

Домашна работа 3

„Електронен онлайн магазин“

Основни изисквания:

1. Да се създаде десктоп приложение „**Електронен онлайн магазин**“ като се използват следните технологии:
 - 1.1. **Java JDK** (версия 1.8 или по-нова)
 - 1.2. Java базирани библиотеки за създаване на десктоп приложения (**AWT** и **Swing**):
<https://docs.oracle.com/javase/8/docs/technotes/guides/awt/>
<https://docs.oracle.com/javase/8/docs/technotes/guides/swing/index.html>
2. Десктоп приложението трябва да предоставя възможност за избор на категория и/или подкатегория от каталог с категории, и след това да визуализира продуктите от съответната подкатегория, позволявайки функционалност за закупуване на съответен продукт.

Информацията за различните категории, подкатегории и продукти трябва да бъде прочетена от файл при зареждането на вашата програма. Този файл ще използваме вместо база данни. Форматът на файла е по-специфичен с цел по-лесното прочитане и обработване на информацията от него.

Например: Всеки път, когато имаме знак „<“ в началото на реда и знак „>“ в края на реда, това означава начало на нова категория, а всеки път, когато срещнем последователно знаците „</“ в началото на реда и знак „>“ в края на реда, това означава край на съответната категория.

Всеки път когато имаме знак „_“ в началото на реда, това означава нова подкатегория от съответната категория (името на самата подкатегория е веднага след знака „_“).

Продуктите от дадена подкатегория ги разграничаваме, като срещнем знак „*“ в началото на всеки ред. Името на продукта е веднага след знака „*“, последвано от цена на продукт и релативен път до файл със снимка на продукта, като името, цената и снимката на продукта са разделени със знак запетая („“).

Повече за релативните пътища до файлове можете да прочетете от следните източници:

- <https://docs.oracle.com/javase/tutorial/essential/io/pathOps.html>
- <https://www.journaldev.com/848/java-file-path-absolute-canonical>
- <https://www.baeldung.com/java-nio-2-path>

Примерно име на файл: **online_shop_database.txt**

Примерен формат на файла с категории, подкатегории и продукти:

```
<clothes>
  _trousers
    *Jack and Jones jeans,100.99,images/jackAndJonesJeans.jpg
    *Christian Dior pants,50.00,images/diorPants.jpg
  _T-shirts
    *Jack and Jones,9.99,images/jackAndJones.jpg
    *Christian Dior,25.00,images/dior.jpg
  _jackets
    *Burberry jacket,500.00,images/burberryJacket.jpg
    *Christian Dior jacket,800.00,images/diorJacket.jpg
</clothes>

<electronics>
  _notebooks
    *Acer Aspire 5750G,1000.99,images/acer5750G.jpg
    *Lenovo V13015IGM,1500.99,images/lenovo.jpg
    *Acer Aspire 3 A31553G35VX,2000.00,images/acer3.jpg
  _smartphones
    *Samsung Galaxy S9,1600.99,images/samsung.jpg
    *LG G5,1200.99,images/lg.jpg
    *Xiaomi Redmi Note 5 Pro,400.00,images/xiaomi.jpg
</electronics>
```

Бележка: Това е само примерен файл с категории, подкатегории и продукти. Свободни сте да експериментирате с повече и различни категории, подкатегории и продукти.

За реализирането на задачата трябва да се имплементират поне три екрана с различна функционалност:

2.1. **Екран номер 1:** На първия екран в най-горната част визуализирайте името на *Вашия онлайн магазин*. От лявата част на екрана трябва да се имплементира функционалност позволяваща възможност за *избор на категория и подкатегория от съответна категория*. От дясната част на екрана трябва да се визуализира *съобщение по подразбиране*.

Например: Поздравяваме клиента и го приветстваме с добре дошъл в нашия електронен онлайн магазин.

Online shop name	
Category 1	WELCOME !!! Please, select a category.
Subcategory 1	
Subcategory 2	
Category 2	
Category 3	
Subcategory 1	
Subcategory 2	
Subcategory 3	

2.2. **Екран номер 2:** След като вече потребителят е избрал подкатегория, в дясната част на екрана трябва да се визуализират всички продукти от съответната подкатегория.

Имплементирайте функционалност, която позволява **сортиране на продукти по азбучен ред** и/или **по текуща цена на продукта**, като сортировката по подразбиране трябва да бъде по цена.

