivore";
    }
}
```

◆ interfaces/animaux/herbivores/Vache.java

java

```
package interfaces.animaux.herbivores;
import interfaces.animaux.Animal;

public class Vache implements Animal {
    public void crier() {
        System.out.println("Meugler");
    }

    public String regimeAlimentaire() {
        return "Herbivore";
    }
}
```

♦ Zoo.java (package principal)

java

```
import interfaces.animaux.Animal;
import interfaces.animaux.mammiferes.Chien;
import interfaces.animaux.mammiferes.Chat;
import interfaces.animaux.herbivores.Vache;

public class Zoo {
    public static void main(String[] args) {
        Animal[] animaux = {
            new Chien(),
            new Chat(),
            new Vache()
        };

        for (Animal a : animaux) {
            a.crier();
            System.out.println("Régime : " + a.regimeAlimentaire());
        }
    }
}
```

Souhaites-tu que je te génère un projet complet avec tous les fichiers Java bien organisés ?