Laboratorium 2

Adrian Wójciak, Lena Obertyńska

1. **Dodatkowe właściwości dla produktów w poszczególnych kategoriach**

a) Dodatkowe klasy

Zostało stworzonych pięć klas, z których każda odpowiada jednej kategorii i jest rozszerzeniem klasy Item:

- Books

```
public class Books extends Item {
      private int pagesCount; // liczba stron
      private boolean hasHardCover; // twarda okładka

      public Books(String name, Category category, int price, int
quantity, int pagesCount, boolean hasHardCover){
          super(name, category, price, quantity);
          this.pagesCount = pagesCount;
          this.hasHardCover = hasHardCover;
      }

      public Books() {
      }

      public int getPagesCount() {
          return this.pagesCount;
      }

      public void setPagesCount(int pagesCount) {
          this.pagesCount = pagesCount;
      }

      public boolean hasHardCover() {
          return this.hasHardCover;
      }

      public void setHasHardCover(boolean hasHardCover) {
          this.hasHardCover = hasHardCover;
      }
}
```

- Electronics

```
public class Electronics extends Item {
	private boolean isMobile; // mobilny
	private boolean hasWarranty; // gwarancja

	public Electronics(String name, Category category, int price, int
quantity, boolean isMobile, boolean hasWarranty) {
		super(name, category, price, quantity);
		this.isMobile = isMobile;
		this.hasWarranty = hasWarranty;
	}

	public Electronics() {
	}

	public boolean isMobile() {
		return this.isMobile;
	}

	public void setMobile(boolean mobile) {
		this.isMobile = mobile;
	}

	public boolean hasWarranty() {
		return this.hasWarranty;
	}

	public void setHasWarranty(boolean hasWarranty) {
		this.hasWarranty = hasWarranty;
	}
}
```

- Food

```
public class Food extends Item {
	private Date expirationDate; // data przydatności

	public Food(String name, Category category, int price, int
quantity,  Date expirationDate) {
		super(name, category, price, quantity);
		this.expirationDate = expirationDate;
	}

	public Food() {
	}
```

```
	public Date getExpirationDate() {
		return this.expirationDate;
	}

	public void setExpirationDate(Date expirationDate) {
		this.expirationDate = expirationDate;
	}
}
```

- Music

```
public class Music extends Item {
	private MusicGenre genre; // gatunek muzyczny
	private boolean isVideoIncluded; // dołączone wideo

	public Music(String name, Category category, int price, int
quantity, String genreName, boolean isVideoIncluded) {
		super(name, category, price, quantity);
		this.genre = MusicGenre.fromString(genreName);
		this.isVideoIncluded = isVideoIncluded;
	}

	public Music() {
	}

	public MusicGenre getGenre() {
		return this.genre;
	}

	public void setGenre(MusicGenre genre) {
		this.genre = genre;
	}

	public boolean isVideoIncluded() {
		return this.isVideoIncluded;
	}

	public void setVideoIncluded(boolean videoIncluded) {
		this.isVideoIncluded = videoIncluded;
	}
}
```

Dodatkowo został stworzony enum MusicGenre:

```
public enum MusicGenre {
	RAP("Rap"), POP("Pop"), JAZZ("Jazz"), BLUES("Blues"),
```

```
CLASSIC("Klasyczna");

	private String displayName;

	public String getDisplayName() {
		return displayName;
	}

	MusicGenre(String displayName) {
		this.displayName = displayName;
	}

	public static MusicGenre fromString(String name) {
	    switch(name) {
			case "Rap":
				return RAP;
			case "Pop":
				return POP;
			case "Jazz":
				return JAZZ;
			case "Blues":
				return BLUES;
			case "Klasyczna":
				return CLASSIC;
			default:
				throw new IllegalArgumentException("Gatunek
nierozpoznany.");
		}
	}
}
```

