Object Oriented System Design

Assignment #2

Game cafe: Part 2

- This assignment uses the code base you created in the first assignment.
- Game types (board, card, electronic)
- I/O File handling
- Exception handling
- Saving game state

Abstract Class Game

Variables:

- private String name
- private double price
- **protected** **<u>int</u>** quality // 0-100

Methods:

- Game(String name, double price, **int quality**)
- public String toString(); // Returns game's
  name, quality and price
- public void repair(); // Sets the quality of
  the game to 100
- public abstract double getRepairCost();
- public abstract void lowerQuality(); // Lowers the quality of the game
- public abstract String getQuality();

Class BoardGame: Extends Game

Methods:

- public double getRepairCost(); // $0.04 per quality point
- public void lowerQuality(); // Lowers the quality of the game by 25 points
- public String getQuality(); // Quality > 70 == Good, 70-50 == Okay, < 50 == Bad

Class CardGame: Extends Game

Methods:

- public double getRepairCost(); // $0.02 per quality point
- public void lowerQuality(); // Lowers the quality of the game by 30 points
- public String getQuality(); // Quality > 60 == Good, 60-25 == Okay, < 25 == Bad

Class ElectronicGame: Extends Game

Methods:

- public double getRepairCost(); // $0.06 per quality point
- public void lowerQuality(); // Lowers the quality of the game by 20 points
- public String getQuality(); // Quality > 80 == Good, 80-60 == Okay, < 60 == Bad

Class CSVLoader

Methods:

- public Game[] loadGames(String fileName);   // create games based on a CSV file
- public void saveGames(Game[] games, String fileName);   // write games to a CSV file

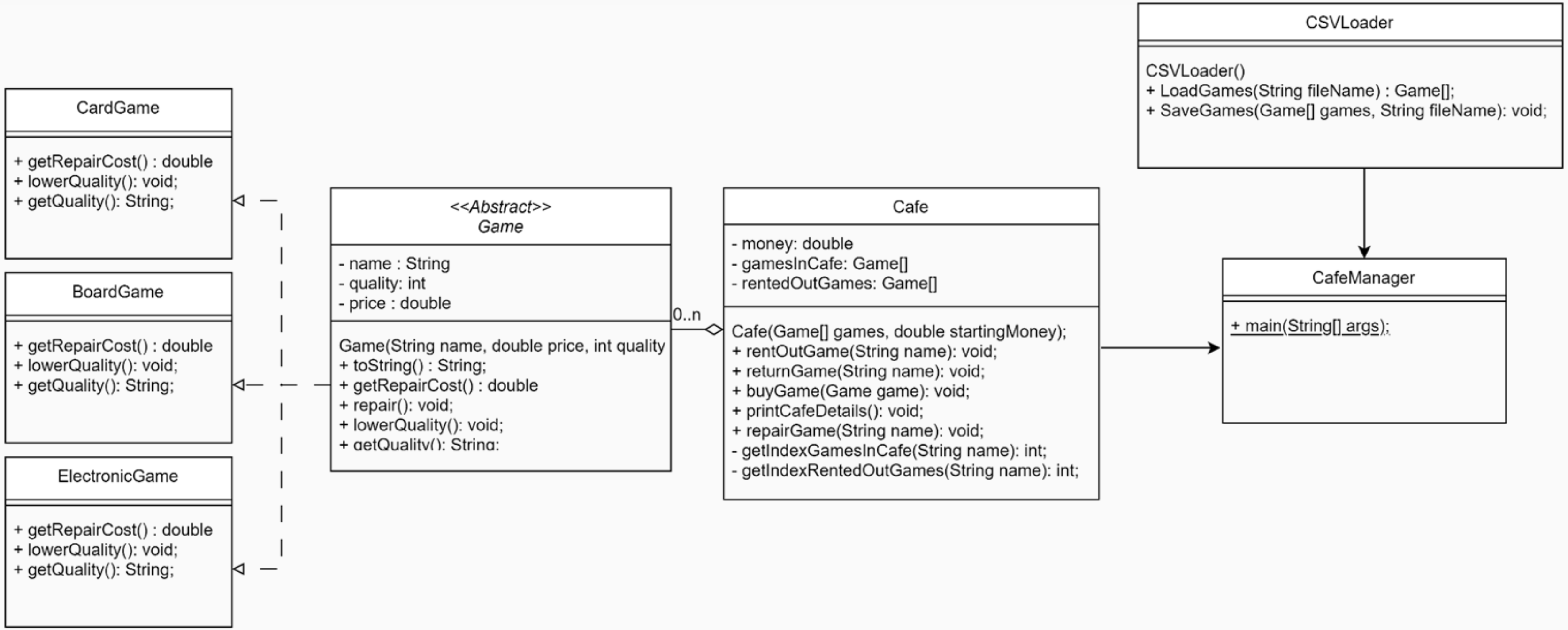
Example files will be available in the assignment description.

