Name: **Object Oriented Programming II** **28.06.2024**
Number: **2023-2024 Spring**
Final

90 Minutes

1) Write a Python code to implement a polymorphic player program (50 pts).

Design a class for Player. This class will be superclass. The Player class will have four private attributes (firstName, lastName, age and nation). The class must have appropriate __init__, accessor and mutator methods.

Players Class

- In print_player method, print the data attributes according to the output which is given below. Use accessor methods in print_ player method.

Design a class for LeaguePlayer. This class will be subclass of Player. The LeaguePlayer class will have five private attributes (branch, game_num, win_num, lost_num and player_score). The class must have appropriate __init__, accessor and mutator methods.

LeaguePlayer Class

- In print_player method, call super class’s print_player method. Then, print data attributes of the LeaguePlayer Class. Use accessor methods in print_player method.

- In statistics method, calculate and print the statistics (statistics = player_score / num_game). Use accessor methods in statistics method to get player_score and num_game.

- In points method, calculate and return the points (points = 10*num_game +3* num_win – 2*num_lost). Use accessor methods in point method to get num_game, num_win and num_lost.

Design a class for NationalPlayer. This class will be subclass of LeaguePlayer. The NationalPlayer class will have two private attribute (nationalGame_num and nationalPlayer_score). The class must have appropriate __init__, accessor and mutator methods.

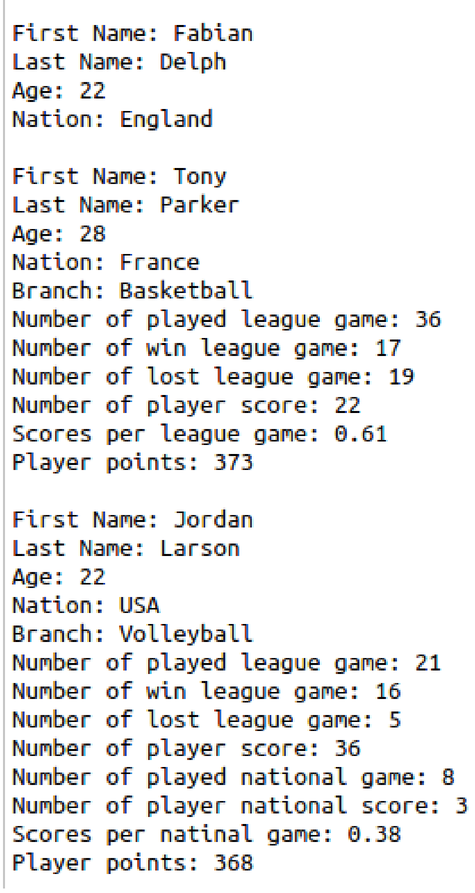
NationalPlayer Class

- In print_player method, call LeaguePlayer class’s print_player method. Then, print data attribute of the NationalPlayer Class. Use accessor methods in print_player method.

- In statistics method, calculate and print the statistics (statistics = nationalPlayer_score / nationalGame_num). Use accessor methods in statistics method to get nationalAll_score and nationalGame_num.

- In points method, calculate and print the points (points = 15* nationalGame_num + (10*num_game +3*num_win – 2*num_lost)). Use accessor methods in points method to get nationalAll_score and nationalGame_num.. Also, use LeaguePlayer class’s points method

The output of the program must be as follows:



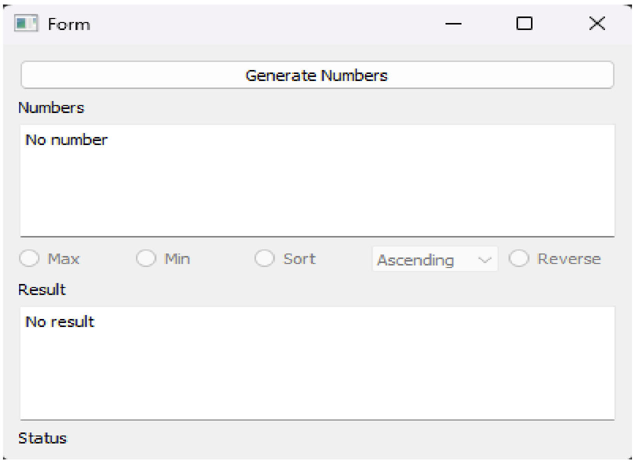
```
First Name: Fabian
Last Name: Delph
Age: 22
Nation: England

First Name: Tony
Last Name: Parker
Age: 28
Nation: France
Branch: Basketball
Number of played league game: 36
Number of win league game: 17
Number of lost league game: 19
Number of player score: 22
Scores per league game: 0.61
Player points: 373

First Name: Jordan
Last Name: Larson
Age: 22
Nation: USA
Branch: Volleyball
Number of played league game: 21
Number of win league game: 16
Number of lost league game: 5
Number of player score: 36
Number of played national game: 8
Number of player national score: 3
Scores per natinal game: 0.38
Player points: 368
```

2) Write a Python program to operate some operations on a list (50 pts).

