Name : Hajiyev Hajiaga **1 assignment 4th task** **7 October 2024**

Naptun code: C499UD

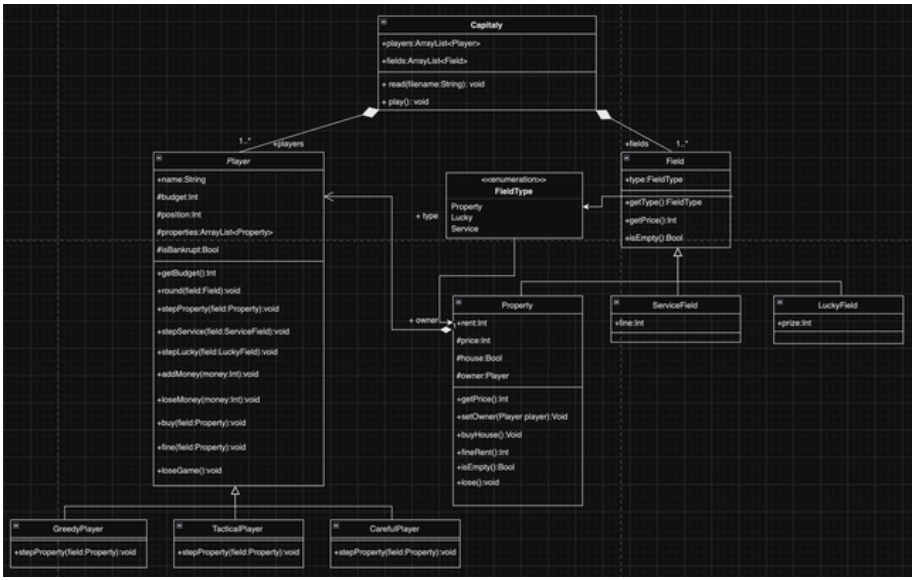
Task

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).

UML Class Diagram



Description of major methods

The methods in Player

1. Player(String name): Constructor that initializes the player with a name, default budget, starting position, and empty property list.
2. getBudget(): Returns the player's budget.
3. round(Field field): Determines actions based on the field type (Property, Service, Lucky).
4. stepProperty(Property field): Placeholder for property interaction.
5. stepService(ServiceField field): Reduces the player's money by a fine.
6. stepLucky(LuckyField field): Increases the player's budget by a prize amount.
7. addMoney(Integer money): Adds money to the player's budget.
8. loseMoney(Integer money): Deducts money from the player's budget, sets the player as bankrupt if they don't have enough money.
9. buy(Property field): Reduces the player's budget to buy a property and adds it to the player's list.
10. fine(Property field): Pays rent when landing on another player's property.
11. loseGame(): Clears the player's properties if they lose the game.

The methods in CarefulPlayer

1. CarefulPlayer(String name): Constructor that initializes a careful player with a given name, inheriting from the Player class.
2. stepProperty(Property field):

If the player already owns the property: The player buys a house if they can afford it (if budget/2 > 4000 and the property doesn't have a house).

If the player doesn't own the property and it's empty: They buy it only if they have more than twice the price of the property.

If the property is owned by someone else:The player pays rent (fine) to the owner.

The methods in GreedyPlayer

1. GreedyPlayer(String name): Constructor that initializes a greedy player with a given name, inheriting from the Player class.
2. stepProperty(Property field):

If the player owns the property:Buys a house if their budget is greater than 4000 and the property doesn't already have one.

If the player doesn't own the property and it's empty: Buys the property if they can afford it.

If the property is owned by someone else: Pays rent (fine) to the owner.

The methods in TacticalPlayer

1. TacticalPlayer(String name): Constructor that initializes a tactical player with a given name and a chance counte (countchance) set to 1.
2. stepProperty(Property field):

If the player owns the property: Buys a house if it doesn't already have one, their budget exceeds 4000, and the countchance is odd.

If the player doesn't own the property and it's empty: Buys the property under the same conditions.

If the property is owned by someone else: Pays rent (fine) to the owner.

