Name: Farid Gurbanov

Neptun: F1GTTQ

Subject: Programming Technology

Task: Assignment 1 / Task 4

Documentation

1. Task

Simulate a simplified Capitaly game. There are some players with different strategies, and a cyclical board with several fields. Players can move around the board, by moving forward with the amount they rolled with a dice. A field can be a property, service, or lucky field. A property can be bought for 1000, and stepping on it the next time the player can build a house on it for 4000. If a player steps on a property field which is owned by somebody else, the player should pay to the owner 500, if there is no house on the field, or 2000, if there is a house on it. Stepping on a service field, the player should pay to the bank (the amount of money is a parameter of the field). Stepping on a lucky field, the player gets some money (the amount is defined as a parameter of the field). There are three different kind of strategies exist. Initially, every player has 10000.

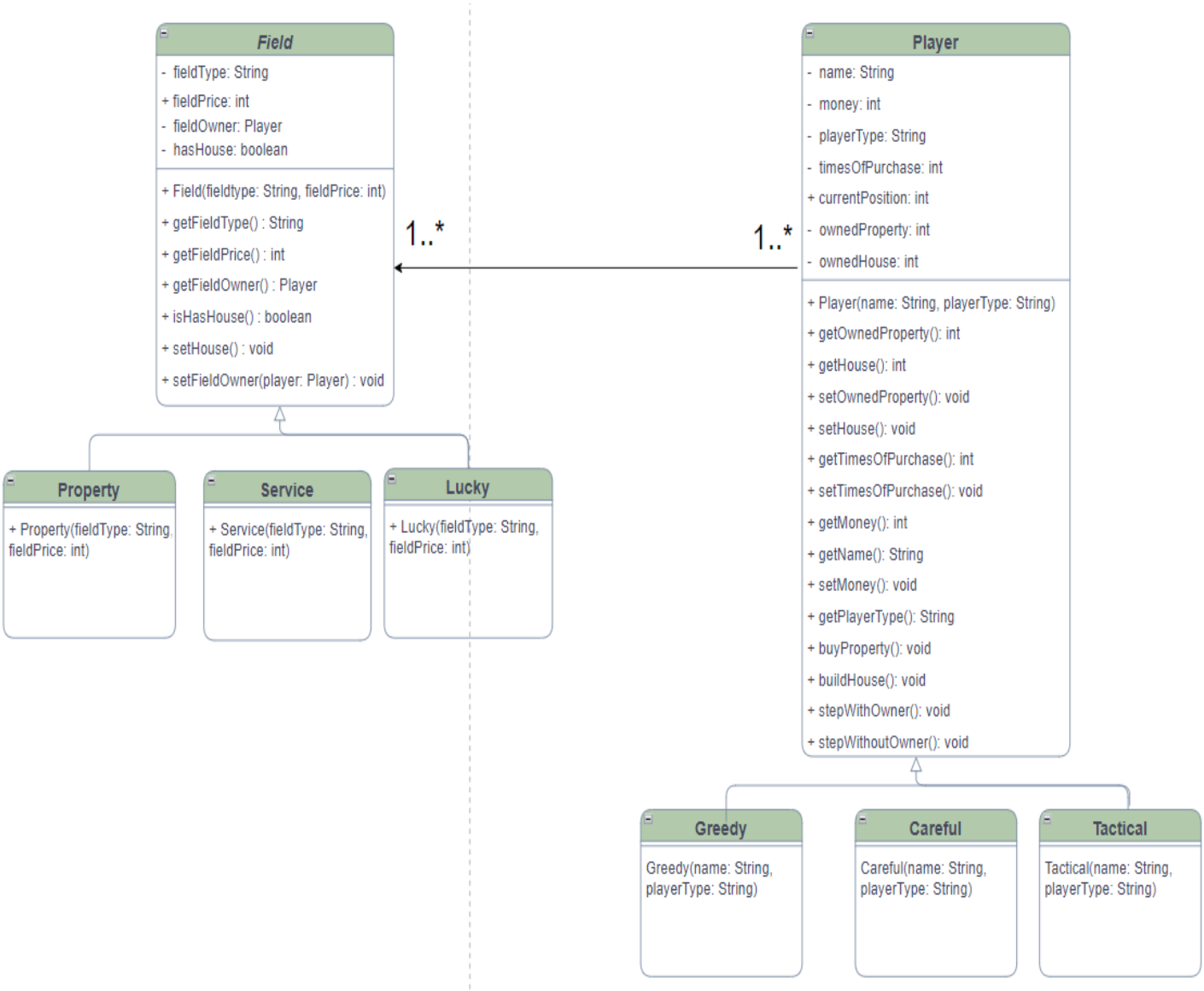
Greedy player: If he steps on an unowned property, or his own property without a house, he starts buying it, if he has enough money for it.
Careful player: he buys in a round only for at most half the amount of his money.
Tactical player: he skips each second chance when he could buy.

If a player has to pay, but he runs out of money because of this, he loses. In this case, his properties are lost, and become free to buy.

Read the parameters of the game from a text file. This file defines the number of fields, and then defines them. We know about all fields: the type. If a field is a service or lucky field, the cost of it is also defined. After the these parameters, the file tells the number of the players, and then enumerates the players with their names and strategies. In order to prepare the program for testing, make it possible to the program to read the roll dices from the file.

Print out which player won the game, and how rich he is (balance, owned properties).

2. Class Diagram



3. Explanation of Methods

- **get…(), isHasHouse()** -> getters for the attributes of **Player** and **Field** classes.

- **Player(), Greedy(), Careful(), Tactical(), Field(), Property(), Service(), Lucky()** -> constructors for my parent and children classes, the children classes inherits the parent's constructor via **super()** keyword.

- **setHouse()** -> setting true if the player buys a house in a certain property

- **setFieldOwner()** -> if the player buys a property, this method sets him as its owner

- **setOwnedProperty(), setHouse(), setTimesOfPurchase()** -> increments the number of properties the player has, the number of houses the player has, times of purchase to decide if tactical player have to pay or not for the property

- **buyProperty(), buildHouse(), stepWithOwner(), stepWithoutOwner()** -> methods to withdraw money from player's account respectively based on the words in methods' names

- **setMoney()** -> accepts a parameter as an amount to pay or to receive based on the field prices of service & lucky

4. Testing

To test my program, I used 6 input files, and the outputs should be as following:

- *input.txt ->* **Result:** *David: 13000 Owned Property: 1 Owned House: 0*
- *input2.txt ->* **Result:** *Kanan: 11000 Owned Property: 0 Owned House: 0*
- *input3.txt ->* **Result:** *Ragnar: 12000 Owned Property: 0 Owned House: 0*
- *input4.txt ->* **Result:** *Throws NoSuchElementException()*
- *input5.txt ->* **Result:** *David: 18000 Owned Property: 1 Owned House: 0*
- *input6.txt ->* **Result:** *Throws InvalidInputException()*

Note: If I pass the file which does not exist as a parameter, it will throw *FileNotFoundException().*