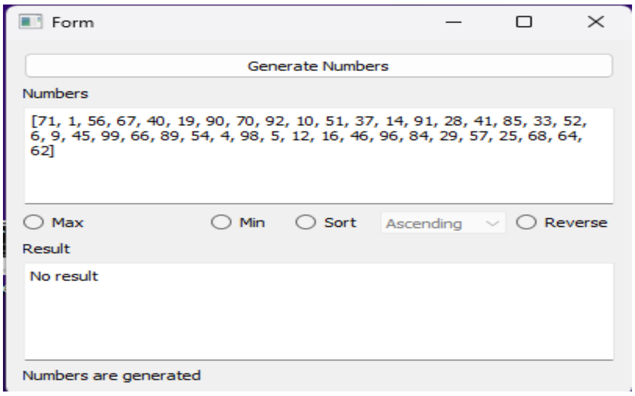
i) Design a Qt application to obtain the following window. In the application, there are 1 Push Button, 3 labels (Numbers, Result, and Status), 2 Text Edits, 4 Toggle Buttons, and 1 Combo Box. Use grid layout (15 pts). **Notice that toggle buttons and combo box are disabled at the beginning.**



No number

No result

ii) When you press "Generate Numbers" button, you must generate 40 numbers randomly between in interval [0,100) and print the numbers into Text Edit. Then, enable toggle buttons and print "Numbers are generated" in Status label (5 pts).

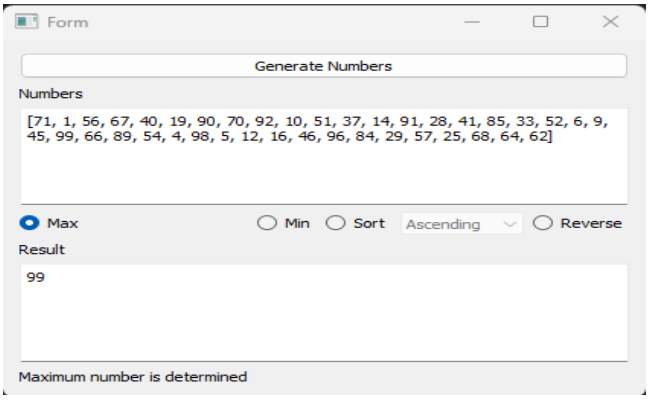


```
[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52,
6, 9, 45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64,
62]
```

No result

Numbers are generated

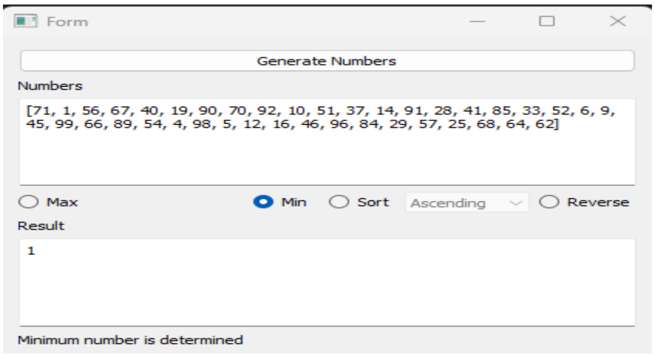
iii) Then, select an operation by using Toggle Buttons. According to your selection, the appropriate operation must be performed, and the result must be displayed in Text Edit. Also, print appropriate texts in Status label (30 pts). (Hint: You can use toggle buttons that is given in https://subscription.packtpub.com/book/application_development/9781788831000/1/ch01lvl1sec13/using-the-radio-button-widget)



```
[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52, 6, 9,
45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64, 62]
```

99

Maximum number is determined



Numbers

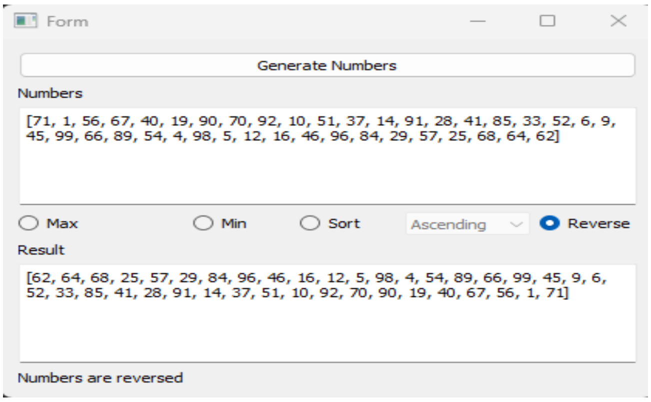
[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52, 6, 9, 45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64, 62]