The methods in Field

1. Field(FieldType type): Constructor that initializes the field with a specific type (passed as FieldType).
2. getType(): Returns the type of the field (e.g., property, service, lucky, etc.).
3. getPrice(): Placeholder method intended to return the price of the field. Currently, it returns null (to be implemented in a subclass)
4. isEmpty():Placeholder method meant to check if the field is empty. Returns null, likely to be implemented later.

The methods in LuckyField

1. LuckyField(FieldType type, Integer prize): Constructor that initializes a lucky field with a specific FieldType and a prize value. It passes the FieldType to the Field class via super(type).

The methods in Property

1. getPrice(): Returns the property price.
2. setOwner(Player player): Sets the owner of the property.
3. buyHouse(): Changes the property to include a house, increasing rent and deducting 4000 from the owner's budget.
4. fineRent(): Adds rent to the owner's budget and returns the rent value.
5. isEmpty(): Checks if the property is unowned.
6. lose(): Resets the property to its initial state if the owner loses.

The methods in ServiceField

1. ServiceField(FieldType type, Integer fine): Constructor that initializes the service field with a specific FieldType and a fine amount.

The methods in Capitaly

1. main(String[] args): Entry point that reads game configuration from a file and initiates the game.
2. read(String filename): Reads the game configuration from a file, initializing fields and players based on input data. Validates the input and throws InvalidInputException.
3. validateNonNegative(String input, String fieldName): Validates that an input string represents a non-negative integer. Throws InvalidInputException for invalid formats or negative values.
4. play(): Implements the game loop where players take turns rolling a die and moving across fields until only one player remains. Handles player actions and bankruptcy, and prints the game result.

Description of major methods

Blackbox testing

!!NOTE :: Expected output is always in the last line of input and code automatically checks if output is the same with desired output

The first input is "input1.txt" file

```
10
Property
Service 3100
Property
Property
Service 2200
Lucky 1000
Property
Property
Property
Service 2000
6
Maho Careful
Ramal Careful
Ismayil Tactical
Ayxan Greedy
Vusal Greedy
Arif Tactical
80
2 1 1 5 1 5 3 1 1 6 4 1 1 6 4 3 6 5 1 1 5 5 4 2 6 6 3 4 1 2 1 4 6 5 6 4 2 4 4 3 6 2 4 2 2 1 3 3 2 2 3 4 5 4 6 6 3 6 1 3 2 4 4 1 1 1 2 3 4 3 5 3 6 3 1 1 5 3 5 2
Name is: Ramal Last budget is: 1000 Number of properties is:  1
```

And the output is the same with desired output

```
Debugger Console   Run (Capitaly)
Manobought property
Ayxanpaid finefalse
Vusalpaid finefalse
Ismayilbought prop
Ayxanpaid finefalse
Vusalpaid finefalse
Mahopaid fine false
Ramalpaid fine false
Vusalpaid finefalse
Mahopaid fine false
Ismayilpaid fine false
Ismayil lost the game
Ayxanpaid finefalse
Ramalpaid fine false
Mahopaid fine false
Ramalpaid fine false
Vusalpaid finefalse
Ramalpaid fine false
Ayxanpaid finefalse
Vusalpaid finefalse
Mahopaid fine false
Ramalpaid fine false
Ayxan lost the game
Arifpaid fine false
Arif lost the game
Vusalpaid finefalse
Mahobought property
Ramalpaid fine false
Vusalpaid finefalse
Vusal lost the game
Ramalpaid fine false
Maho lost the game
Name is: Ramal Last budget is: 1000 Number of properties is:  1 If it is same: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.222 s
Finished at: 2024-10-09T10:59:55+02:00
------------------------------------------------------------------------
```

The first input is “input2.txt” file

```
input2.txt
8
Property
Lucky 500
Property
Service 3000
Service 2000
Lucky 200
Property
Property
6
Maho Careful
Ramal Careful
Ismayil Tactical
Ayxan Greedy
Vusal Greedy
Arif Tactical
69
2 4 6 4 1 6 3 4 1 5 3 5 5 2 3 5 1 5 5 2 2 1 5 2 3 2 5 6 4 6 2 3 6 6 2 3 2 3 6 5 1 1 6 4 5 6 6 4 1 6 4 5 4 1 4 1 3 2 5 5 3 2 6 3 6 2 4 2
Name is: Vusal Last budget is: 3400 Number of properties is:  1
```