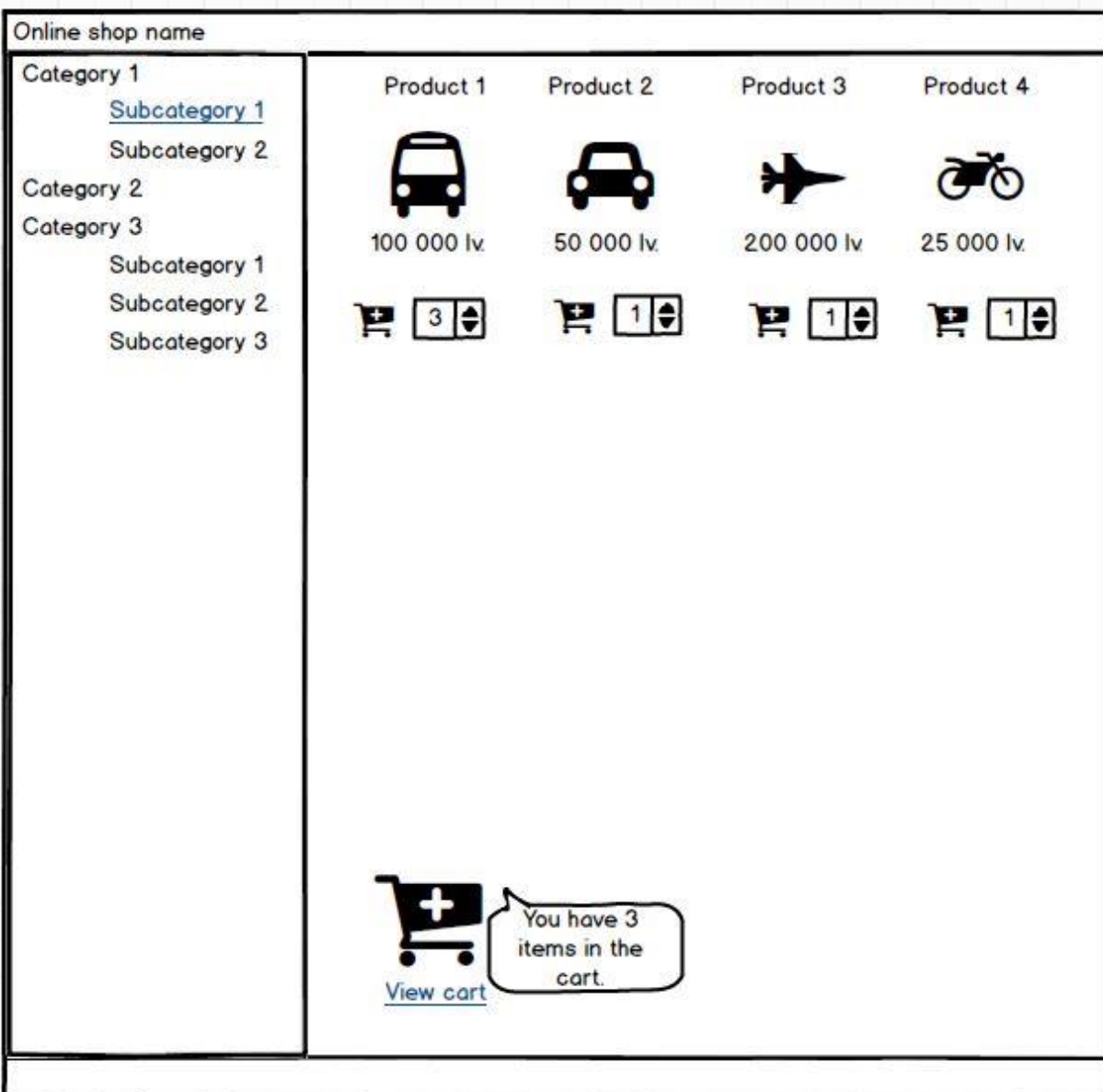
Всеки продукт от дадена подкатегория трябва да бъде описан с неговото **име, снимка, цена и възможност за избор на брой**, тоест колко продукта иска да закупи съответният клиент.

Непосредствено до възможността за добавяне на брой продукти трябва да има бутон **„Добави в количка“**, като на примерния дизайн за екран този бутон е реализиран посредством *снимка на пазарска количка*.

Най-долу на този екран трябва да имплементирате бутон **„Количка“**, който да отведе потребителя към количката с всички продукти избрани от него до момента.

Добавете функционалност, която позволява на потребителя, когато посочи бутона **„Количка“**, с курсора на мишката, да се визуализира информация за това **колко продукта са избрани към настоящия момент**. Имате свободата да изберете начина за визуализиране на тази информация.

Примерен екран може да намерите на [страница 5 от документа](#).



2.3. **Екран номер 3:** След натискането на бутон „Количка” визуализирайте **модален прозорец** с информация за всеки един от избраните продукти в табличен вид в следния формат: *име на продукт, количество, единична цена, обща цена*. Под таблицата с продукти визуализирайте **крайна цена за фактуриране** на цялата поръчка.