Main class

method main:

- Initialize the CSVLoader
- Initialize the Cafe
- Rent out a game until it’s quality is ‘bad’ and repair it.
- Rent out more games until you have enough money to buy a new one (think of one yourself).
- Buy a new game.

UML



CardGame

+ getRepairCost() : double
+ lowerQuality(): void;
+ getQuality(): String;

BoardGame

+ getRepairCost() : double
+ lowerQuality(): void;
+ getQuality(): String;

ElectronicGame

+ getRepairCost() : double
+ lowerQuality(): void;
+ getQuality(): String;

<<Abstract>>
Game

- name : String
- quality: int
- price : double

Game(String name, double price, int quality
+ toString() : String;
+ getRepairCost() : double
+ repair(): void;
+ lowerQuality(): void;
+ getQuality(): String;

0..n

Cafe

- money: double
- gamesInCafe: Game[]
- rentedOutGames: Game[]

Cafe(Game[] games, double startingMoney);
+ rentOutGame(String name): void;
+ returnGame(String name): void;
+ buyGame(Game game): void;
+ printCafeDetails(): void;
+ repairGame(String name): void;
- getIndexGamesInCafe(String name): int;
- getIndexRentedOutGames(String name): int;

CSVLoader

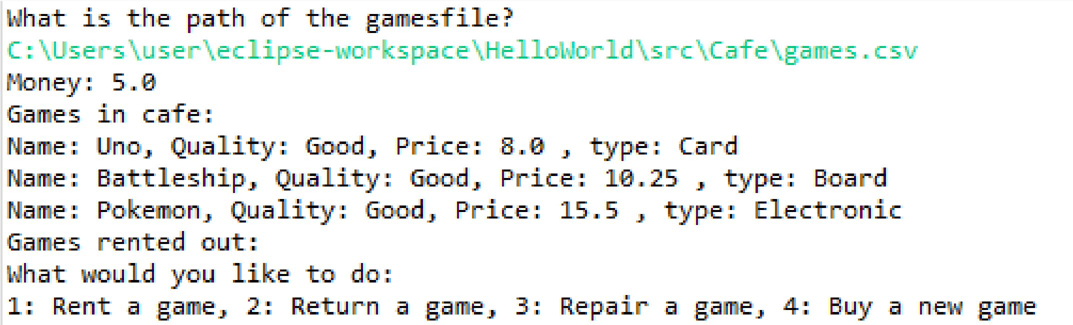
CSVLoader()
+ LoadGames(String fileName) : Game[];
+ SaveGames(Game[] games, String fileName): void;

CafeManager

+ main(String[] args);

Visualization

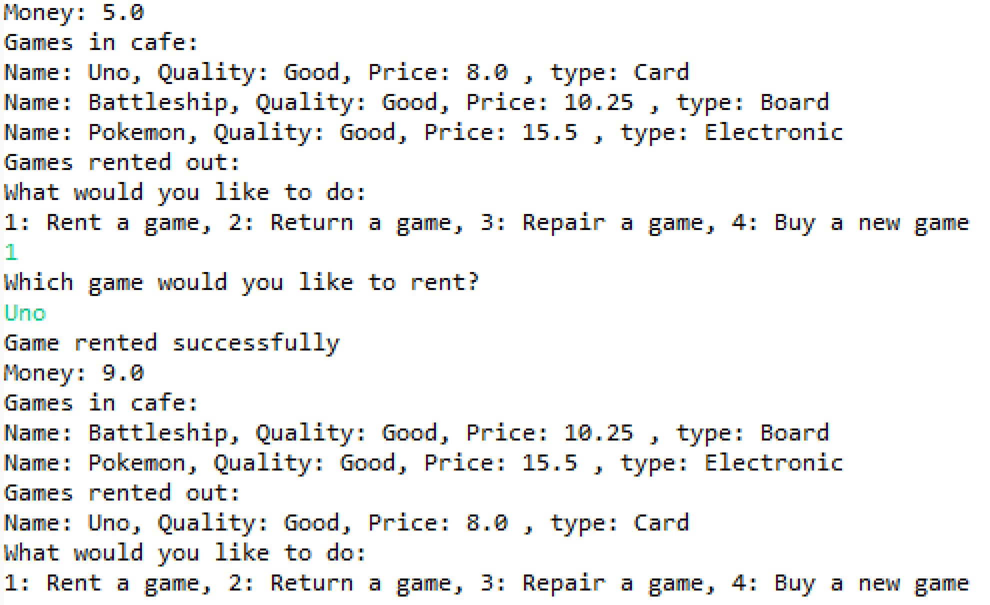
Initialization



```
What is the path of the gamesfile?
C:\Users\user\eclipse-workspace\HelloWorld\src\Cafe\games.csv
Money: 5.0
Games in cafe:
Name: Uno, Quality: Good, Price: 8.0 , type: Card
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Games rented out:
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
```

