Pynguin: Tool Analysis

Team members: Iliescu Andrei, Iancu Aurelian, Gligor Ovidiu

Getting started:

To get started started simply use pip:

```
pip install pynguin
```

This is the class that we are going to test using pynguin:

```python
class Queue:
    def __init__(self, size_max: int) -> None:
        assert size_max > 0
        self.max = size_max
        self.head = 0
        self.tail = 0
        self.size = 0
        self.data = array.array("i", range(size_max))

    def empty(self) -> bool:
        return self.size != 0

    def full(self) -> bool:
        return self.size == self.max

    def enqueue(self, x: int) -> bool:
        if self.size == self.max:
            return False
        self.data[self.tail] = x
        self.size += 1
        self.tail += 1
        if self.tail == self.max:
            self.tail = 0
        return True

    def dequeue(self) -> int | None:
        if self.size == 0:
            return None
        x = self.data[self.head]
        self.size -= 1
        self.head += 1
        if self.head == self.max:
            self.head = 0
        return x
```

After that you should be able to to run a command like this:

```
pynguin \
    --project-path ./docs/source/_static \
    --output-path /tmp/pynguin-results \
    --module-name example \
    -v
python -m penguyn
```

And everything should work but unfortunately I runned into some problems that you might run into as well and here is how I fixed them:

- First thing to keep in mind is that I used vscode for the entire time
- After installing pynguin with pip I tried simply calling it like above but it didn't work
- After some research I found that you should use python 3.10 and I was using python 3.11
- After installing python 3.10 I had to choose this version as the python interpreter in vscode. Then I had to restart vscode.
- Then again I tried simply running pynguin and again no luck so the fix was running pynguin with python as a module like this: `python -m penguyn [other-arguments]`
- I again got an error saying that environment variable PYNGUIN_DANGER_AWARE is not set. So the final fix was to go into the system environment variables table, add a new environment variable with the name PYNGUIN_DANGER_AWARE and the value 1, then restart vscode and run the command one more time and boom!...it finally worked!

```
[18:27:38] INFO     Running tests on mutant  17/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  18/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  19/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  20/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  21/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  22/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  23/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  24/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  25/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  26/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  27/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  28/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  29/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  30/31                                  assertiongenerator.py:295
           INFO     Running tests on mutant  31/31                                  assertiongenerator.py:295
           INFO     Mutant 0 killed by Test(s): 1, 2, 3, 4, 5, 6                    assertiongenerator.py:374
           INFO     Mutant 1 killed by Test(s): 1, 3, 4, 5, 6                       assertiongenerator.py:374
           INFO     Mutant 2 killed by Test(s): 1, 3, 5, 6                          assertiongenerator.py:374
           INFO     Mutant 3 killed by Test(s): 1, 3, 5, 6                          assertiongenerator.py:374
           INFO     Mutant 4 killed by Test(s): 1, 2, 3, 4, 5, 6                    assertiongenerator.py:374
           INFO     Mutant 5 killed by Test(s): 1, 3, 4, 5, 6                       assertiongenerator.py:374
           INFO     Mutant 6 killed by Test(s): 1, 3, 5, 6                          assertiongenerator.py:374
           INFO     Mutant 7 killed by Test(s): 1, 3, 5, 6                          assertiongenerator.py:374
           INFO     Mutant 8 killed by Test(s): 0, 1, 2, 5, 6                       assertiongenerator.py:374
           INFO     Mutant 9 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                 assertiongenerator.py:374
           INFO     Mutant 10 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                assertiongenerator.py:374
           INFO     Mutant 11 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                assertiongenerator.py:374
           INFO     Mutant 12 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                assertiongenerator.py:374
           INFO     Mutant 13 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                assertiongenerator.py:374
           INFO     Mutant 14 killed by Test(s): 1, 4, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 15 killed by Test(s): 1, 2, 3, 4, 5, 6                   assertiongenerator.py:374
           INFO     Mutant 16 killed by Test(s): 1, 3, 4, 5, 6                      assertiongenerator.py:374
           INFO     Mutant 17 killed by Test(s): 1, 5, 6                            assertiongenerator.py:374
           INFO     Mutant 18 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 19 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 20 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 21 killed by Test(s): 1, 5, 6                            assertiongenerator.py:374
           INFO     Mutant 22 killed by Test(s): 1, 5, 6                            assertiongenerator.py:374
           INFO     Mutant 23 killed by Test(s): 0, 1, 2, 3, 4, 5, 6                assertiongenerator.py:374
           INFO     Mutant 24 killed by Test(s): 1, 5, 6                            assertiongenerator.py:374
           INFO     Mutant 25 killed by Test(s): 1, 4, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 26 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 27 killed by Test(s): 1, 2, 3, 4, 5, 6                   assertiongenerator.py:374
           INFO     Mutant 28 killed by Test(s): 1, 3, 4, 5, 6                      assertiongenerator.py:374
           INFO     Mutant 29 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Mutant 30 killed by Test(s): 1, 3, 5, 6                         assertiongenerator.py:374
           INFO     Number of Surviving Mutant(s): 0 (Mutants: )                    assertiongenerator.py:386
           INFO     Calculating resulting FinalBranchCoverage                                generator.py:438
           INFO     Written 7 test cases to D:\IntelliJ Projects\Pynguin\pynguin-result\test_queue_example.py   generator.py:708
           INFO     Writing statistics                                                      statistics.py:369
           INFO     Stop Pynguin Test Generation…                                            generator.py:110
```

And the file that was generated:

```python
import pytest
import queue_example as module_0


def test_case_0():
    bool_0 = True
    queue_0 = module_0.Queue(bool_0)
    assert (
        f"{type(queue_0).__module__}.{type(queue_0).__qualname__}"
        == "queue_example.Queue"
    )
    assert queue_0.max is True
    assert queue_0.head == 0
    assert queue_0.tail == 0
    assert queue_0.size == 0
    assert (
        f"{type(queue_0.data).__module__}.{type(queue_0.data).__qualname__}"
        == "array.array"
    )
    assert len(queue_0.data) == 1
    var_0 = queue_0.dequeue()


def test_case_1():
    bool_0 = False
    with pytest.raises(AssertionError):
        module_0.Queue(bool_0)


def test_case_2():
    int_0 = -2314
    bool_0 = True
    queue_0 = module_0.Queue(bool_0)
    assert (
```