Online Filmverleih

Frank Hedecke

Voraussetzungen:	Testing
Zeithorizont:	45 Minuten
Lernziele:	fremden Quelltext verstehen

1 Beschreibung

Das Programm implementiert einen Online Filmverleih. Kunden können sich dort anmelden, Filme erstehen und diese danch streamen.

2 Aufgaben

1. Erstelle ein neues Eclipse Projekt und füge die fünf Klassen ein.
2. Lies dir den Quelltext durch.
3. Erstelle eine Testklasse für die Klasse *Person* und teste das Programm ausgiebig.

Z. Finde einen Fehler um in der Rolle einer *Person* den Filmverleih zu betrügen.

3 Hinweise

- Du kannst natürlich auch Testklassen für die anderen Klassen erstellen.
- Teste erst einmal die einfachen Aktionen, wie Anmelden bei einem *Shop*.
- Vergiss nicht den *Shop* mit Filmen zu bestücken.
- Fehler im Quelltext bitte an `mailto:frank@ifsr.de`
  Zeige den Fehler bitte vorher deinem Tutor.

4 Programm

```java
import java.util.HashMap;
import java.util.Map;

public class Catalog {

    private Map<String, Movie> movies;

    public Catalog() {
        this.movies = new HashMap<String, Movie>();
    }

    public boolean addTitle(Movie movie) {
        if (this.movies.keySet().contains(movie.toString())) {
            return false;
        } else {
            this.movies.put(movie.toString(), movie);
            return true;
        }
    }

    public Movie search(String title) {
        if (movies.keySet().contains(title)) {
            return this.movies.get(title);
        } else {
            return null;
        }
    }
}
```

```java
public class Customer {

    private Catalog movies;
    private Person user;

    public Customer(Person user) {
        this.movies = new Catalog();
        this.user = user;
    }

    public boolean addMovie(Movie movie) {
        return this.movies.addTitle(movie);
    }

    public boolean hasMovie(Movie movie) {
        Movie mov = this.movies.search(movie.toString());
        if (mov == null) {
            return false;
        } else {
            return true;
        }
    }

    public int getMoney() {
        return this.user.getMoney();
    }

    public void pay(int amount) {
        this.user.pay(amount);
    }

    @Override
    public String toString() {
        return user.toString();
    }
}
```

```java
public class Movie {

    private String title;
    private int price;

    public Movie(String title, int price) {
        this.title = title;
        this.price = price;
    }

    public int getPrice() {
        return this.price;
    }

    @Override
    public String toString() {
        return this.title;
    }
}
```

```java
public class Person {

    private int money;
    private int customerID;
    private String name;

    public Person(String name) {
        this.name = name;
        this.money = 100;
    }

    public void register(Shop shop) {
        this.customerID = shop.becomeCustomer(this);
    }

    public void pay(int amount) {
        this.money -= amount;
    }

    public int getMoney() {
        return this.money;
    }

    public boolean buy(Shop shop, Movie movie) {
        return shop.buy(movie, this.customerID);
    }

    public boolean watch(Shop shop, Movie movie) {
        return shop.stream(movie, this.customerID);
    }

    @Override
    public String toString() {
        return this.name;
    }
}
```

```java
import java.util.LinkedList;
import java.util.List;

public class Shop {

    private List<Customer> customers;
    private Catalog availableMovies;
    private String name;

    public Shop(String name) {
        this.customers = new LinkedList<Customer>();
        this.availableMovies = new Catalog();
        this.name = name;
    }

    public int becomeCustomer(Person newUser) {
        this.customers.add(new Customer(newUser));
        System.out.println(newUser + " is now a customer from " + this.
    name);
        return this.customers.size() - 1;
    }

    public boolean addMovie(Movie movie) {
        return this.availableMovies.addTitle(movie);
    }

    public Movie search(String title) {
        return this.availableMovies.search(title);
    }

    public boolean buy(Movie movie, int customerID) {
        int price = movie.getPrice();
        Customer c = this.customers.get(customerID);

        if (c.getMoney() < movie.getPrice()) {
            System.out.println(c + " can't afford to buy " + movie);
            return false;
        }

        if (c.hasMovie(movie)) {
            System.out.println(c + " already has " + movie);
            return false;
        }

        c.pay(price);
        c.addMovie(movie);

        System.out.println(c + " buys " + movie + " for just " + movie.
    getPrice());

        return true;
    }

    public boolean stream(Movie movie, int customerID) {
        Customer c = this.customers.get(customerID);

        if (! c.hasMovie(movie)) {
            System.out.println(c + " can't stream " + movie);
            return false;
        }

        System.out.println(c + " watches " + movie);

        return true;
    }
}
```