

```
26 import java.util.Objects;
25
24 final class ImmutableClass {
23     ImmutableClass(MutableClass firstField, String secondField, Integer thirdField) {
22         this.firstField = new MutableClass(other: Objects.requireNonNull(firstField));
21         this.secondField = secondField;
20         this.thirdField = thirdField;
19     }
18
17     public MutableClass getFirstField() {
16         return new MutableClass(other: firstField);
15     }
14
13     public String getSecondField() {
12         // Копировать не нужно так как String является Immutable
11         return secondField;
10     }
9
8     public Integer getThirdField() {
7         // Копировать не нужно так как Integer является Immutable
6         return thirdField;
5     }
4
3     private final MutableClass firstField;
2     private final String secondField;
1     private final Integer thirdField;
27 }
```



MutableClass.java ×



ImmutableClass... ×

```
19 class MutableClass {
18     MutableClass(int value) {
17         this.setValue(value);
16     }
15
14     // Конструктор копирования
13     public MutableClass(MutableClass other) {
12         this.value = other.value;
11     }
10
9     public int getValue() {
8         return value;
7     }
6
5     public void setValue(int value) {
4         this.value = value;
3     }
2
1     private int value;
20 }
```