- Sport

```
public class Sport extends Item {
	public Sport(String name, Category category, int price, int
quantity) {
		super(name, category, price, quantity);
	}

	public Sport() {
	}
}
```

b) Rozszerzenie funkcjonalności klasy PropertiesHelper

Metoda getPropertiesMap została zmodyfikowana tak, by do wyniku jej działania były dodawane właściwości zależne od kategorii:

```
public static Map<String, Object> getPropertiesMap(Item item) {
	Map<String, Object> propertiesMap = new LinkedHashMap<>();

	propertiesMap.put("Nazwa", item.getName());
	propertiesMap.put("Cena", item.getPrice());
	propertiesMap.put("Kategoria",
item.getCategory().getDisplayName());
	propertiesMap.put("Ilość", Integer.toString(item.getQuantity()));
	propertiesMap.put("Tanie bo polskie", item.isPolish());
	propertiesMap.put("Używany", item.isSecondhand());

	if (item instanceof Books) {
		propertiesMap.put("Liczba stron", ((Books)
item).getPagesCount());
		propertiesMap.put("Twarda oprawa", ((Books)
item).hasHardCover());
	} else if (item instanceof Electronics) {
		propertiesMap.put("Mobilny", ((Electronics)
item).isMobile());
		propertiesMap.put("Gwarancja", ((Electronics)
item).hasWarranty());
	} else if (item instanceof Food) {
		propertiesMap.put("Data przydatności", ((Food)
item).getExpirationDate());
	} else if (item instanceof Music) {
		propertiesMap.put("Gatunek muzyczny", ((Music)
item).getGenre().getDisplayName());
		propertiesMap.put("Dołączone wideo", ((Music)
item).isVideoIncluded());
	}

	return propertiesMap;
}
```

c) Rozszerzenie funkcjonalności klasy ShopProvider

Metoda readItems została zmodyfikowana tak, by wczytywała właściwości zależne od danej kategorii:

```
private static List<Item> readItems(CSVReader reader, Category category)
{
	List<Item> items = new ArrayList<>();

	try {
		reader.parse();
		List<String[]> data = reader.getData();
```

```
            for (String[] dataLine : data) {

                  String name = reader.getValue(dataLine,"Nazwa");
                  int price =
Integer.parseInt(reader.getValue(dataLine, "Cena"));
                  int quantity =
Integer.parseInt(reader.getValue(dataLine,
                              "Ilość"));

                  boolean isPolish =
Boolean.parseBoolean(reader.getValue(
                              dataLine, "Tanie bo polskie"));
                  boolean isSecondhand =
Boolean.parseBoolean(reader.getValue(
                              dataLine, "Używany"));
                  Item item;

                  switch(category) {
                        case BOOKS:
                              int pagesCount =
Integer.parseInt(reader.getValue(dataLine, "Liczba stron"));
                              boolean hasHardCover =
Boolean.parseBoolean(reader.getValue(dataLine, "Twarda oprawa"));
                              item = new Books(name, category, price,
quantity, pagesCount, hasHardCover);
                              break;
                        case ELECTRONICS:
                              boolean isMobile =
Boolean.parseBoolean(reader.getValue(dataLine, "Mobilny"));
                              boolean hasWarranty =
Boolean.parseBoolean(reader.getValue(dataLine, "Gwarancja"));
                              item = new Electronics(name, category,
price, quantity, isMobile, hasWarranty);
                              break;
                        case FOOD:
                              String timestring =
reader.getValue(dataLine, "Data przydatności");
                              Date expirationDate =  (new
SimpleDateFormat("yyyy-MM-dd")).parse(timestring);
                              item = new Food(name, category, price,
quantity, expirationDate);
                              break;
                        case MUSIC:
                              String genreName =
```

```
reader.getValue(dataLine, "Gatunek muzyczny");
                              boolean isVideoIncluded =
Boolean.parseBoolean(reader.getValue(dataLine, "Dołączone wideo"));
                              item = new Music(name, category, price,
quantity, genreName, isVideoIncluded);
                              break;
                        case SPORT:
                              item = new Sport(name, category, price,
quantity);
                              break;
                        default:
                              item = new Item(name, category, price,
quantity);
                  }

                  item.setPolish(isPolish);
                  item.setSecondhand(isSecondhand);

                  items.add(item);

            }

      } catch (IOException | ParseException e) {
            e.printStackTrace();
      }

      return items;
   }
```

d) Modyfikacja danych

W plikach books.csv i electronics.csv znajdowały się już wartości nowych właściwości, zatem te należało dodać tylko w plikach food.csv i music.csv

