



Department of Computer Science & Engineering

BBM104 Introduction to Programming Laboratory

Experiment 4

EMRE HANCI - 21604552

Programming Language : Java

Submission Date : 25.04.2018

Due Date : 16.05.2018

Advisors : Dr. Gönenç Ercan, Dr. Öner Barut, Dr. Cumhuriyet Yiğit  
Özcan, Dr. Ali Seydi Keçeli, R. A. Feyza Nur Kılıçaslan

## -What was the problem?

In this assignment, the expectation from us write Java code of “Bejeweled” type game, using abstract classes and interfaces.

## -Rules

In this program there are two type of jewels and this two type of jewels has nine child type jewels.

Math Jewels;

“ / ” operator: Matches any other mathematical symbol jewel only in right diagonal. Search 3 first, then 7.

“ - ” operator: Matches any other mathematical symbol jewel by searching horizontally (4 and 6).

“ + ” operator: Matches any other mathematical symbol jewel by searching first horizontally (4 and 6) then vertically (2 and 8)

“ \ ” operator: Matches any other mathematical symbol jewel only in left diagonal. Search 1 first, then 9.

“ | ” operator: Matches any other mathematical symbol jewel vertically. Search 2 first, then 8.

Shape Jewels;

“ D ” operator: Matches only D the program searches triple first in left (1 and 9) and then in right diagonal direction (3 and 7).

“ S ” operator: Matches only S the program searches triple first in left and then in right horizontal direction.

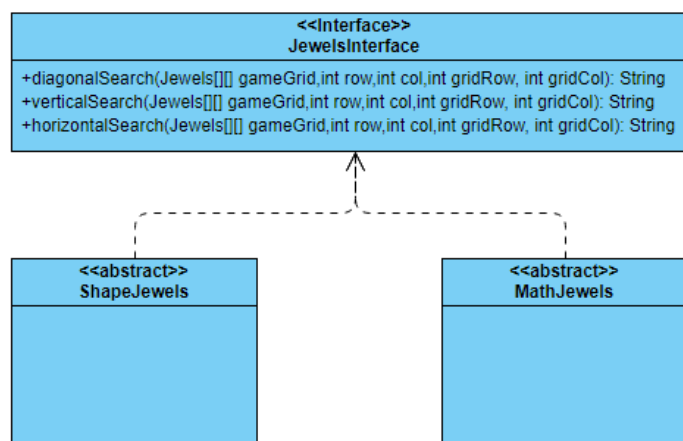
“ T ” operator: Matches only T the program searches the triple first in upward and then in downward vertical direction

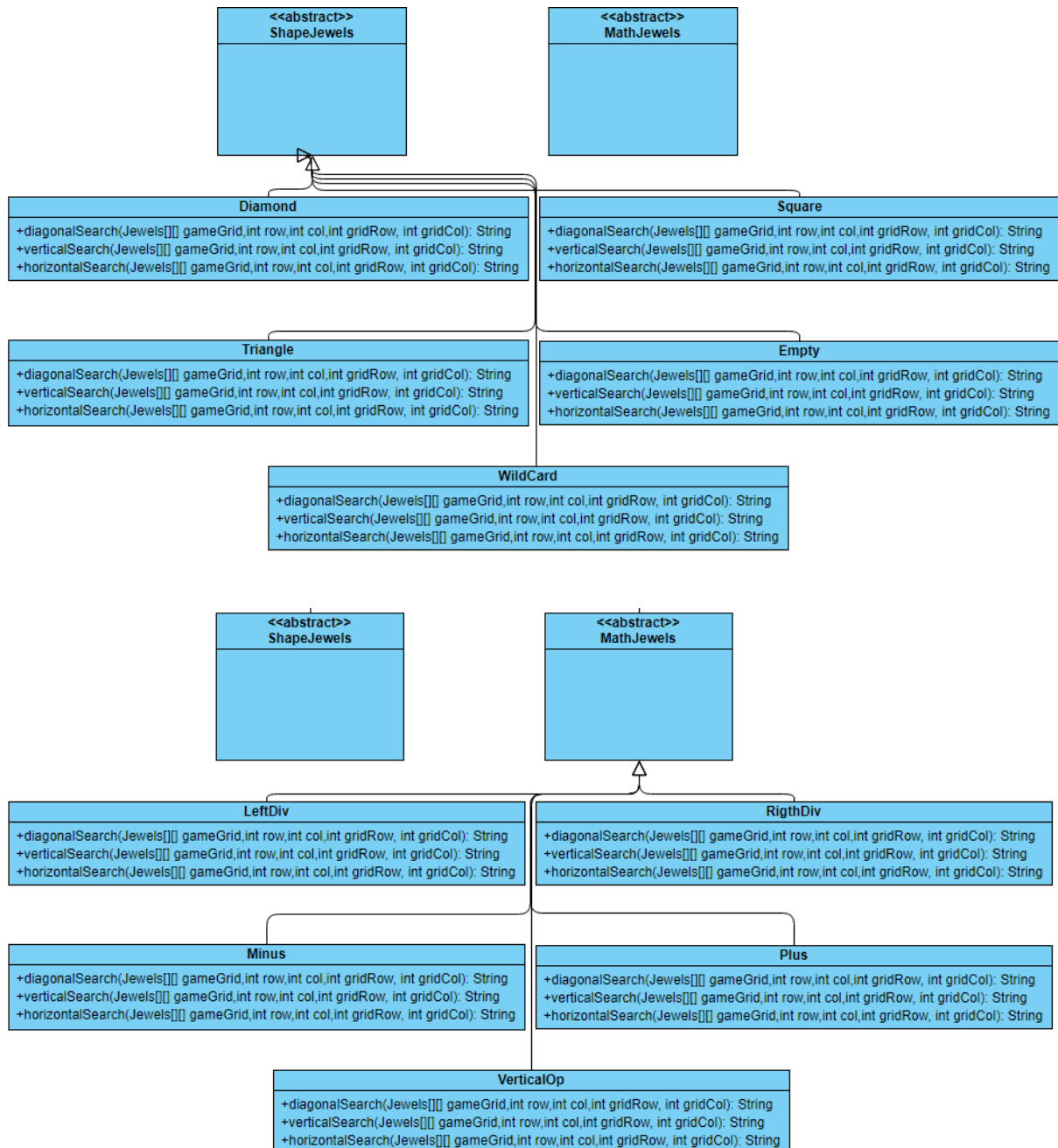
“ W ” operator: Matches with all type of jewels the program starts searching from vertical (2 and 8) followed by horizontal (4 and 6) and finally diagonals left (1 and 9) and right (3 and 7).