Чрез **double-click** функционалност на модалния прозорец, потребителят трябва да може да **редактира количеството** на избраните от него продукти.

Потребителят трябва да може да плати своята поръчка по два начина. Чрез **кредитна/дебитна карта** или чрез **PayPal**. Реализирайте два бутона. По един за всяко едно от видовете плащания.

При натискане на някой от двата бутона, трябва да се **генерира фактура** на потребителя. За целта, трябва да запишете във файл цялата информация от модалния прозорец за всеки продукт и за крайната цена.

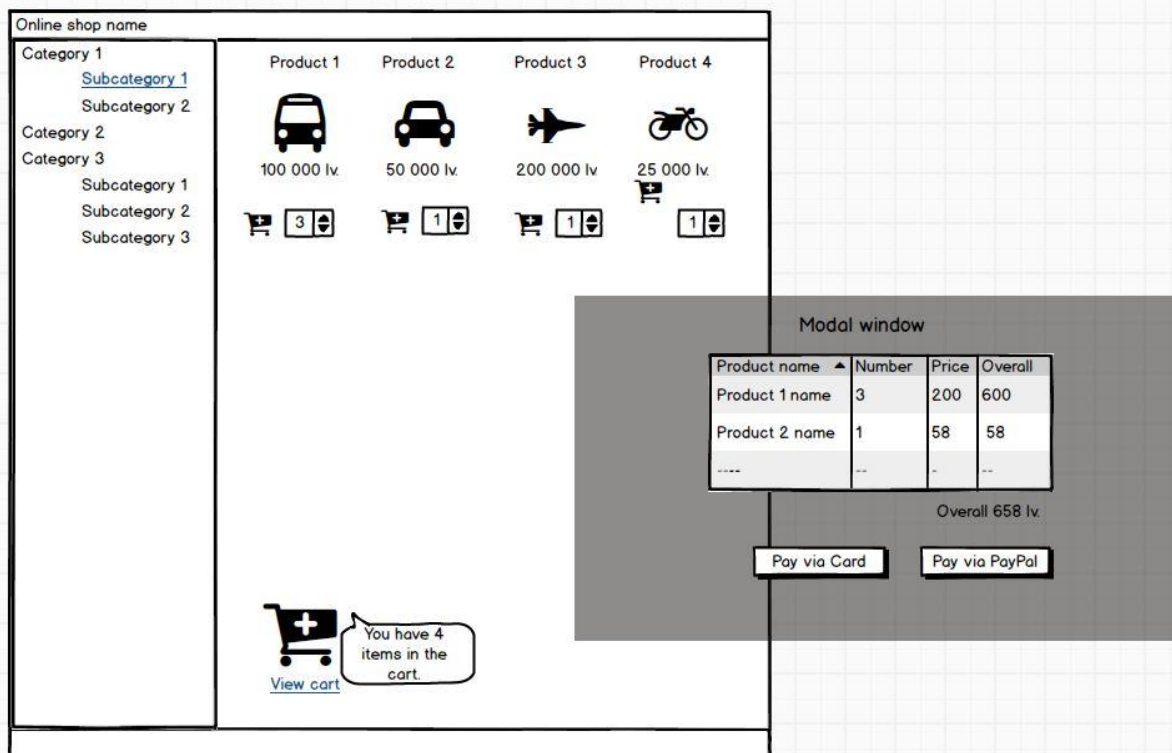
Форматът за записване във файла е по ваш избор. Може да използвате **.csv**, **.txt** или друг формат за записване на информацията. Този файл ще играе роля на фактура за тази поръчка.

Условие по желание: За по-любознателните, опитайте да запишете информацията в табличен вид в **Excel** файл. За целта трябва да използвате допълнителна **Java** библиотека, която не е налична в **JDK**. Може да я добавите директно като **.jar** файл във вашия проект или при създаването на вашия **Eclipse** проект, да го създадете като **Maven** проект и да добавите библиотеката, под формата на **dependency** в **pom.xml** файла, който ще ви се създаде автоматично след като сте създали вашия проект, като **Maven** проект. Библиотеката, която трябва да използвате се казва **Apache POI**.

Информация може да намерите в следните източници:

- <https://maven.apache.org/guides/introduction/introduction-to-repositories.html>
- <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
- <https://www.baeldung.com/maven>
- <https://www.tutorialspoint.com/maven/>
- <http://www.vogella.com/tutorials/EclipseMaven/article.html>
- <http://poi.apache.org/>
- <http://viralpatel.net/blogs/java-read-write-excel-file-apache-poi/>
- <https://www.baeldung.com/java-microsoft-excel>

Примерен трети екран може да намерите на [**страница 7 от документа**](#).