- food.csv

```
Nazwa,Cena,Ilość,Tanie bo polskie,Używany,Data przydatności
Zupa Studencka Instant,2,100,TRUE,FALSE,2137-04-02
Sznycel mrożony,6,30,TRUE,FALSE,2020-06-06
```

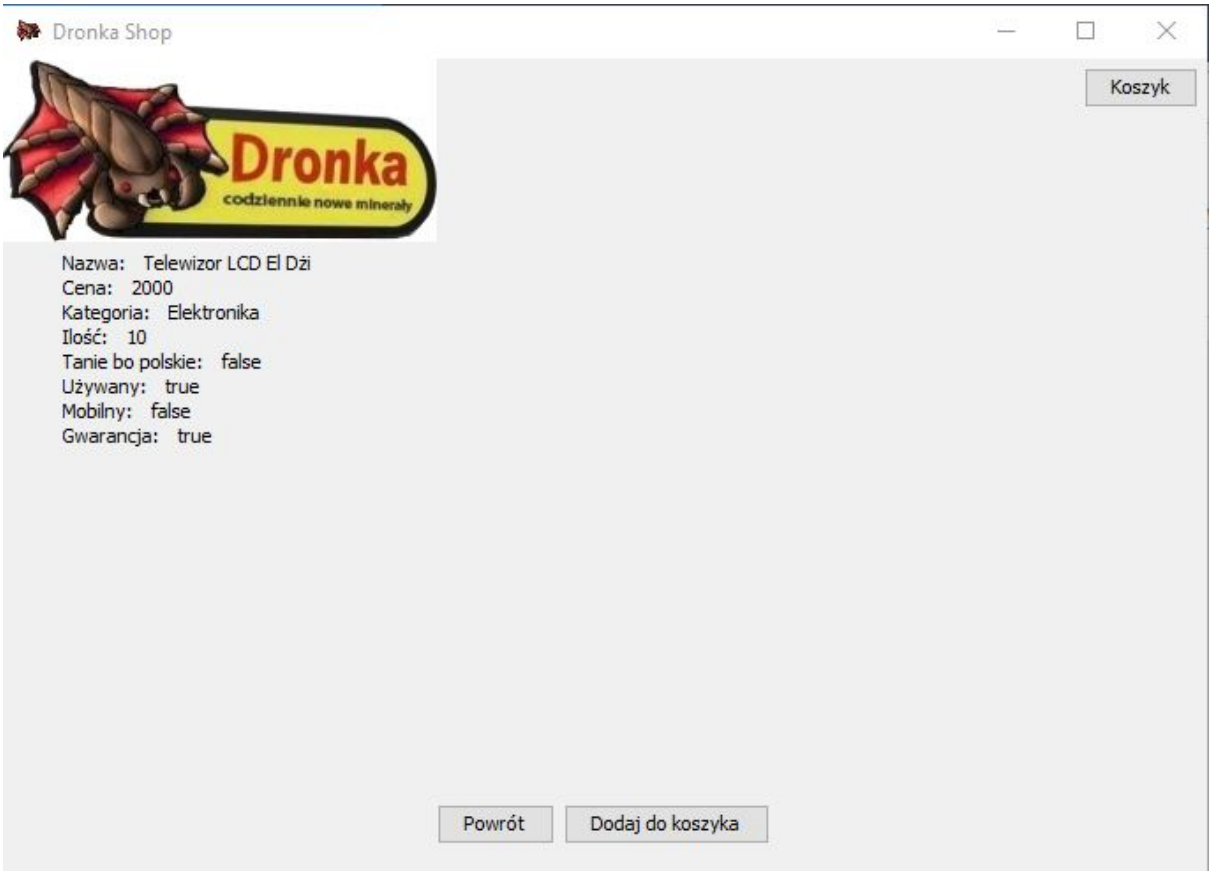
- music.csv

```
Nazwa,Cena,Ilość,Tanie bo polskie,Używany,Gatunek muzyczny,Dołączone
wideo
Ciley Myrus - Big Ball of Mud,60,20,TRUE,FALSE,Jazz,TRUE
```