## -My solution

For solving this problem, I use abstract classes and and one interface.

My UML Class Diagrams;





I wrote an interface and implement it in two abstract class because of this structre is math jewels are only match with math jewels. The benefit of this structre is provide to using polymorphism. I create this type of jewel and add them grid array which is two dimension according to gameGrid.txt. When my program search and find a triple match jewels this three jewels remove from array and adding new empty type jewels to gameGrid array, then this empty type jewels relocate to top of the array. When my program search and find an triple for any direction it return a string which is has a template "x1;y1;x2;y2;x3;y3;PointOfTriple" the benefit of this strategy is return more than one info not using a string array.

## -The algorithm of my program

The algorithm of my codes;

- Grid type object created.
  - Grid()
  - gameGrid.txt opened.
  - row and collumn counted.
  - gameGrid which is two dimensions array created.
  - new type Jewels created according to txt file and put into array.
  - gameGrid printed to output.
- Program wants an input.
  - If input is not "E"
    - Program check the given coordinate,
      - If coordinate has a objcet which is instance of RightDiv.
        - Program search the given coordinate right diagonal.
          - If there is a triple match
            - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
          - Else
            - Program return null.
      - If coordinate has a objcet which is instance of LeftDiv.
        - Program search the given coordinate left diagonal.
          - If there is a triple match
            - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
          - Else
            - Program return null.
      - If coordinate has a objcet which is instance of VerticalOp.
        - Program search the given coordinate vertical.
          - If there is a triple match
            - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
          - Else
            - Program return null.
      - If coordinate has a objcet which is instance of Minus.
        - Program search the given coordinate horizontal.
          - If there is a triple match
            - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
          - Else
            - Program return null.
      - If coordinate has a objcet which is instance of Plus.
        - Program search the given coordinate horizontal.
          - If there is a triple match
            - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
          - Else
            - Program return null.

- If return value null
  - Program search the given coordinate vertical.
    - If there is a triple match
      - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
- If coordinate has a object which is instance of Diamond.
  - Program search the given coordinate diagonal.
    - If there is a triple match
      - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
    - Else
      - Program return null.
- If coordinate has a object which is instance of Square.
  - Program search the given coordinate horizontal.
    - If there is a triple match
      - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
    - Else
      - Program return null.
- If coordinate has a object which is instance of Triangle.
  - Program search the given coordinate vertical.
    - If there is a triple match
      - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
    - Else
      - Program return null.
- If coordinate has a object which is instance of WildCard.
  - Program search the given coordinate vertical.
    - If there is a triple match
      - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
    - Else
      - Program return null.
  - If return value null
    - Program search the given coordinate horizontal.
      - If there is a triple match
        - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
      - Else
        - Program return null.
    - If return value null
      - Program search the given coordinate diagonal.
        - If there is a triple match
          - Program return a string (this strings template is "x1;y1;x2;y2;x3;y3;PointOfMatch")
        - Else
          - Program return null.
  - If return value null
    - Program prints "There is no match."
  - Else

- Program split the return value
- Program send the coordinates of match to removing function.
  - Removing function change this coordinates object to Empty.
  - Program send those coordinates to shift methods.
    - Shift methods change the coordinates of Empty objects until relocated them to row 0
  - Program print the point of triple.
- Program prints total point.
- Program take a name for save.
- Leaderboard.txt loaded to leaderboard array.
- Leaderboard method add the total point and name to leaderboard array.
- Program prints leaderboard array to leaderboard.txt
- Program prints "Goodbye!"