2.4. **Допълнителни условия:** Да се използват различните концепции разгледани в упражненията, където е възможно (**интерфейси, наследяване, полиморфизъм, вътрешни и анонимни класове, компаратори, събития и т.н.**).

Забележка: Това е само примерен графичен потребителски интерфейс. Чувствайте се свободни да изградите графичния интерфейс, така както желаете, стига да удовлетворява функционалните изисквания описани по-горе в документа.

ЗАДАЧИ ЗА ИМПЛЕМЕНТАЦИЯ

Всяка една от разработените задачи ще се оценява в точкова система с арбитрарни тежести. Максималното количество точки, които можете да получите от решението на поставените ви условия е **100 точки**. Повече информация за предаването на вашите решения можете да намерите в секция **ПРЕДАВАНЕ НА РЕШЕНИЯТА**.

функционалност	точки
създаване на необходимите класове, характеристики, методи, конструктори, get/set методи, използване на правилни и подходящи ООП принципи, правила и парадигми	10
избор на подходящи модификатори за достъп и типове данни	5
реализация на логиката за сортиране на артикулите по различни критерии	5
реализиране на логиката по четене на артикули от файл	15
реализация на логиката за записване, на добавените в количката, артикули във файл	15
реализация на логиката по добавяне на артикули в количката	10
създаване на основен екран	5
създаване на менюто с всички категории и подкатегории	10
създаване на бутон „Количка“ и отваряне на модален прозорец при натискане на този бутон	5
създаване на екран за визуализация на всички артикули	10
създаване на екран за визуализиране на всички артикули в количката в табличен вид, опция за редактиране на количеството артикули и бутони за плащане	10
ОБЩО	100

КРИТЕРИИ ЗА ВАЛИДНА ОЦЕНКА

В тази част са описани критериите, които добавят положителен резултат към крайната оценка. За да бъдат приложени е необходимо да не е влязло в сила, нито едно от правилата описани в предходната част.

Едно домашно няма да бъде проверявано или ще бъде наказвано с **0 точки** при неговото предаване, ако налице са някои от следните обстоятелства.

обстоятелство	наказание
<ul style="list-style-type: none">Програмата не се компилира;Програмата е предадена във вид, който изисква допълнителна намеса от страна на проверяващия, за да стартира;Изпълними файлове с грешно разширение.	0 точки
Домашна работа, предадено извън срока описан в заданието	0 точки
<i>Нагло плагиатство, дословно копиране и погазване на моралните принципи за справедливо предаване на домашна работа</i>	<i>Слаб (2) за редовна дата</i>
Грубо погазване на описаните правила и тотално неспазване на условията на домашните работи	0 точки

ПРОЦЕНТНО СЪОТНОШЕНИЕ НА КРИТЕРИИТЕ, СПРЯМО ДАДЕНИТЕ ТОЧКИ

Графична имплементация	проценти
<ul style="list-style-type: none">Именуване на променливи и методи	30%
<ul style="list-style-type: none">Коректност на имплементираната функционалност	70%

ИМЕНУВАНЕ НА ПРОМЕНЛИВИ И МЕТОДИ

Правилното именуване на методите и променливите са ключови за лесното разбиране на кода, което е свързано с предишните два критерия.

КОРЕКТНА ФУНКЦИОНАЛНОСТ

Това е критерият с най-голяма тежест, тъй като целта на всяко задание в крайна сметка е коректно работещо приложение. Функционалността в текущото задание е разделена в няколко точки. Затова и всяка правилно реализирана функция прибавя резултат към крайната оценка.

ПРЕДАВАНЕ НА РЕШЕНИЯТА

Всички участници трябва да предадат задачата си, използвайки съвременна и проверена схема за дистрибуция на софтуерни инструкции.

Задължително е да си направите нов проект в **GitLab**-а на курса, със заглавие: ***pu-fmi--java-oop-h3***

Името на хранилището не бива да бъде бъркано, преправено, импровизирано или подложено на творческа интерпретация. Ако това се случи, за ваше съжаление задачата ви няма да бъде приета и проверена, което ще донесе само и единствено тъга в сърцата на всички панди в средната земя.

Линк към хранилището трябва да бъде предоставен в **Google Classroom** преди крайния срок на заданието. За целта, трябва да го прикачите към заданието и след това да го предадете като натиснете бутон „**Предай**“. Хранилище, което е предоставено като коментар няма да бъде отворено и оценено!

Ако изпитвате затруднение комуникирайте с преподавател за правилния начин, по който да предадете решението. Като е добре това да се случва в срокове различни от последната минута на последния ден.

Успех!