e) Przykładowe screeny

- Książki

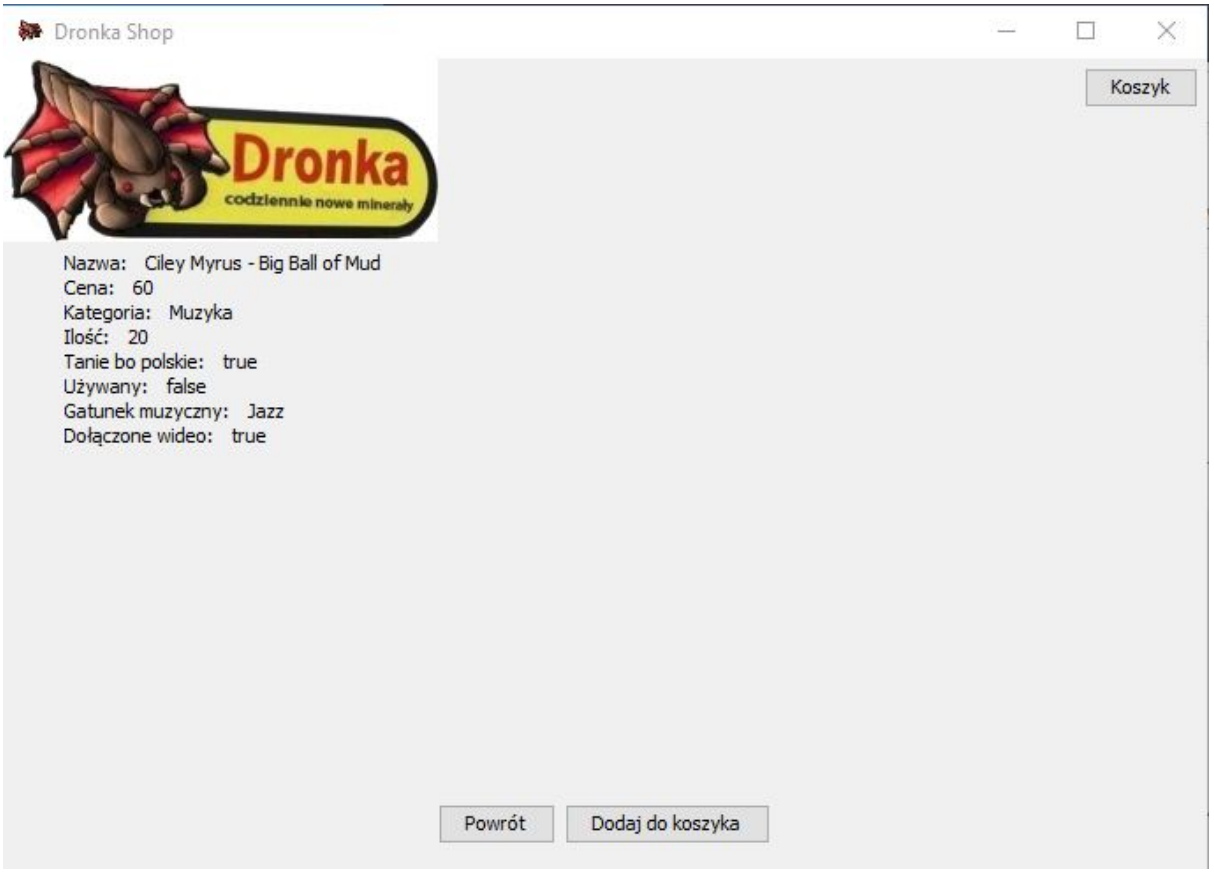
- Elektronika


Dronka Shop
Koszyk
Dronka
codziennie nowe minerały
Nazwa: Telewizor LCD El Dżi
Cena: 2000
Kategoria: Elektronika
Ilość: 10
Tanie bo polskie: false
Używany: true
Mobilny: false
Gwarancja: true
Powrót
Dodaj do koszyka

- Jedzenie


Dronka Shop
Koszyk
Dronka
codziennie nowe minerały
Nazwa: Sznycel mrożony
Cena: 6
Kategoria: Żywność
Ilość: 30
Tanie bo polskie: true
Używany: false
Data przydatności: Sat Jun 06 00:00:00 CEST 2020
Powrót
Dodaj do koszyka

- Muzyka



2. **Rozszerzenie panelu**
    a) Klasa ItemFilter została zmieniona tak, żeby było możliwe filtrowanie produktów po właściwościach typu boolean, specyficznych dla danej kategorii.

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

import pl.edu.agh.dronka.shop.model.*;


public class ItemFilter<T extends  Item> {


  private T itemSpec;


  public ItemFilter(T iSpec){
     itemSpec=iSpec;
  }
  public T getItemSpec() {
     return itemSpec;
```

```
  }


  public boolean appliesTo(Item item) {
     Category itemCategory=itemSpec.getCategory();
     if (itemSpec.getName() != null
           && !itemSpec.getName().equals(item.getName())) {
        return false;
     }
     if (itemSpec.getCategory() != null
           && !itemSpec.getCategory().equals(item.getCategory())) {
        return false;
     }

     if (itemSpec.isSecondhand() && !item.isSecondhand()) {
        return false;
     }

     if (itemSpec.isPolish() && !item.isPolish()) {
        return false;
     }

     if (itemCategory==Category.BOOKS && ((Books)itemSpec).hasHardCover() &&
!((Books)item).hasHardCover()) {
        return false;
     }

     if (itemCategory==Category.ELECTRONICS &&
((Electronics)itemSpec).isMobile() && !((Electronics)item).isMobile()) {
        return false;
     }
```

```
     if (itemCategory==Category.ELECTRONICS &&
((Electronics)itemSpec).hasWarranty() && !((Electronics)item).hasWarranty())
{

        return false;

     }


     if (itemCategory==Category.MUSIC && ((Music)itemSpec).isVideoIncluded()
&& !((Music)item).isVideoIncluded()) {

        return false;

     }


     return true;