○ Max ● Min ○ Sort Ascending ○ Reverse

Result

1

Minimum number is determined



Numbers

[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52, 6, 9, 45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64, 62]

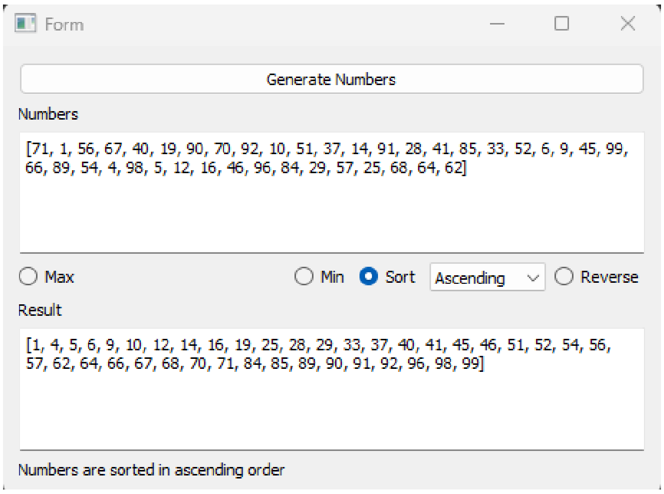
○ Max ○ Min ○ Sort Ascending ● Reverse

Result

[62, 64, 68, 25, 57, 29, 84, 96, 46, 16, 12, 5, 98, 4, 54, 89, 66, 99, 45, 9, 6, 52, 33, 85, 41, 28, 91, 14, 37, 51, 10, 92, 70, 90, 19, 40, 67, 56, 1, 71]

Numbers are reversed

When you select “Sort” toggle button, the combo box must be enabled. The combo box has two items: Ascending and Descending. After you select one of these items, according to item, the numbers must be sorted, and the result must be displayed in Text Edit. Also, print appropriate texts in Status label.



Numbers

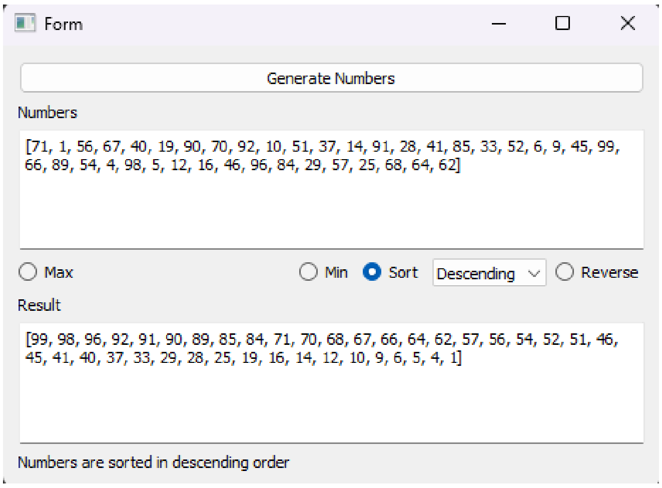
[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52, 6, 9, 45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64, 62]

○ Max ○ Min ● Sort Ascending ○ Reverse

Result

[1, 4, 5, 6, 9, 10, 12, 14, 16, 19, 25, 28, 29, 33, 37, 40, 41, 45, 46, 51, 52, 54, 56, 57, 62, 64, 66, 67, 68, 70, 71, 84, 85, 89, 90, 91, 92, 96, 98, 99]

Numbers are sorted in ascending order



Numbers

[71, 1, 56, 67, 40, 19, 90, 70, 92, 10, 51, 37, 14, 91, 28, 41, 85, 33, 52, 6, 9, 45, 99, 66, 89, 54, 4, 98, 5, 12, 16, 46, 96, 84, 29, 57, 25, 68, 64, 62]

○ Max ○ Min ● Sort Descending ○ Reverse

Result

[99, 98, 96, 92, 91, 90, 89, 85, 84, 71, 70, 68, 67, 66, 64, 62, 57, 56, 54, 52, 51, 46, 45, 41, 40, 37, 33, 29, 28, 25, 19, 16, 14, 12, 10, 9, 6, 5, 4, 1]

Numbers are sorted in descending order

Create a folder named 151220xxxxxx_BxxxxKxxxxx in D. Save your source files in this folder.

Upload source files and text files to ESOGU UZEM.

Good Luck!