Practical task

Problem definition
Your task is to implement a Python GUI program which collects and displays the environment data from this page: http://agromet.mkgp.gov.si/APP2/sl/Home/Index?id=2&archive=0&graphs=1#esri_map_iframe. Display temperature, wind and pressure for the last 48 hours for all listed cities. Data refresh should be performed on a button click. Preferred data source format is .xml file.

You should create a GitHub repository for the program and set up a Continuous Integration (CI) pipeline using GitHub Actions. The pipeline should include the following steps:

- Prepare a virtual environment (venv) for the program and its dependencies.
- Run the test suite (unit tests)
- Package the program and its dependencies into a distributable format (.whl for example).
- Publish the package to a package repository (test PyPI for example).
- Report pipeline execution status via email
- You should provide a README file that includes instructions for building, testing, and running the program.

Requirements

- The program should be implemented in Python 3.6 and up.
- GUI separation from the data processing (MVC concept).
- The program should use appropriate libraries and APIs for GUI design (PyQt5), downloading and analyzing the URLs.
- The test suite should cover as much of the code as possible, and should be run automatically as part of the CI pipeline. Use pre-commit hook for unit tests execution and static code analysis.
- The program should be packaged in a way that makes it easy to install and use on other systems.
- The CI pipeline should be defined using GitHub Actions.
- Predict situation that data source can change (a separate layer for dependency injection)
- Predict situation that GUI library can change
- Use object oriented concepts (OOP)

Evaluation criteria

- Correctness and completeness of the implementation.
- Clarity and readability of the code, including use of appropriate Python language features and idioms.
- Coverage and thoroughness of the test suite.
- Proper handling of error cases, including edge cases and corner cases.
- Proper use of Git and GitHub, including use of branching, merging, and pull requests if needed.

- Proper setup of the CI pipeline, including use of GitHub Actions and appropriate packaging, distribution, and deployment of the program.
- Meaningfulness and usefulness of the generated GUI.

Helpful links:

- **venv** manipulation: https://realpython.com/python-virtual-environments-a-primer/

- For **unit tests** development a **pytest** library can be used: https://realpython.com/pytest-python-testing/

- Some basics on **object oriented concepts**: https://www.geeksforgeeks.org/python-oops-concepts/

- **MVC** concept and Python: https://www.giacomodebidda.com/posts/mvc-pattern-in-python-introduction-and-basicmodel/

- **PyQt** startup sample https://build-system.fman.io/pyqt5-tutorial

- **GitHub Actions** startup: https://codefresh.io/learn/github-actions/github-actions-tutorial-and-examples/

- **Git and GitHub** usage: https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

- **Dependency injection** with Python: https://python-dependency-injector.ets-labs.org/introduction/di_in_python.html

Contact for any details: robert.sedevcic@rls.si