

AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

Projektowanie obiektowe

Wprowadzanie zmian w istniejącej aplikacji

Laboratorium 2

Przemysław Lechowicz

1. Zadania

1.1. Dodatkowe właściwości dla produktów w poszczególnych kategoriach

W celu wykonania zadania dodajemy w klasie Item pola oznaczające: ilość stron(pages), czy książka posiada twardą oprawę (hardCover), czy przedmiot jest mobilny (mobile), czy przedmiot posiada gwarancję (warranty), datę przydatności do spożycia(expirationDate), gatunek muzyczny (genre), czy jest dodane wideo (addedVideo), wraz z getterami i setterami.

```
public void setPages(int pages) {
    this.pages = pages;
}
public boolean getHardCover() {
    return this.hardCover;
}
public void setHardCover(boolean hardCover) {
    this.hardCover = hardCover;
}
public boolean getMobile(){
    return mobile;
}
public void setMobile(boolean mobile){
    this.mobile = mobile;
}
public boolean getWarranty(){
    return warranty;
}
public void setWarranty(boolean warranty){
    this.warranty = warranty;
}
public boolean getAddedVideo(){
    return addedVideo;
}
public void setAddedVideo(boolean addedVideo){
    this.addedVideo = addedVideo;
}
public Date getExpirationDate(){
    return expirationDate;
}
public void setExpirationDate(Date expirationDate){
    this.expirationDate = expirationDate;
}
public Genre getGenre() {
    return genre;
}
public void setGenre(Genre genre){
    this.genre = genre;
}
```

Utworzony został również enum Genre, który będzie określał gatunek dla kategorii muzyka, wraz z metodą parse, którą wykorzystamy w klasie ShopProvider w celu zmiany obiektu String na Genre:

```
package pl.edu.agh.dronka.shop.model;

public enum Genre {
    ROCK, POP, CLASSIC, METAL;

    public static Genre parse(String genre) {
        if (genre.equals("ROCK")) {
            return Genre.ROCK;
        }
        if (genre.equals("POP")) {
            return Genre.POP;
        }
        if (genre.equals("CLASSIC")) {
            return Genre.CLASSIC;
        }
        if (genre.equals("METAL")) {
            return Genre.METAL;
        }
        return ROCK;
    }
}
```

Następnie, w klasie ShopProvider dodajemy do obiektu Item odpowiednie parametry w zależności od kategorii. Dla kategorii Book są to: ilość stron i czy książka ma twardą oprawę, dla kategorii Elektronika są to czy przedmiot posiada gwarancję i czy jest mobilny, dla żywności jest to data przydatności do spożycia, a dla muzyki czy jest dołączone wideo i gatunek.

```
if(category == Category.BOOKS) {
    boolean hardCover = Boolean.parseBoolean(reader.getValue(
        dataLine, "Twarda oprawa"));
    int pages = Integer.parseInt(reader.getValue(dataLine,
        "Liczba stron"));
    item.setPages(pages);
    item.setHardCover(hardCover);
}
else if(category == Category.ELECTRONICS) {
    boolean warranty = Boolean.parseBoolean(reader.getValue(
        dataLine, "Gwarancja"));
    boolean mobile= Boolean.parseBoolean(reader.getValue(
        dataLine, "Mobilny"));
    item.setMobile(mobile);
    item.setWarranty(warranty);
}
else if(category == Category.FOOD) {
    Date expirationDate = new
SimpleDateFormat("dd/MM/yyyy").parse(reader.getValue(
        dataLine, "Data spozycia"));
    item.setExpirationDate(expirationDate);
}
else if(category == Category.MUSIC) {
    boolean addedVideo= Boolean.parseBoolean(reader.getValue(
        dataLine, "Dodane Wideo"));
    Genre genre = Genre.parse(reader.getValue(dataLine, "Dodane Wideo"));
    item.setAddedVideo(addedVideo);
    item.setGenre(genre);
}
```

W klasie ItemDetailsPanel tworzymy dodatkowe etykiety w zależności od kategorii, które będą wyświetlane w panelu ItemDetails:


```
if (item.getCategory() == Category.BOOKS) {
    createInfoLabel("Liczba stron", String.valueOf(item.getPages()));
    createInfoLabel("Twarda Oprawa", String.valueOf(item.getHardCover()));
}
else if (item.getCategory() == Category.ELECTRONICS) {
    createInfoLabel("Gwarancja", String.valueOf(item.getWarranty()));
    createInfoLabel("Mobilny", String.valueOf(item.getMobile()));
}
else if (item.getCategory() == Category.FOOD) {
    createInfoLabel("Data przydatności do spożycia", String.valueOf(item.getExpirationDate()));
}
else if (item.getCategory() == Category.MUSIC) {
    createInfoLabel("Gatunek", String.valueOf(item.getGenre()));
    createInfoLabel("Dodane video", String.valueOf(item.getAddedVideo()));
}
```


