

CSE 112 3. Midterm

Olcay Taner YILDIZ

I. QUESTION (39 POINTS)

Declare three classes named **Player**, **ComputerPlayer**, and **HumanPlayer** to represent players in a Nim game. In the Nim game, there are N coins on the deck in the beginning of the game. Each player (in turn) takes 1 or 2 coins from deck. Whoever takes the last coin wins. The abstract class **Player** contains:

- (2 pts) A string data field named **name**, an integer data field named **point**.
- (2 pts) One argument constructor **Player(String name)** that sets point to 0.
- (2 pts) Two getter methods for the two fields **name** and **point**.
- (2 pts) Two setter methods for the two fields **name** and **point**.
- (2 pts) Two abstract methods **int play(int N)** and **void display()**.

The class **WrongPlayException** extends Exception and give a message about wrong number of coins played. The class **WrongPlayException** contains:

- (2 pts) One argument constructor **WrongPlayException(int numberOfCoins)**.
- (1 pts) **String toString()** method to print the exception message.

The class **HumanPlayer** extends **Player** by implementing

- (1 pts) One argument constructor that calls its super class's constructor.
- (2 pts) **void display()** that displays the name of the player and its point in the screen.
- (4 pts) **int play(int N)** that takes the number of coins played by the human player as input and returns. It will throw the exception **WrongPlayException** if the player tries to take less than 1 or more than 2 coins from the desk.

The class **ComputerPlayer** extends **Player** by implementing

- (1 pts) One argument constructor that calls its super class's constructor.
- (2 pts) **void display()** that displays "Computer" and its point in the screen.
- (4 pts) **int play(int N)** where the computer takes $(3 - N \% 3)$ coins if $N \% 3 == 1$ or $N \% 3 == 0$, takes a random number of coins (1 or 2) if $N \% 3 == 0$. Here N represents the number of coins that is currently on the deck.

Using the methods and classes declared, implement the main method in a **Deck** class which

- (2 pts) Declares two players Olcay and the computer.
- (2 pts) Takes as input the number of coins on the deck.

- (8 pts) Calls play method of the two players until there are no coins left on the deck. Your code will also display the current situation of the game. The main method must also catch the **WrongPlayException** which is fired by the function play.

II. QUESTION (61 POINTS)

The class **WrongAxisValue** extends Exception and give a message about the axis value. The class **WrongAxisValue** contains:

- (2 pts) One argument constructor **WrongAxisValue(int value, String axisName)**.
- (1 pts) **String toString()** method to print the exception message.

Declare class **Point2D** to represent points in the first quadrant of the two dimensional plane ($x \geq 0, y \geq 0$). The class **Point2D** contains:

- (2 pts) An integer field x and an integer field y to represent the x and y axis value of the point respectively.
- (2 pts) Two argument constructor **Point2D(int x, int y)**.
- (3 pts) Two getter methods for the two fields x and y .
- (6 pts) Two setter methods for the two fields x and y . They throw exception **WrongAxisValue** if x or y is less than 0.
- (6 pts) Implements comparator by creating the function **int compareTo(Object o1, Object o2)**. Two points are compared with each other first according to their x axis values, if they are equal they are compared according to their y axis values.
- (5 pts) The methods **double length()** which calculates the length of the vector with the formula $\sqrt{x^2 + y^2}$.
- (10 pts) A static method **void sort(Point2D[] points)**, that sorts an array of points in increasing order of length.
- (8 pts) A static method **void totalLength(Point2D[] points)**, that calculates the sum of the lengths of the points.
- (8 pts) A static method **void writeToFile(Point2D[] points, String fileName)**, that prints the points to the file with name `fileName`. The first number will represent the number of points, then each line contain the x and y axis values of each point.

```
3
1 2
3 5
5 1
```

- (8 pts) A static method **Point2D[] readFromFile(String fileName)**, that reads points from the file with name `fileName` and creates an array of points. The format of the file is the same as the previous question.