  }


}
```

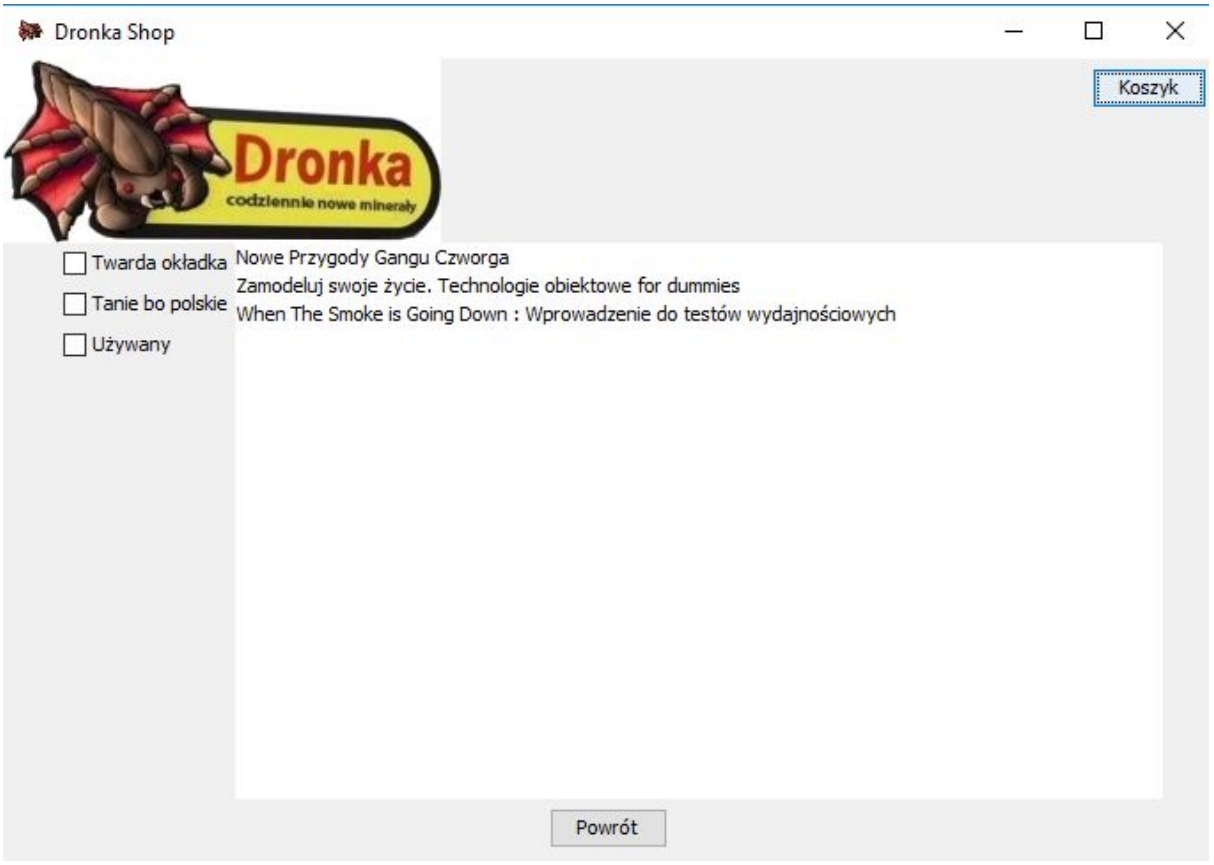
b) Również metoda `fillProperties()` klasy `PropertiesPanel` została zmodyfikowana

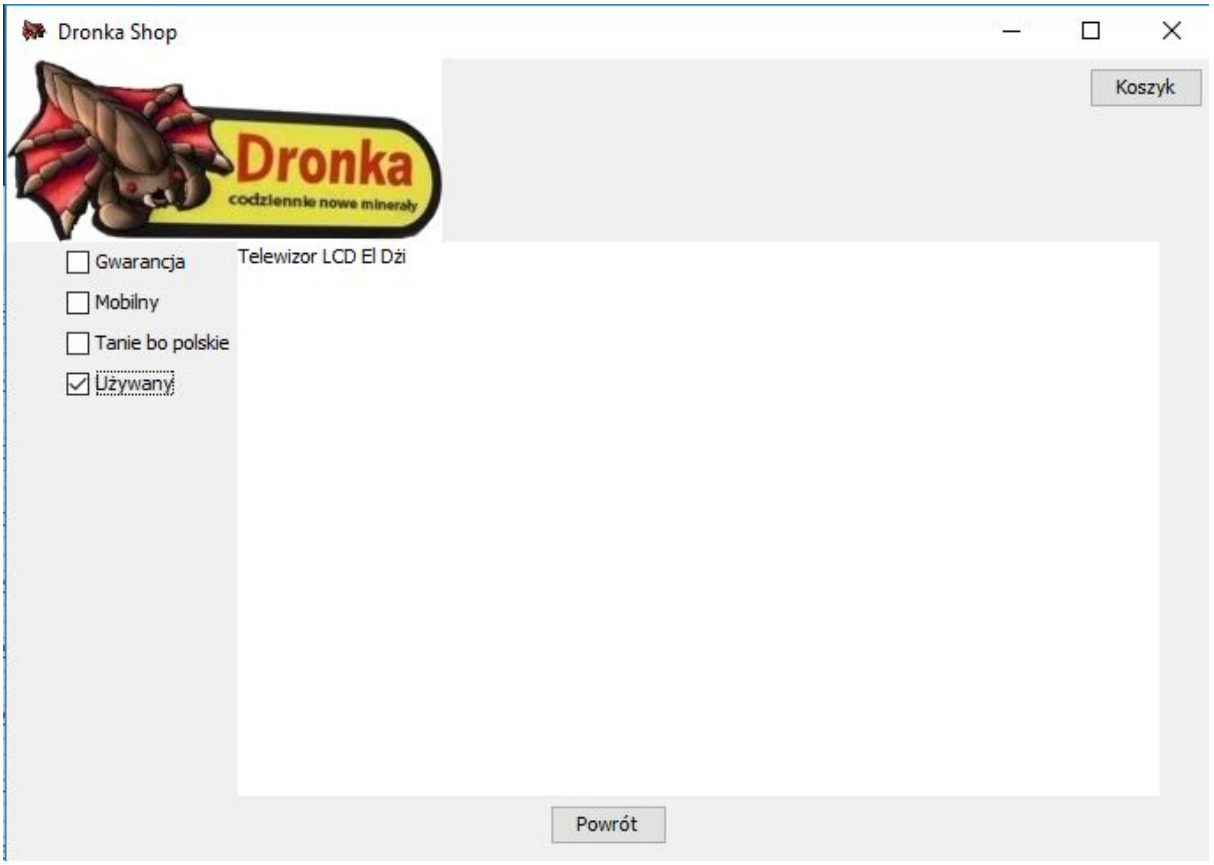
```
public void fillProperties() {