Visualization

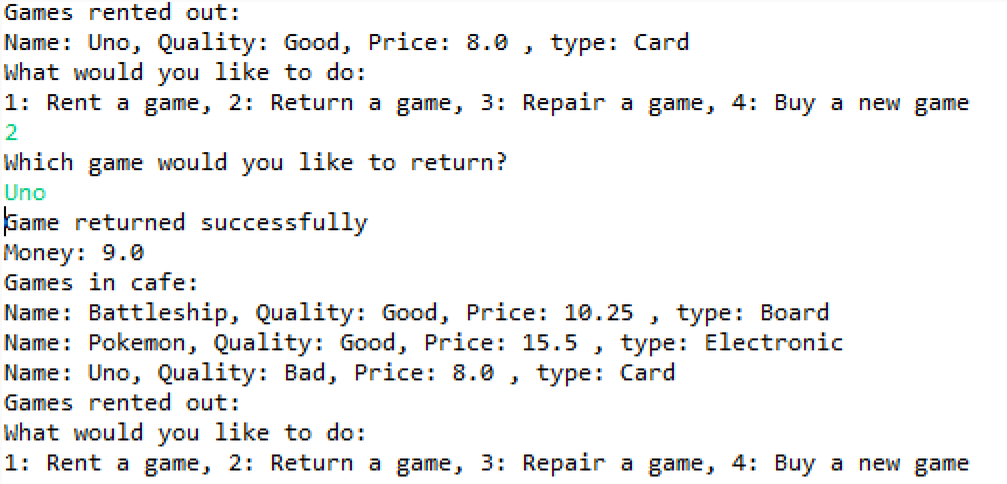
Renting a game



```
Money: 5.0
Games in cafe:
Name: Uno, Quality: Good, Price: 8.0 , type: Card
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Games rented out:
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
1
Which game would you like to rent?
Uno
Game rented successfully
Money: 9.0
Games in cafe:
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Games rented out:
Name: Uno, Quality: Good, Price: 8.0 , type: Card
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
```

Visualization

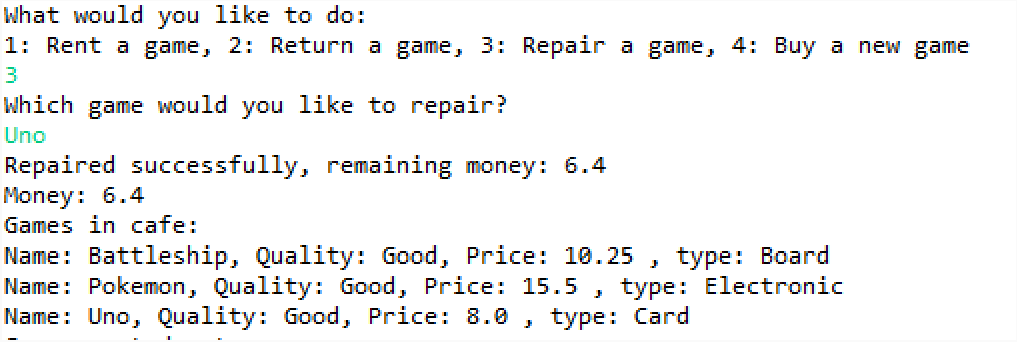
Returning a game



```
Games rented out:
Name: Uno, Quality: Good, Price: 8.0 , type: Card
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
2
Which game would you like to return?
Uno
Game returned successfully
Money: 9.0
Games in cafe:
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Name: Uno, Quality: Bad, Price: 8.0 , type: Card
Games rented out:
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
```