Modyfikujemy pliki .csv dodając do pliku:

- electronics.csv pola Mobilny i Gwarancja;
- food.csv pole Data spożycia
- music.csv pola Wideo i Gatunek

Po dodaniu odpowiednich pól w plikach .csv otrzymujemy następujące widoki przedmiotów: (przykładowo zostanie pokazane po jednym przedmiocie z każdej kategorii, oprócz kategorii sport, która nie wymagała zmian):

Kategoria KSIĄŻKI:

 Dronka Shop



Nazwa: When The Smoke is Going Down : Wprowadzenie do testów wydajnościowych
Cena: 90
Kategoria: Książki
Ilość: 3
Tanie bo polskie: true
Używany: false
Liczba stron: 2000
Twarda Oprawa: true

Kategoria ELEKTRONIKA:

 Dronka Shop



Nazwa: Telewizor LCD El Dzi
Cena: 2000
Kategoria: Elektronika
Ilość: 10
Tanie bo polskie: false
Używany: true
Gwarancja: true
Mobilny: false

Kategoria ŻYWNOŚĆ:

 Dronka Shop



Nazwa: Sznycel mrożony
Cena: 6
Kategoria: Żywność
Ilość: 30
Tanie bo polskie: true
Używany: false
Data przydatności do spożycia: Fri Nov 20 00:00:00 CET 2020

Kategoria MUZYKA:

 Dronka Shop



Nazwa: Ciley Myrus - Big Ball of Mud

Cena: 60

Kategoria: Muzyka

Ilość: 20

Tanie bo polskie: true

Używany: false

Gatunek: ROCK

Dodane video: true

2. Rozszerzenie panelu

W klasie Properties panel został dodany kod, który tworzy nowe checkbox'y i na ich podstawie wywołuje po zaznaczeniu metodę filtrowania:

```
if(shopController.getCurrentCategory() == Category.BOOKS) {
    add(createPropertyCheckbox("Twarda oprawa", new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent event) {
            filter.getItemSpec().setHardCover(
                ((JCheckBox) event.getSource()).isSelected());
            shopController.filterItems(filter);
        }
    }));
}
if(shopController.getCurrentCategory() == Category.MUSIC) {
    add(createPropertyCheckbox("Dołączone wideo", new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent event) {
            filter.getItemSpec().setAddedVideo(
                ((JCheckBox) event.getSource()).isSelected());
            shopController.filterItems(filter);
        }
    }));
}
if(shopController.getCurrentCategory() == Category.ELECTRONICS) {
    add(createPropertyCheckbox("Gwarancja", new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent event) {
            filter.getItemSpec().setWarranty(
                ((JCheckBox) event.getSource()).isSelected());
            shopController.filterItems(filter);
        }
    }));
    add(createPropertyCheckbox("Mobilny", new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent event) {
            filter.getItemSpec().setMobile(
                ((JCheckBox) event.getSource()).isSelected());
            shopController.filterItems(filter);
        }
    }));
}
```


Natomiast w klasie ItemFilter zostały dodane warunki filtrowania:

```
if (itemSpec.getHardCover() && !item.getHardCover()) {  
    return false;  
}  
if (itemSpec.getMobile() && !item.getMobile()) {  
    return false;  
}  
if (itemSpec.getWarranty() && !item.getWarranty()) {  
    return false;  
}  
if (itemSpec.getAddedVideo() && !item.getAddedVideo()) {  
    return false;  
}
```

Teraz panel wyświetlania produktów dla każdej z kategorii wygląda następująco:

Kategoria KSIĄŻKI:

 Dronka Shop



- ☐ Tanie bo polskie
- ☐ Używany
- ☒ Twarda oprawa
- Nowe Przygody Gangu Czworga
- When The Smoke is Going Down : Wprowadzenie do testów wydajnościowych

Kategoria ELEKTRONIKA:

 Dronka Shop



- ☐ Tanie bo polskie
- ☐ Używany
- ☐ Gwarancja
- ☐ Mobilny

Telewizor LCD El Dzi
Legendarny Bulbulator

Kategoria MUZYKA:

 Dronka Shop



- ☐ Tanie bo polskie
- ☐ Używany
- ☒ Dołączone wideo

Ciley Myrus - Big Ball of Mud