

Raport z laboratorium nr 2

Bartłomiej Plewnia

Grupa: piątek 12:50 – 14:20 B

Zad 2.1

Zmiany w projekcie:

- Stworzenie klas dziedziczących z klasy *Item*:

- BookItem – zawiera dodatkowo atrybuty: numberOfPages oraz hardCover

```
public class BookItem extends Item {  
  
    private int numberOfPages;  
    private boolean hardCover;  
  
    public BookItem(String name, Category category, int price, int quantity, int numberOfPages, boolean hardCover){  
        super(name, category, price, quantity);  
        this.numberOfPages=numberOfPages;  
        this.hardCover=hardCover;  
    }  
  
    public int getNumberOfPages(){  
        return this.numberOfPages;  
    }  
  
    public void setNumberOfPages(int numberOfPages){  
        this.numberOfPages = numberOfPages;  
    }  
  
    public boolean isHardCover(){ return this.hardCover; }  
  
    public void setHardCover(boolean hardCover){  
        this.hardCover = hardCover;  
    }  
}
```

- ElectronicItem – zawiera dodatkowo atrybuty: mobile oraz quarantee

```
public class ElectronicItem extends Item{  
    private boolean mobile;  
    private boolean quarantee;  
  
    public ElectronicItem(String name, Category category, int price, int quantity, boolean mobile, boolean quarantee) {  
        super(name, category, price, quantity);  
        this.mobile = mobile;  
        this.quarantee = quarantee;  
    }  
  
    public boolean isMobile() {  
        return mobile;  
    }  
  
    public boolean isQuarantee() {  
        return quarantee;  
    }  
}
```

- FoodItem – zawiera dodatkowo atrybut expiryDate oraz zmieniony getter getExpiryDate() zwracający datę w postaci przejrzystej

```
public class FoodItem extends Item{
    private Date expiryDate;

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

    public String getExpiryDate() {
        return new SimpleDateFormat("dd-MM-yyyy").format(expiryDate);
    }

    public void setExpiryDate(Date expiryDate) {
        this.expiryDate = expiryDate;
    }
}
```

- MusicItem – zawiera dodatkowo atrybuty: attachedVideo oraz musicGenre

```
public class MusicItem extends Item{
    private MusicGenre musicGenre;
    private boolean attachedVideo;

    public MusicItem(String name, Category category, int price, int quantity, String musicGenre, boolean attachedVideo) {
        super(name, category, price, quantity);
        this.musicGenre = MusicGenre.StringToGenre(musicGenre);
        this.attachedVideo = attachedVideo;
    }

    public void setMusicGenre(MusicGenre musicGenre) { this.musicGenre = musicGenre; }

    public void setAttachedVideo(boolean attachedVideo) { this.attachedVideo = attachedVideo; }

    public MusicGenre getMusicGenre() {
        return musicGenre;
    }

    public boolean isAttachedVideo() { return attachedVideo; }
}
```

Atrybut musicGenre jest typu enum MusicGenre:

```
public enum MusicGenre {
    POP( name: "POP"),
    ROCK( name: "ROCK"),
    HOUSE( name: "HOUSE"),
    DUBSTEP( name: "DUBSTEP"),
    COUNTRY( name: "COUNTRY"),
    LATIN( name: "LATIN");

    private String infoName;

    MusicGenre(String name) {
        this.infoName = name;
    }

    public String getInfoName() {
        return infoName;
    }

    public static MusicGenre StringToGenre(String name){
        switch(name.toUpperCase()){
            case "POP":
                return MusicGenre.POP;
            case "ROCK":
                return MusicGenre.ROCK;
            case "HOUSE":
                return MusicGenre.HOUSE;
            case "DUBSTEP":
                return MusicGenre.DUBSTEP;
            case "COUNTRY":
                return MusicGenre.COUNTRY;
            case "LATIN":
```

- SportItem – nie zawiera dodatkowych atrybutów

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

- Aby klasy dziedziczące mogły używać atrybutów name oraz category, zmieniłem modyfikator z private na protected

```
public class Item {  
  
    protected String name;  
  
    protected Category category;  
  
    private int price;  
  
    private int quantity;  
  
    private boolean secondhand;  
  
    private boolean polish;  
  
    public Item(String name, Category category, int price, int quantity) {  
        this.name = name;  
        this.category = category;  
        this.price = price;  
        this.quantity = quantity;  
    }  
}
```

- W klasie ShopProvider zmodyfikowałem metodę readItems() tak, by zwracała listę produktów różnego typu.
(Załącznik nr 1)

- W klasie PropertiesHelper zmodyfikowałem metodę getPropertiesMap tak, aby zwracała mapę atrybutów dla poszczególnych produktów.

```
public class PropertiesHelper {

    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 BookItem) {
            propertiesMap.put("Liczba stron", ((BookItem) item).getNumberOfPages());
            propertiesMap.put("Twarda oprawa", ((BookItem) item).isHardCover());
        } else if (item instanceof ElectronicItem) {
            propertiesMap.put("Mobilny", ((ElectronicItem) item).isMobile());
            propertiesMap.put("Gwarancja", ((ElectronicItem) item).isQuarantee());
        } else if (item instanceof FoodItem) {
            propertiesMap.put("Data przydatności", ((FoodItem) item).getExpiryDate());
        } else if (item instanceof MusicItem) {
            propertiesMap.put("Gatunek", ((MusicItem) item).getMusicGenre());
            propertiesMap.put("Dołączone Video", ((MusicItem) item).isAttachedVideo());
        }

        return propertiesMap;
    }
}
```

- Dodałem do poszczególnych plików CSV dodatkowe cechy produktów

```
food.csv x music.csv x
1 Nazwa,Cena,Ilość,Tanie bo polskie,Używany,Gatunek,Dołączone Video
2 Ciley Myrus - Big Ball of Mud,60,20,TRUE,FALSE,POP,TRUE
3 Agnieszka Chylińska - Nie mogę Cię zapomnieć,10,1000,TRUE,FALSE,POP,FALSE
```

```
food.csv x
1 Nazwa,Cena,Ilość,Tanie bo polskie,Używany,Data przydatności
2 Zupa Studencka Instant,2,100,TRUE,FALSE, 10-10-2021
3 Sznyceł mrożony,6,30,TRUE,FALSE, 09-09-2020
4 Pierogi Giza,10,25,FALSE,FALSE,24-12-2024
5
```

Efekt końcowy:





Koszyk

Nazwa: Pierožki Giza
Cena: 10
Kategoria: Żywność
Ilość: 25
Tanie bo polskie: false
Używany: false
Data przydatności: 24-12-2024

Powrót

Dodaj do koszyka



Zad 2.2

Brak zadania nr 2.2