|  |  |
| --- | --- |
| **Practicum Case** |  |
| COMP6175  Object Oriented Programming |
| **Computer Science** | **E201-COMP6175-RV01-09** |
| ***Valid on*** *Even Semester Year 2019/2020* | **Revision 00** |

## Learning Outcomes

* The additional features of OOP
* A program using additional features of OOP

## Topic

* Interface and Abstract Class

## Subtopics

* Abstract Class and Method
* Interface VS Abstract Class

## Soal

*Case*

**BlueJack Run**

**BlueJack Run** is an annual event held by the **BlueJackets**. It’s the most awaited event in the entire year, where each personnel join either **Hare** or **Turtle** group and takes various kinds of sports event. From basketball, baseball, to long-distance run, it’s an event where everybody can have fun. Each year, there will be someone who notes down all the activities to decide which group wins the race. Unfortunately, this year, the person responsible for managing the data is unavailable for the task. You as the substitute are asked to make a simple program to manage the long-distance run data using **Java Programming Language**.

* In the beginning, the program will show the title



* The program will consist of 4 **menus**:

1. Add Runner
2. Join Run
3. View Standing
4. Close Program

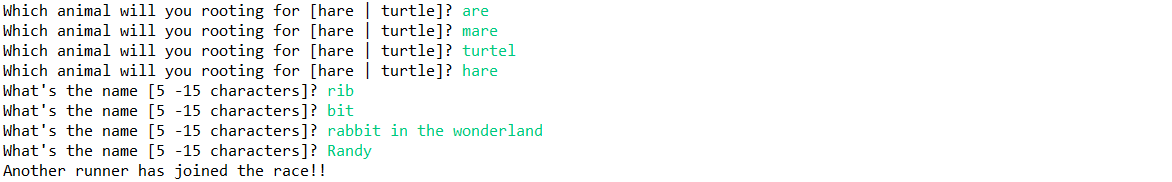
* The program will ask the user to input **choose menu**, which must be inputted **between 1** and **4**



* If user choose menu 1, then:
* The program will ask the user to input **runner detail**, which consists of:
* **Type**, which must be between **hare** and **turtle (case sensitive)**
* **Name**, which **length** must be **between** **5** and **15 characters**
* After all required input has been filled, **add the data to a vector/array** list
* If the user picked **hare**, then:
* Generate the basic **run speed** based on the following formula:

* If the user picked **turtle**, then:
* Generate the basic **run speed** based on the following formula:

* After that, display a message indicating the corresponding personnel has been added to the runner list and return to the main menu



* If the user chooses menu 2, then:
* Display message indicating the table is currently being made

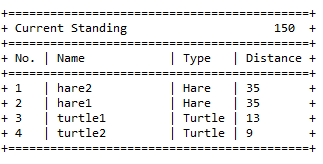


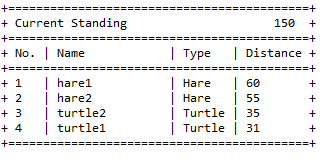
* Display current standing of the competition based on **total distance reached** sort by **descending**  and **calculate new distance reached** for each personnel while there is no winner yet
* **Total Distance** is gotten from random values **between 150**, **250**,and **500**
* **Distance** is gotten from the accumulated distance and the **move** formula
* **Move** formula is based on the following formula:
* If the current type is **hare**, then:

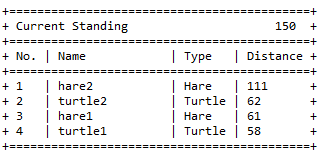
* If the current type is **turtle**, then:

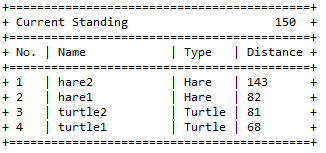
* **special\_move** is gotten from the following formula

* If the **special\_move** **is more than** 7, then give **15 boosts** to the move
* Otherwise **do not give a boost** to the move
* If one of the personnel has **reached the total distance**, then the personnel will be the **winner**

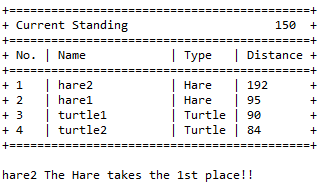








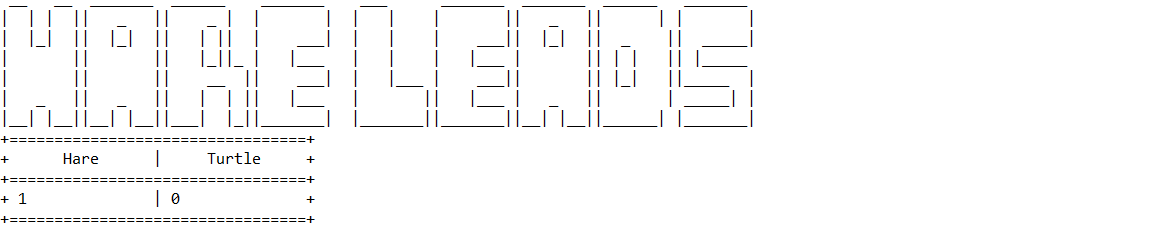
* Display message indicating the winner’s **name** and **increase the winner’s side (**hare or turtle**) by 1**



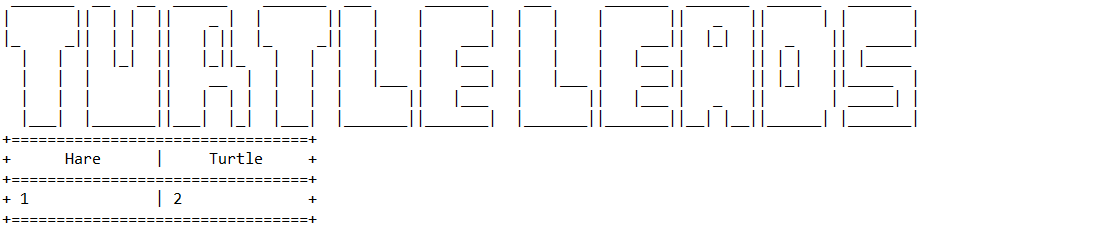
* If the user chooses menu 3, then:
* Display the current standing between **hare** and **turtle**
* If both **hare** and **turtle** have **equal total win**, **display draw** status



* If **the hare** team **has more win than turtle**, **display** **hare win** status



* If the **hare** team **has less win than turtle**, **display** **turtle win** status



* After that, return to the main menu
* If the user choose menu 4, then the program will be **closed**

**Please ask your teaching assistant if there are any related questions.**