Hacettepe University
Computer Science and Engineering Department
Name and Surname : İdil CAN
Identity Number : 21727011
Course : BBM104
Experiment : Assignment 4
Subject : Abstract Classes and Interfaces
Data Due : 27th April 2018
Advisors : Feyza Nur KILIÇASLAN & Gönenç ERCAN
e-mail : idil.can@hacettepe.edu.tr

INDEX OF THE REPORT

Topic ..... **Page No**

1. ***Software Using Documentation*** ***3***
   - ***1.1.Software Usage*** ***3***
   - ***1.2.Provided Possibilities*** ***3***
   - ***1.3.Input Format*** ***3***
2. ***Software Design Notes*** ***3***
   - ***2.1.Description of the program*** ***3***
   - ***2.2.Algorithm*** ***4***
   - ***2.3.UML Class Diagram*** ***5***

1. SOFTWARE USING DOCUMENTATION

1.1. Software Usage

Because of the nature of the assignment and the need of being able to append any other jewel, we need to use abstract classes and interfaces. To add new jewels only thing the programmer must do is adding the search methods to the wildcard's code and creating a class that inherits from baseSearcher.

1.2.Provided Possibilities

This design gives the user a great chance of modularity and changeability. Besides, I tried to avoid all code repetitions. Code is plain, simple, clean. Code explains itself, without comment lines. Although, I used comments where I thought it was necessary (Ex: Wildcard's searching system was too complicated to human eye.).

1.3.Input Format

To play the game, user should give a coordinate as

<x-coordinate><space><y-coordinate>

and to end the game he should enter the letter 'E'. Case is not important. After pressing 'E' program prints out the score and ask for user's name.

2. SOFTWARE DESİGN NOTES

2.1. Description of the program

The program is a simplified version of the game "Bejeweled". Simply it connects three of the same or compatible jewels. Than it tells you your rank and score. Every jewel has different points.

2.2.Algorithm

Program can take inputs from both files and keyboard. For file inputs, it takes inputs and creates a list and executes it while going though that list. This program can take the same command line with linux. I coded it in that way. System is different but output is the same.

After deciding how to take arguments, program figures out the input is an endgame or location. If it's an location it checks its surrounding like it's been written in Assignment4.pdf. It checks second and third jewels according to second jewel's searching format. Since it needs too much afford and it came out a bit late, it was kind of hard code but the fact that I don't need to reuse this algorithm anywhere else in the program I didn't see the need of modularize it.

After finding the match, it changes it's spot on the grid to null and decide if all jewels are in the same horizontal direction. If they are, it uses the function that makes the illusion of falling, while copying the data downwards, three times for the bottom value. If not, it executes that function once for all jewels destroyed.

When the command "E" has been entered, the program prints out the total score and takes user's name. Since name is case sensitive, it only checks if the name exists by ".equals()" function. If character exists, it changes its score. If it does not, program creates a new player and adds it to the leaderboard and prints it to the file. If file cannot be found, it opens a new file and writes into it.

2.3. UML Class Diagram