  removeAll();

  Category itemCategory=shopController.getCurrentCategory();


  if(itemCategory== Category.BOOKS){

     filter = new ItemFilter<Books>(new Books());

     filter.getItemSpec().setCategory(itemCategory);


     add(createPropertyCheckbox("Twarda okładka", new ActionListener() {


        @Override
```

```
        public void actionPerformed(ActionEvent event) {
            ((Books)filter.getItemSpec()).setHasHardCover(
                    ((JCheckBox) event.getSource()).isSelected());
            shopController.filterItems(filter);
        }
    }));
}

if(itemCategory== Category.ELECTRONICS){
    filter = new ItemFilter<Electronics>(new Electronics());
    filter.getItemSpec().setCategory(itemCategory);

    add(createPropertyCheckbox("Gwarancja", new ActionListener() {

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

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

```
if(itemCategory== Category.MUSIC){
   filter = new ItemFilter<Music>(new Music());
   filter.getItemSpec().setCategory(itemCategory);

   add(createPropertyCheckbox("Dołączone wideo", new ActionListener() {

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

if(itemCategory== Category.SPORT || itemCategory==Category.FOOD){
   filter = new ItemFilter<Item>(new Item());
   filter.getItemSpec().setCategory(itemCategory);
}



add(createPropertyCheckbox("Tanie bo polskie", new ActionListener() {

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

```
  add(createPropertyCheckbox("Używany", new ActionListener() {


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


Dronka Shop
Koszyk
Dronka
codziennie nowe minerały
Twarda okładka
Tanie bo polskie
Używany
Nowe Przygody Gangu Czworga
Zamodeluj swoje życie. Technologie obiektowe for dummies
When The Smoke is Going Down : Wprowadzenie do testów wydajnościowych
Powrót


Dronka Shop
Koszyk
Dronka
codziennie nowe minerały
Gwarancja
Mobilny
Tanie bo polskie
Używany
Telewizor LCD El Dżi
Powrót


Dronka Shop
Koszyk
Dronka
codziennie nowe minerały
Gwarancja
Mobilny
Tanie bo polskie
Używany
Telewizor LCD El Dżi
Powrót