Hacettepe University
Computer Science and Engineering Department

Name and Surname : **Mustafa Sercan AMAÇ**

Identity Number : **21626915**

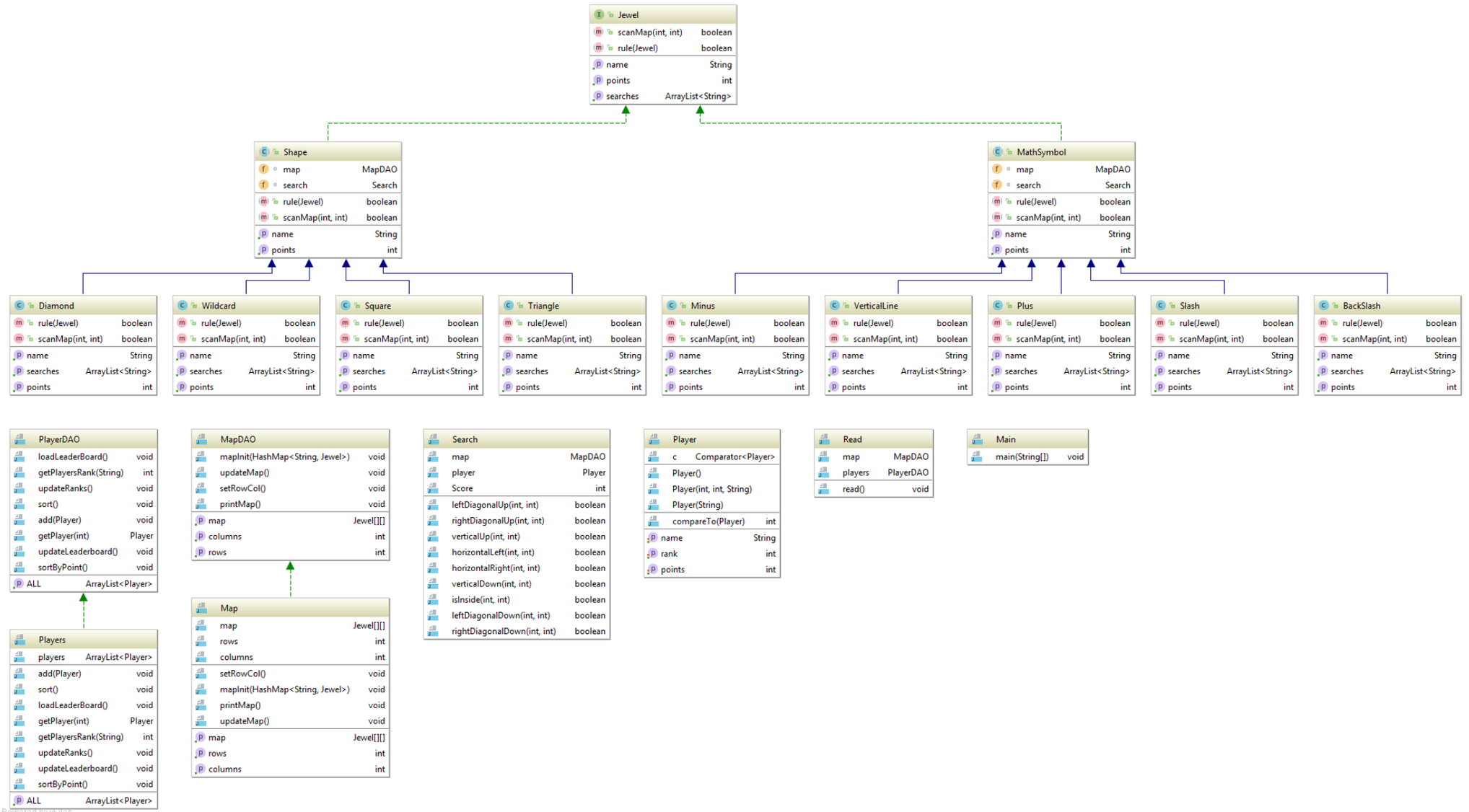
Course : **Bbm104**

Experiment : **Assignment4**

Subject : **Bejewel game**

Data Due : **16.05.2018**

Advisors : **Dr. Gönenç Ercan, Dr. Öner Barut, Dr. Cumhur Yiğit Özcan, Dr. Ali Seydi Keçeli, R.A Feyza Nur Kılıçarslan**



DESIGN

Problem

The problem is design a game (bejewel) using java such that there is no need to change too much code if user wants to add other types of jewels.

Solution

My solution is to use an interface and abstract classes that implements that interface.These abstract classes are the superclasses of the sub jewel types.Such as MathSymbol is an abstract class and Plus class inherits from MathSymbol.So when comparing surroundings with the selected jewel it gives more flexibility and also it makes easy to add new jewel types and subjewels.I created a Search class that has methods to search every direction.That reduced duplicate code and made my code more flexible.

Design For Jewels

I first created an interface called Jewel.I thought every jewel should have a name,point and the directions it can search for.Also a rule method and a scanmap method to scan map with appropriate order and rule according to that jewel.Jewel types are abstract classes that implements from Jewel interface.Every jewel type class should also have a Search object.Every jewel should have their own rule and scanmap method and an arraylist that consists of search directions of that jewel.

Design For Map

I used Data Access Object Pattern for Map.Map class implements MapDAO interface.It has a 2D jewel array.It has methods to create,manipulate and print the map.

