Pynguin: Automated Unit Test Generation for Python

Paper doi: https://dl.acm.org/doi/10.1145/3510454.3516829

Team members: Abrudan Rebeca Rafaela, Ciama Andreea Elena

**Tool description**

Pynguin is a robust tool designed for automated unit test generation in Python projects. Using Pynguin was a straightforward process, offering significant benefits in terms of ease of installation and usage.

**Installation**

Installing Pynguin was simple, requiring only Python 3.8 or higher to run. It can be installed from the Python terminal using the pip utility tool (-pip install pynguin) , making it easily accessible for users. The only extra requirement is that Rust has to be installed too. Once installed, Pynguin can be executed as a standalone command-line application or inside a Docker container, providing flexibility in deployment options.

**Benefits**

Pynguin simplifies the process of generating unit tests in Python projects by offering a user-friendly command-line interface. Beyond automated test generation, Pynguin boasts a modular design that facilitates easy extension and customization.

**Project**

The tool was used on the a simple code that tests whether a triangle is isoscel, scalene or equilateral.

**Use of application**

****

This is the installation command. (make sure ot have rust installed). Pynguin needs to also set the PYNGUIN\_DANGER\_AWARE to true in order to funciton properly:



This is the command to generate tests: pynguin --project-path C:/Users/rebec/Desktop/PynguinProject2 --output-path C:\Users\rebec\Desktop\PynguinProject2\Tests --module-name triangle. The project path, output path, and the module name must be specified.

Pynguin then generates a few test cases for out module. Here is an example.A computer screen shot of a program

Description automatically generated