And the output is the same with desired output

```
Output
Debugger Console   Run (Capitaly)
Mahobought house
Ramalpaid fine true
Ismayilpaid fine true
Ayxanpaid finefalse
Arifpaid fine false
Mahobought property
Ayxanpaid finefalse
Vusalpaid finetrue
Arifpaid fine true
Ramalpaid fine false
Vusalpaid finefalse
Arifpaid fine false
Ismayilpaid fine false
Vusalpaid finefalse
Mahopaid fine false
Ayxanpaid finefalse
Maho lost the game
Ismayilbought prop
Ayxanpaid finefalse
Vusalbought property
Arifpaid fine false
Ismayilpaid fine false
Ayxanpaid finefalse
Arif lost the game
Ramalpaid fine false
Ramal lost the game
Ayxanbought property
Vusalpaid finefalse
Ayxanpaid finefalse
Ismayil lost the game
Ayxan lost the game
Name is: Vusal Last budget is: 3400 Number of properties is:  1 If it is same: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.073 s
Finished at: 2024-10-09T11:03:20+02:00
------------------------------------------------------------------------
Output
```

The first input is “input3.txt” file

```
Lucky 500
Property
Service 3000
Service 2000
Lucky 200
Property
Property
Property
Service 2500
Service 5000
Lucky 900
Property
Property
6
Haji Careful
Ramal Careful
Philip Tactical
Ayxan Greedy
Ibo Greedy
Salmon Tactical
30
5 2 3 1 4 3 2 2 3 6 6 5 3 1 4 2 4 1 6 4 3 1 2 3 4 4 2 1 4 4
Name is: Ibo Last budget is: 1600 Number of properties is:  0
```

And the output is the same with desired output

```
Output
Debugger Console × | Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/Applications
Scanning for projects...

-----------------------< com.mycompany:Capitaly >-----------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Ramalbought property
Hajibought property
Philipbought prop
Ayxanpaid finefalse
Salmonbought prop
Philip lost the game
Salmon lost the game
Hajipaid fine false
Ramal lost the game
Ayxanbought property
Haji lost the game
Ayxanbought property
Ayxan lost the game
Name is: Ibo Last budget is: 1600 Number of properties is:  0 If it is same: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.886 s
Finished at: 2024-10-09T11:13:46+02:00
------------------------------------------------------------------------
Output
```

The first input is “input4.txt” file

```
input4.txt
10
Property
Service 3000
Lucky 500
Property
Service 4000
Lucky 700
Property
Service 2000
Lucky 1000
Property
3
Alice Greedy
Bob Careful
Charlie Tactical
44
2 1 3 4 2 4 5 5 1 4 1 5 4 6 5 1 1 4 2 4 1 2 5 1 5 4 3 2 5 2 2 6 4 2 4 6 4 4 1 2 2 5 3 2
Name is: Bob Last budget is: 1400 Number of properties is:  2
```

And the output is the same with desired output

```
Output
Debugger Console × | Run (Capitaly) ×
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Charliebought prop
Alicebought property
Bobpaid fine false
Bobbought property
Alicepaid finefalse
Alicebought property
Bobpaid fine false
Bobpaid fine false
Charliebought house
Alicepaid finefalse
Charliepaid fine false
Alicepaid finetrue
Bobpaid fine false
Charliepaid fine false
Alicepaid finefalse
Charliepaid fine false
Alice lost the game
Bobbought property
Bobpaid fine true
Charlie lost the game
Name is: Bob Last budget is: 1400 Number of properties is:  2 If it is same: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.169 s
Finished at: 2024-10-09T11:23:46+02:00
------------------------------------------------------------------------
Output
```

