

Objektovo-orientované programovanie

Tea suits – výroba luxusných oblekov

Dokumentácia pre pracovnú verziu programu

Roman Bitarovský

Obsah

Tea suits – výroba luxusných oblekov	1
Prístupy pre používateľov	3
Aký má názov a hlavné črty vášho projektu	3
Spustíte program a prevedíte všetky dôležité vetvy programu.....	3
V zdrojových súboroch ukážte aspoň 1 hierarchiu dedenia tried (3 vrstvovú).....	3
Ukážte využitie polymorfizmu = aspoň jedno prekonávanie metódy vo vybudovanej hierarchii tried.....	4
Ukážte použitie agregácie (aspoň jeden atribút je typu trieda).	5
Ukážte oddelenie aplikačnej(business) logiky od používateľského rozhrania.....	6
Uved'te ďalšie črty projektu.	7

Prístupy pre používateľov

Manager

ID: 420

Heslo: 0000

Designer

ID: 421

Heslo: 0000

Tailor

ID: 430

Heslo: 0000

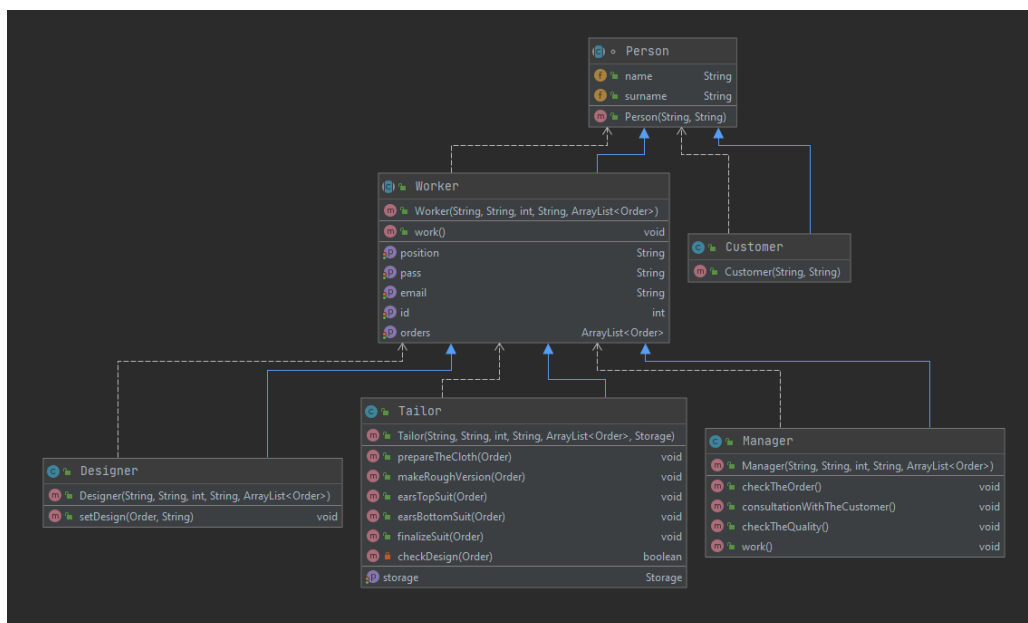
Aký má názov a hlavné črty vášho projektu

Názov je už spomínaný Tea suits – výroba luxusných oblekov. Jedná sa o butik/krajčírstvo, kt. cieľom je výroba luxusných oblekov na objednávku ako som už uvádzal v zámere projektu z prvého zadania. Cieľom môjho zadania je vytvoriť systém na spravovanie objednávok a reprezentácia určitých výrobných procesov. Hlavnou pointov systému je že máme jednotlivých používateľov, kt. sa do systému prihlasujú a podľa toho akú majú pozíciu môžu vykonávať rôzne úkony v systéme.

Spustíte program a prevedte všetky dôležité vetvy programu

Tú časť som odprezentoval na cvičení a podľa Vašich slov to vyzerá veľmi dobre.

V zdrojových súboroch ukážte aspoň 1 hierarchiu dedenia tried (3 vrstvovú)



```
4
5 2 usages 5 implementations
6 abstract class Person implements Serializable {
7
8 10 usages 3 implementations
9 public abstract class Worker extends Person implements Serializable {
10
11
12
13
14
15
16
17
18 2 usages
19 public class Manager extends Worker implements Serializable {
20
21
22
23
24
25
26
27
```

Ukážte využitie polymorfizmu = aspoň jedno prekonávanie metódy vo vybudovanej hierarchii tried.

