Object-oriented programming in Python

Objectives

Development of Python modules to solve complex problems

- Develop Python modules and classes
- Work with standard and compound data types in Python
- Use test-driven development
- Familiarize with special libraries e.g. matplotlib

Deadline

During **lab 6**: present class $\boldsymbol{MyPoint}$

During **lab 7**: present extra functions (some will be given during lab 7)

Beginning of **lab 8**: upload the whole solution

Requirements

1. Implement a solution for the following problem using **classes** and **feature driven development**
2. The solution should offer a console type interface that allows the user to input the data and visualize the output
3. Use only the standard and compound data types available in Python

The solution should ensure:

- Providing at least 10 data examples in the application
- Documentation and testing of each function
- Validation of data – when the user introduces invalid commands or data, a warning should be generated

Problem specification

A math teacher needs a program that helps students perform simple operations with points in two-dimensional space.

1st. Iteration

A point (class $\boldsymbol{MyPoint}$) is identified by the following properties:

- $coord\_x$ given as a number
- $coord\_y$ given as a number
- $color$ given as string (possible values ‘red’, ‘green’, ‘blue’, ‘yellow’ and ‘magenta’)

The following features are to be provided (at the level of class $\boldsymbol{MyPoint}$):

1. Get and set the value of all properties for a point.

2. Provide the string representation of a point.
   For example, for a point with coordinates $coord\_x = 1$, $coord\_y = 2$ and $color = 'red'$, the string format should be "*Point (1, 2) of color red.*"

2nd. Iteration

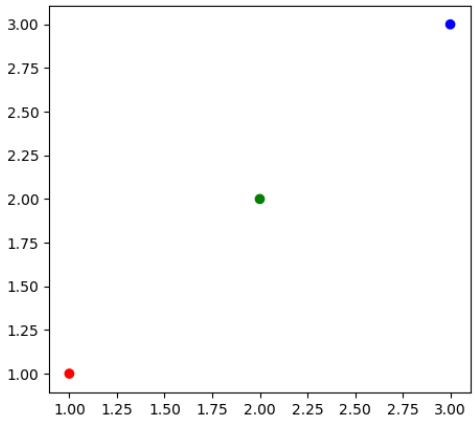
The program manages several points (class ***PointRepository***) and allows operations such as:

1. Add a point to the repository
2. Get all points
3. Get a point at a given index
4. Get all points of a given color
5. Get all points that are inside a given square (up-left corner and length given)
6. Get the minimum distance between two points
7. Update a point at a given index
8. Delete a point by index
9. Delete all points that are inside a given square
10. Plot all points in a chart (using library matplotlib)

Note: Matplotlib (https://matplotlib.org/index.html) is a special library useful for creating quality figures such as plots, bar charts, scatterplots and histograms.

For example, to plot 3 points with coordinates (1,1), (2,2), (3,3) you can use a code like:

```
import matplotlib.pyplot as plt
x = [1, 2, 3]
y = [1, 2, 3]
col = ["red", "green", "blue"]
plt.scatter(x, y, c = col)
plt.show()
```



Submission

Total points: **10**

You need to submit an **archive** (e.g. .zip, .rar, etc) with the source code (**only** your own *.py* files created, without venv or other generated files) to the assignment on **Teams** before the deadline. Please use the following convention to name the archive file:

$sfmie1234\_A3.zip$, where s – first letter of your surname
f – first letter of your first name
mie – stand for mathematics informatics in English
1234 – is your registration number
$A3$ – number of the assignment

If something is not clear, please ask me.

Key

1p Default
1p Work during lab 6
1p Work during lab 7
4p All features correctly implemented
1p At least 10 data examples for each iteration
1p At least 3 assertions for each iteration
1p Documentation