 Programmation Objet en Java - Feuille de TD 4

Exercice 1.

On considère les déclarations d'interfaces, de records et de méthodes suivantes, ainsi que les instructions de la méthode `main`. Quelles sont les lignes qui posent problème (à la compilation ou à l'exécution) ? Pourquoi ?

```
public interface A {}
public interface B {}
public interface C extends B {}
```

```
public record X() implements A {}
public record Y() implements C {}
public record Z() implements A, B {}
```

```
public class Test {
    public void p1(A a) {}
    public void p2(B b) {}
    public void p3(C c) {}
    public void p4(X x) {}

    public A r1() {
        return new A();
    }

    public A r2() {
        return new X();
    }

    public B r3() {
        return new Y();
    }

    public C r4() {
        return new Z();
    }
}
```

```
public static void main(String[] args) {
    X x1 = new X();
    p1(x1);
    p4(x1);

    A x2 = new X();
    p1(x2);
    p4(x2);
    p4((X) x2);
    p4(r2());

    B b1 = new Z();
    p1(b1);
    p2(b1);
    C c1 = b1;
    C c2 = (C) b1;

    B b2 = new Y();
    C c3 = b2;
    C c4 = (C) b2;
}
```

Exercice 2. On souhaite programmer un jeu d'aventure dans lequel les personnages ont un inventaire pour stocker leurs affaires. Chaque élément qui peut être stocké dans l'inventaire doit pouvoir indiquer son poids (avec une méthode `weight()`, par exemple).

- Dans un premier temps, l'inventaire pourra contenir des épées, des arcs, des flèches et des potions qui ont tous des façons différentes de calculer leur poids. Les épées et les arcs ont une durabilité et font des points de dégâts alors que les potions soignent. Les flèches sont de simples consommables. Les arcs ont également une capacité de tir (nombre de flèches simultanées).
- Un personnage peut vouloir trier son inventaire en ordre croissant de valeur des items qu'il contient. Pour toutes les armes, la valeur est calculée de la même façon, en fonction de leur poids et de leur durabilité, alors que pour les autres éléments, cela dépend de leurs caractéristiques propres.
- Un personnage peut aussi avoir des sorts qui font également des dégâts et qui peuvent avoir un effet spécial, mais qui ne sont pas stockés dans l'inventaire.
- Lorsqu'il attaque un ennemi, un personnage doit pouvoir passer en revue tous les éléments qui font des dégâts pour choisir celui qui fait le plus de dégâts.
- Dans le jeu, il y a également un forgeron qui vend uniquement des armes et une sorcière qui vend des potions et des sorts.

Créer une hiérarchie des types permettant de respecter toutes ces contraintes.