

Weapon.java

```

1 public interface Weapon {
2     int getDamage();
3     int getPrice();
4 }

```

Sword.java

```

1 public abstract class Sword implements Weapon {
2     public static final int PRICE_FACTOR = 2;
3     public static final int AGE_FACTOR = 100;
4     protected Material material;
5     protected int length;
6     protected int age = 0;
7
8     protected Sword(Material material, int length) {
9         this.material = material;
10        this.length = length;
11    }
12
13    @Override
14    public int getPrice() {
15        return this.length * this.material.getPrice() * this.getDamage() * PRICE_FACTOR - this.getAge()
16           ↪ * AGE_FACTOR;
17    }
18
19    public int getAge() {
20        return this.age;
21    }
22 }

```

Claymore.java

```

1 public class Claymore extends Sword {
2     public Claymore() {
3         super(Material.STEEL, 14);
4     }
5
6     @Override
7     public int getDamage() {
8         return 14;
9     }
10 }

```

Katana.java

```

1 public class Katana extends Sword {
2     public static final int PRICE_FACTOR = 3;
3
4     public Katana() {
5         super(Material.TAMAHAGANE, 7);
6     }
7
8     @Override
9     public int getDamage() {
10        return 12;
11    }
12
13    @Override
14    public int getPrice() {
15        return super.getPrice() * PRICE_FACTOR;
16    }
17 }

```

Material.java

```

1 public enum Material {
2     WOOD(2),
3     STEEL(8),
4     TAMAHAGANE(10);
5
6     private int price;
7
8     private Material(int price) {
9         this.price = price;
10    }
11
12    public int getPrice() {
13        return this.price;
14    }
15 }

```