Programmation orientée objets

Documents et appareils électroniques interdits. 1.5 heures.

Exercice 1 (Questions) (12 points)

Question 1 : Expliquez la notion de sous-typage ? Quel en est l'intérêt ?

Question 2 : Pourquoi est-il préférable de ne dépendre que des interfaces ?

Question 3 : Expliquez le but et le fonctionnement des exceptions.

Question 4 : Qu'est-ce qu'une classe ?

Question 5 : Expliquez les notions de méthode abstraite et de classe abstraite.

Question 6 : Comment mettre en œuvre l'encapsulation d'un objet ?

Question 7 : Comment une sous-classe peut-elle invoquer le constructeur de sa super-classe ?

Question 8 : Quelle est la différence entre la redéfinition de méthodes et la surcharge ?

Question 9 : Quelle est la différence entre une classe abstraite et une interface Java ?

Question 10 : Quels sont les rôles du constructeur et du destructeur ?

Question 11 : Expliquer la notion de visibilité ? À quoi sert-elle ?

Question 12 : Quelle est la particularité de Java par rapport à un langage objet *compilé* (comme C++ par exemple) ?

Exercice 2 (Deux ternaires) (2 points)

Le programme Java suivant s'exécute sans erreur.

```
public class Ternary {
  public static void main(String[] args) {
    char x = 'X';
    int i = 0;
    System.out.print(true ? x  : 0);
    System.out.print(false ? i  : 'X');
  }
}
```

Qu'affiche-t-il ? Pourquoi ?

Exercice 3 (To be or not to be...) (3 points)

Le programme Java suivant compile et s'exécute sans erreurs. Qu'affiche-t-il et *pourquoi* ?

```
public class Indecision {
  public static void main(String[] args) {
    System.out.println(decision());
  }
  static String decision() {
    try {
      return "to be";
    } finally {
      return "not to be";
    }
  }
}
```

Exercice 4 Soit le listing Java suivant

```
import java.util.List;
import java.util.LinkedList;

class A {
  protected String name;
  public A(String n) { this.name = n ;}
  public String name() { return "A :" + this.name ;}
  public void bar(A a) {
    System.out.format("%s :A.bar(%s)\n", this.name(), a.name());
  }
}

class B extends A {
  public B(String n) { super(n); }
  public String name() { return "B :" + this.name ;}
  public void bar(B b) {
    System.out.format("%s :B.bar(%s)\n", this.name(), b.name());
  }
}
class C {
  public void baz(A a) { System.out.format("C.baz(A :%s)\n", a.name()); }
  public void baz(B b) { System.out.format("C.baz(B :%s)\n", b.name()); }
}

class TestP {
  public static void main(String[] args) {
    A a = new A("a");
    B b = new B("b");
    A b2 = new B("b2");
    C c = new C();

    List<A> s = new LinkedList<A>();
    s.add(a); s.add(b); s.add(b2);
    for (A e : s) {
      for (A f : s) {
        e.bar(f);
      }
      c.baz(e);
    }
  }
}
```

Donnez et expliquer la sortie de son exécution (le code compile et s'exécute sans erreur).