The first input is “input5.txt” file

```
20
Lucky 800
Property
Service 2500
Service 1500
Lucky 500
Property
Property
Property
Service 3000
Service 4500
Lucky 1200
Property
Property
Property
Lucky 700
Property
Service 3500
Lucky 1000
Service 2000
Property
10
Alice Careful
Bob Greedy
Charlie Tactical
David Careful
Eva Tactical
Frank Greedy
Grace Tactical
Hannah Careful
Isaac Greedy
Jack Tactical
141
1 1 1 1 1 6 4 5 3 3 2 5 6 4 4 2 5 1 5 3 5 6 5 1 1 4 5 2 1 2 3 2 3 5 5 4 4 6 1 5 1 3 5 4 6 6 3 5 6 4 5 2 5 5 2 1 3 3 3 6 1 1 2 4 1 5 1 2 1 4 6 1 1 6 2 6 3 3 2 5 4 1 3 2 4
Name is: Eva Last budget is: 4500 Number of properties is:  2
```

And the output is the same with desired output

```
Output
Debugger Console ×   Run (Capitaly) ×
Jack lost the game
Bobpaid finefalse
Davidbought property
Gracepaid fine false
Isaac lost the game
Bobpaid finefalse
Charliepaid fine false
Charlie lost the game
Evabought prop
Alicepaid fine false
Bobpaid finefalse
Hannah lost the game
Alice lost the game
Bobbought property
Davidbought property
Evapaid fine false
Bobpaid finefalse
Davidbought property
Gracepaid fine false
Gracepaid fine false
Bobbought property
Grace lost the game
Bobbought house
Evapaid fine false
Davidbought property
Bob lost the game
Evapaid fine false
Evapaid fine false
Evapaid fine false
Evabought prop
David lost the game
Name is: Eva Last budget is: 4500 Number of properties is:  2 If it is same: true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.084 s
Finished at: 2024-10-09T11:26:39+02:00
------------------------------------------------------------------------
```

Blackbox testing

When we give a file that is not present in the directory it triggers the FileNotFoundException and we get error message

```
Output
Debugger Console ×   Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/
Scanning for projects...

-----------------------< com.mycompany:Capitaly >-----------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Error: File not found: input51.txt (No such file or directory)
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.885 s
Finished at: 2024-10-09T15:31:16+02:00
------------------------------------------------------------------------
```

When we give a file that have negative number it triggers the InvalidInputException and we get error message testing1.txt

```
Output
Debugger Console ×   Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/
Scanning for projects...

------------------------< com.mycompany:Capitaly >------------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Error: Field count must be a non-negative integer.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.855 s
Finished at: 2024-10-09T15:35:13+02:00
------------------------------------------------------------------------
```

When we give a file that have not notmal field type it triggers the InvalidInputException("Invalid field type: " and we get error message: testing2.txt

```
Output
Debugger Console ×   Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/
Scanning for projects...

------------------------< com.mycompany:Capitaly >------------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Error: Invalid field type: Propertyskndaj
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.182 s
Finished at: 2024-10-09T15:37:51+02:00
------------------------------------------------------------------------
```

When we give a file that have dice roll which is not in 1,6 range it triggers the InvalidInputException("Dice roll should be in range (1,6)") and we get error message : testing3.txt

```
Output
Debugger Console ×   Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/
Scanning for projects...

------------------------< com.mycompany:Capitaly >------------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Error: Dice roll should be in range (1,6)
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.976 s
Finished at: 2024-10-09T15:45:23+02:00
------------------------------------------------------------------------
```

When we give a file that dont have an value for Lucky or Service field triggers the InvalidInputException("Service/Lucky field requires cost.") and we get error message: testing4.txt

```
Output
Debugger Console ×   Run (Capitaly) ×
cd /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly; JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home "/
Scanning for projects...

-----------------------< com.mycompany:Capitaly >-----------------------
Building Capitaly 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Capitaly ---
skip non existing resourceDirectory /Users/hajiaga/Desktop/ELTE/ProgrammingTech/Capitaly/src/main/resources

--- compiler:3.11.0:compile (default-compile) @ Capitaly ---
Changes detected - recompiling the module! :source
Compiling 12 source files with javac [debug target 21] to target/classes

--- exec:3.1.0:exec (default-cli) @ Capitaly ---
Error: Service field requires cost.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.939 s
Finished at: 2024-10-09T16:14:52+02:00
------------------------------------------------------------------------
```

Ending

-- All the input files are in the folder I uploaded so you can check them one by one but I already did it and put screenshot of its output

-- Thank you for taking time and reading the documentation