Design For Players

I used Data Access Object pattern again.Players implements PlayerDAO.It has an Arraylist that consists of players and methods to manipulate,get and sort them.Its used to keep the leaderboard.Every player has a name ,rank and total points.

Design For Searching

I have a search class that consists of methods that looks for directions that can jewels look for.There is a MapDAO object to access to map and manipulate it.It stores the last match Score and the current player object that plays the game.

How do i add new Jewel type ?

If it is diffrent from MathSymbols and Shapes follow theses steps;
1.Create an abstract Class that implements jewel interface.
2.Put a Search object in to it so subclasses can use it.
3.Create a string array “searches” for every subclass that consists of their search directions.The strings should be same with the method names in the Search Class.
4.For getName method return a string so that its 0st char is the Symbol of it on the map.
5.For getPoints method just put how much does it worth.
6.For rule method determine its rule.Which type of jewels it can match ?Under what condition ?
7.For scanMap method just call search.directionyouwant in if else if blocks in the order you want to search your surroundings.
8.Now its ready to use no need to change any other code in any other class.
If you are just going to add a new jewel as a subclass of existing jewel types its enough to do steps 3-8.

DATA STRUCTURES

JEWEL

getName();
Returns a string that contains a char at 0th index how to check it from Map.

scanMap(int i, int j);
It is scanning jewel's surroundings.If there is a match it returns true if not false.

getPoints();
Return the point of the jewel.

Rule(Jewel shape);
It takes one of the surrounding jewels as a parameter and determines whether it obeys the rule or not.If it obeys returns true if not returns false.

getSearches();

It returns which directions can that jewel look for as a String arraylist.

MAP

setRowCol();
It sets the map Array's rows and columns depend on the rows and columns in gameGrid.txt

mapInit(HashMap<String, Jewel> hashmap);
It takes a hashmap that contains which symbol on the map equals to which type of jewel.It puts objects on the map Array.

updateMap();
It shifts the jewels after a match.

printMap();
It prints the map.

getColumns();
It returns number of columns.

getRows();
It returns number of rows.

SEARCH

It contains search methods for all directions,current player object and last score of the last match.Every method returns true if there is a match returns false if not.

PLAYER

Every player has rank,points and a name.It has 2 compare methods one for names used to get a player's rank by using his/her name using binarySearch.The other is to sort by points to update the ranks after a new player added.And getters setters for fields.

PLAYERS

It has an ArrayList that keeps players in it.

loadLeaderBoard();
It loads a leaderboard if it exists on the directory.

getPlayersRank(String name);

It returns a player's rank using his/her name via binarySearch.

updateRanks();

It updates the rank of the existing players.

Sort();
It sorts by name.

getALL();
It returns the ArrayList containing players.

Add(Player player);
It adds a new player .

getPlayer(int i);
It returns the player on that index.

updateLeaderboard();
It updates the leaderboard.txt

sortByPoint();
It sorts ArrayList by points.

Algorithms

How program starts?

1.Get the filepath of Jewel.java
2.Set it to a string.
3.Get list of files on the same directory.
4.Create an ArrayList called classNames.
5.While iterating on list of files if it's name contains "class" and its not an abstract or interface and if Jewel is assignable from that class add its name in to classNames.
6.While iterating on classNames get the class of the string set it to a variable called clzz store substring(0,1) as key by calling it's getName method and an instance of that class as a value in a hashmap called dict.
7.Create a mapDAO object.
8.Set its rows and columns.
 8.1.Calculate the rows and columns on the gameGrid.txt.
 8.2.Set arrays dimensions to it.
9.Call mapInit with dict.Initiliaze the map using gameGrid.txt.
 9.1.Create a new instance of an object and store it on the map using strings on gameGrid.txt.For example "D" key equals to a Diamond object.Using that create a new object and store it on the map array.
10.Print the Map.
 10.1.for i,j rows and columns if it equals to null print " " if not print substring(0,1) of getName method of that jewel.
11.Create a Read object.

12.Start Reading the input.

How do i read the input?

1.While the command is not E
  1.1.i,j are selected coordinates.
  1.2.if i,j is inside the map and i,j is not empty and if there is a match
      1.2.1.update the map , print the map and print the match score.
 1.3.Else print there is no match
 1.4.If command is equal to E.
      1.4.Print total score ask for players name set it to a string variable.
      1.4.Try to get players rank by using its name if its not in the list already it returns -1 otherwise its over 0 set it to variable b.
      1.4.1.If b equals -1
        1.4.1.1.Set current players name to string.
        1.4.1.2.Add it to players.
        1.4.1.3.Update ranks.
        1.4.1.4.Get its rank.
        1.4.1.5.Print His/her score and current rank and higher/lower scored players. Break the loop
      1.4.2.If b is a positive number
        1.4.2.1.If player had a higher or equal score to his/her current score print its highest rank and tell him/her scored higher or equal to his/her current score. Break the loop.
        1.4.2.2.If player scored higher than his/her highest score than change his/her points and updateranks.Print his/her new rank and higher/lower scored players.Break the loop.

2.Update the leaderboard.txt

How do i search surroundings?

Lets assume that im looking for left horizontal.The logic is same for every direction.
1.If the given coordinate is inside the map and its not equal to null
1.1. if the left box is inside the map and it obeys the rule of the given coordinate and not equal to null

1.1.1.If the left of the left box is inside the map and it obeys the rule of the left box and not null

1.1.1.1.We have a match set Score variable to score of this match increment the players point by that and make matched coordinates null return true.

1.2.Otherwise return false.