Visualization

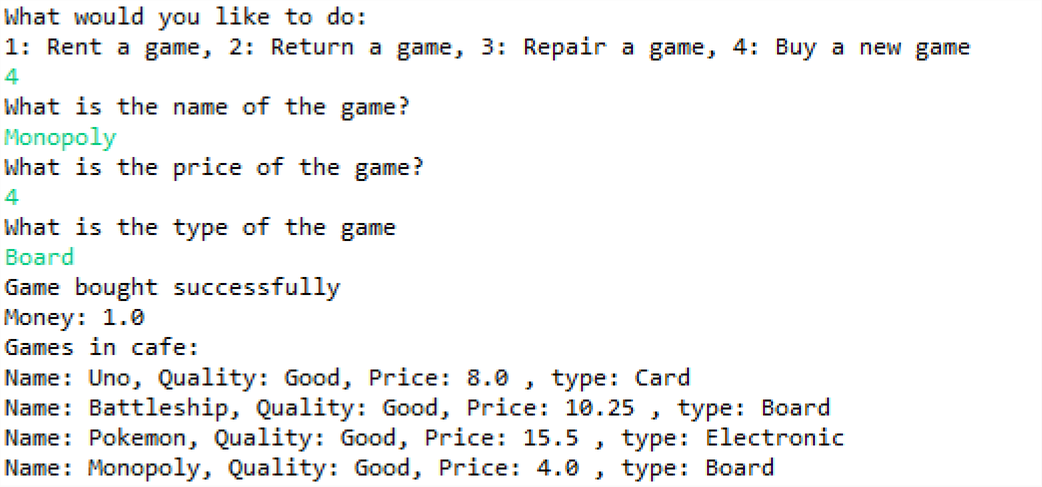
Repairing a game



```
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
3
Which game would you like to repair?
Uno
Repaired successfully, remaining money: 6.4
Money: 6.4
Games in cafe:
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Name: Uno, Quality: Good, Price: 8.0 , type: Card
```

Visualization

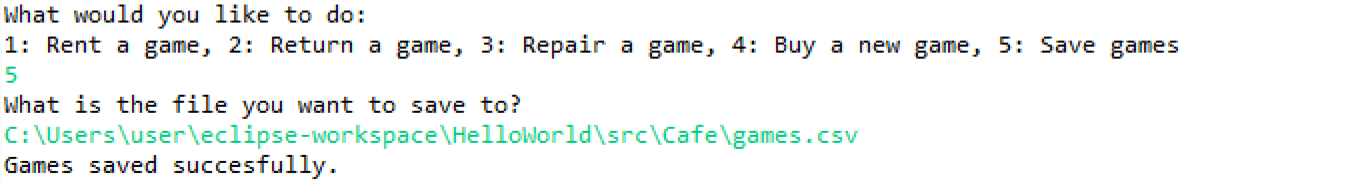
Buying a game



```
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game
4
What is the name of the game?
Monopoly
What is the price of the game?
4
What is the type of the game
Board
Game bought successfully
Money: 1.0
Games in cafe:
Name: Uno, Quality: Good, Price: 8.0 , type: Card
Name: Battleship, Quality: Good, Price: 10.25 , type: Board
Name: Pokemon, Quality: Good, Price: 15.5 , type: Electronic
Name: Monopoly, Quality: Good, Price: 4.0 , type: Board
```

Visualization

Saving games



```
What would you like to do:
1: Rent a game, 2: Return a game, 3: Repair a game, 4: Buy a new game, 5: Save games
5
What is the file you want to save to?
C:\Users\user\eclipse-workspace\HelloWorld\src\Cafe\games.csv
Games saved succesfully.
```

Rules

- Only regular arrays [], no Lists or Maps.
- Use getters and setters when needed and only make them public if they are actually meant to be used from outside the class.
- Reading code is more difficult than writing code, place comments in complex methods.

Rules (2)

- You are not allowed to add extra arrays for the games, make use of the existing gamesInCafe and rentedOutGames arrays.
- You are not allowed to use external libraries to read the CSV file.
- The CSVLoader has to utilize exception handling
  - The system should not crash when the given filename is not found.
  - The system should not crash when the data in the file is not well formatted (ex: empty file, wrong data types).

Submission

Submit your program code package(Assignment2_code_이름_학번) and design document (Assigment2_이름_학번) in LMS. (Korean/English)

The design document describes the main methods, flow of the code and overall design of the program. It should also mention and explain **every** method that you added that was not described in this powerpoint.

Deadline: 19th of may (5월 19일) 23:59

Late penalties:

Up to 24 hours = -25%,

24-48 hours = -50%,

48-72 hours = -75%

after 72 Hours you get an automatic 0.