Metóda work v triede Worker

```
22
23 1 usage 1 implementation
24 public void work(){
25     System.out.println("I like working");
26 }
27 // getters and setters
```

je prekonaná v triede Manager.

```
28
29 2 usages
30 @Override
31 public void work() { // Manager má kao prácu nejaké organizačné veci
32     super.work();
33     System.out.println("Manager pracuje 24/7");
34     checkTheOrders(); // napríklad vie zistiť koľko práve existuje objednávok
35 }
```

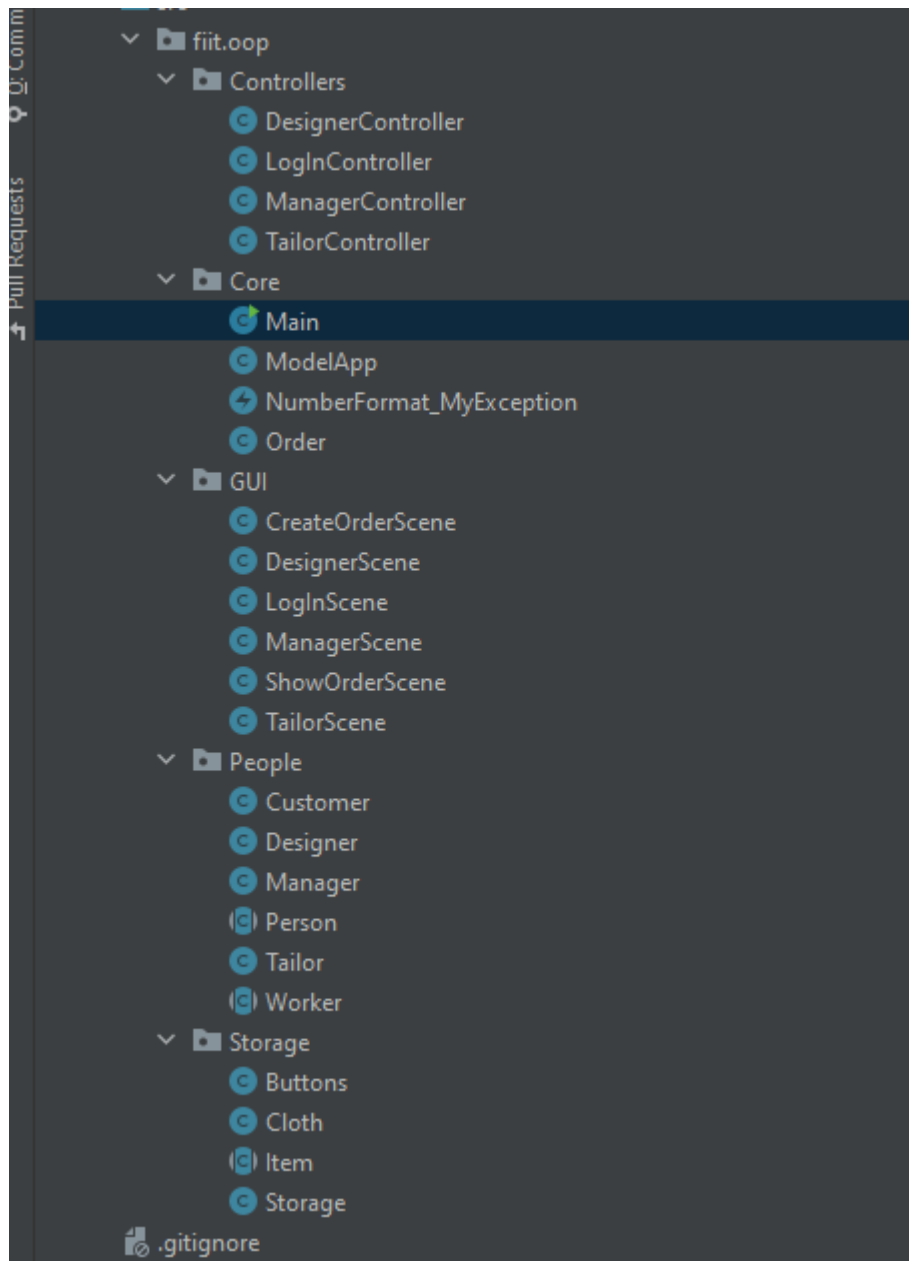
Ukážte použitie agregácie (aspoň jeden atribút je typu trieda).

Riadok 20 a 21.

```
14
21 usages
15 public class ModelApp implements Serializable {
16
17     3 usages
18     public ArrayList<Worker> workers;
19     public ArrayList<Order> orders;
20     4 usages
21     public Worker actualUser;
22     public Storage storage = new Storage();
23
24     1 usage
25     public ModelApp(){
26         this.workers = staffInitialization();
27         this.orders = new ArrayList<>();
28     }
```

Ukážte oddelenie aplikačnej(business) logiky od používateľského rozhrania.

Tak isto bolo prezentované počas prezentácie na cvičení.



Uved'te ďalšie črty projektu.

V projekte je už implementovaná serializácia. Konkrétne v triede LoginController keby sa volá pri ukončení programu.

```
47
1 usage
50 private void serializable() throws ClassNotFoundException, IOException {
51     FileOutputStream fos = new FileOutputStream( name: "modelAppData.txt");
52     ObjectOutputStream oos = new ObjectOutputStream(fos);
53     oos.writeObject(modelApp);
54     oos.close();
55 }
56 }
57
```

A tak isto pri spúšťaní v triede Main sa objekty deserializujú z príslušného súboru.

```
22
23 File file = new File( pathname: "modelAppData.txt");
24
25 try {
26     // deserializácia
27     FileInputStream fis = new FileInputStream(file);
28     ObjectInputStream ois = new ObjectInputStream(fis);
29     modelApp = (ModelApp) ois.readObject(); // down-casting object
30     ois.close();
31 } catch (ClassNotFoundException e) {
32     e.printStackTrace();
33 } catch (IOException ioException) {
34     ioException.printStackTrace();
35 }
36
```