

Fabiola Dąbroś

Inżynieria Obliczeniowa gr.1

SKLEP Z ODZIEŻĄ

W ramach laboratorium stworzyłam program w Javie, który odzwierciedla sklep z odzieżą. Po uruchomieniu programu dokonujemy wyboru czy interesuje nas odzież męska czy damska po czym wyświetla nam się dostępna lista produktów.

```
CLOTHES SHOP

Please make a choice. Choose for who you want to buy a clothes.

1.Women
2.Men
3.Exit.

1
Available products for women:
Kind: trousers      Size: 38      Color: Jeans      Price: 99.90      Year: 2018
Kind: dress         Size: 36      Color: Prink      Price: 59.90      Year: 2018
Kind: blouse        Size: 34      Color: Black      Price: 29.90      Year: 2018
Kind: socks         Size: 37      Color: colourful   Price: 19.90      Year: 2018
Kind: trousers      Size: 40      Color: Grey       Price: 79.90      Year: 2018

Do you want to buy something or exit?
1. I want to buy.
2. Exit.
```

Jeśli użytkownik jest zainteresowany zakupem musi wprowadzić informację o obiekcie, który chciałby kupić. Oprócz tego w celu ustalenia kosztu przesyłki konieczne jest wprowadzenie odległości w kilometrach od Krakowa.

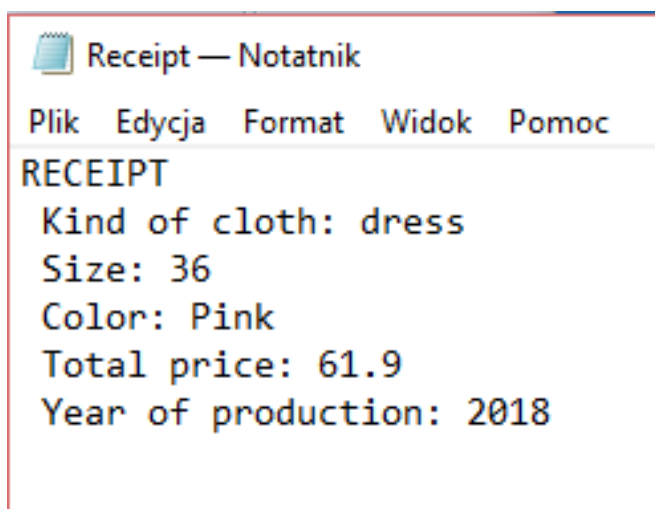
```
Please enter all information about your clothes.

Enter kind. Enter size. Enter color. Enter price. Enter year.
dress
36
Pink
59,90
2018
Serialized data is saved in 'Object.ser'
Deserialization : Kind: dress Size: 36 Color: Pink Year: 2018

Do you want to print receipt and pay for product or exit?
1.Print receipt and pay.
2.Exit.
1

Please remember you have to pay for delivery. The price depends on your distance in kilometers from Cracow.
Please put a kilometers
2
Your delivery cost is: 2.0
```

Kolejno zapisywany jest do pliku tekstowego paragon przechowujący wszystkie informacje o obiekcie, który klient zakupił.



Oprócz tego program oczekuje kilka sekund aż klient otrzyma swój zakup po czym istnieje możliwość przymierzenia ubrania. Użytkownik musi podać jaki nosi rozmiar w celu określenia zgodności z zamówioną rzeczą.

```
Your receipt has been saved to 'Receipt.txt'
Your status of purchase is: true
Your code:
7381
3485

Today is: 13-03-2018
You have to wait one day for your delivery. Let's wait together!

Incredible! You have already get your product! Do you want to try it on?
1. Of course.
2. Exit.
1
Please give me your size in a number.
36
It is perfect for you!
```

W ramach wykonania programu stworzyłam odpowiednie klasy. Wykorzystałam w nich dziedziczenie dzięki czemu utworzyłam spójną i łatwą do zrozumienia hierarchię klas. Dziedziczenie spowodowało przekazanie pewnych cech innym klasom. W moim przypadku klasy WomanGear oraz MenGear dziedziczą po klasie Gear. Dodatkową jest ona klasą abstrakcyjną, która zawiera abstrakcyjną metodę trySthOn, która jest następnie nadpisywana przez klasy dziedziczące.

Stworzyłam również interfejs Buying, który jest wymogiem jaki muszą spełniać klasy dziedziczące ten interfejs. Znajduje się w nim metoda odpowiedzialna za zakup produktu. Zwraca ona kwotę uzależnioną od ceny produktu oraz odległości od Krakowa. Metoda ta nie posiada żadnej implementacji, ani nawet nawiasów klamrowych, ponieważ interfejs to kontrakt, który mówi co klasa implementująca może robić, ale nie musi.

W wielu przypadkach konieczne było zastosowanie wyjątków. W tym celu stosuje się blok try catch finally. W bloku try znajduje się kod, który może generować wyjątki. Blok catch zajmuje się obsługą wyjątku zadeklarowanego odpowiednim typem. Natomiast to co wykona się w bloku finally wykona się zawsze. Przykładem użycia w moim programie może być np. próba odczytania danych z pliku. Wyłapywany jest wyjątek w przypadku gdy dana operacja się nie powiedzie. Natomiast instrukcja zamknięcia strumienia wykonuje się zawsze. Oprócz wyjątków z grupy wyjątków

obsługiwanych utworzyłam również swoją klasę wyjątków, która musi dziedziczyć do klasie Exception. Wykorzystywany on jest w metodzie trySthOn.

Istnieje również inny sposób zgłaszania wyjątków poprzez throws oraz throw new. Wykorzystujemy to głównie wtedy kiedy mamy do czynienia z wyjątkami, których nie da się obsłużyć lub ich obsługa nie ma sensu. W moim przypadku dotyczy to metody kupowania kiedy nie można podać liczby kilometrów mniejszej od zera.

W celu pogrupowania klas według ich znaczenia dokonałam podziału na pakiety. Utworzyłam pakiet clothes, myException oraz recordReading. Dzięki temu oddzieliłam od siebie klasy tego samego typu. W celu używania klas z innych pakietów posługujemy się importowaniem lub wypisujemy po kropce kolejne poziomy zagnieżdżenia pakietu.

W programie korzystam również z odczytu z pliku tekstowego. Podstawowa klasa, która umożliwi nam utworzyć obiekt przechowujący dany pliku to FileReader. Musimy określić ścieżkę do naszego pliku. W celu odczytania tekstu liniami dlatego opakowujemy ta klasę klasą BufferedReader. Należy również pamiętać o zwolnić zajęte zasoby. W moim programie wykorzystuje odczyt z pliku tekstowego w celu wyświetlenia dostępnej listy produktów. Natomiast w przypadku zapisu do pliku tekstowego korzystamy z klasy PrintWriter, która musi otrzymać nazwę pliku.

Dokonałam również zapisu i odczytu z użyciem serializacji. Jest to proces przekształcania obiektów w strumień bajtów z zachowaniem aktualnego stanu. Każdy obiekt, który zaimplementował interfejs Serializable może serializować oraz deserializować. W tym celu tworzymy obiekt do zapisu i w konstruktorze przekazujemy adres do pliku. Obiekt FileOutputStream łączy się z plikiem. Dokonujemy zapisu obiektu oraz zamykamy strumień. Następnie możemy wykonać deserializację czyli proces odwrotny. Tym razem korzystamy z FileInputStream.