

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №6
дисциплины «Объектно-ориентированное программирование»
Вариант 3

Выполнил:
Говоров Егор Юрьевич
3 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

(подпись)

Проверил:
Воронкин Роман Александрович,
доцент департамента цифровых,
робототехнических систем и
электроники

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Классы данных в Python

Цель: приобретение навыков по работе с классами данных при написании программ с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий: https://github.com/Artorias1469/object-oriented-programming_6.git

Ход работы:

1. Проработка примера:

Задача: Для примера 2 лабораторной работы 9 добавьте возможность работы с классами данных, а также сохранения и чтения данных в формат XML.

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_6\Prog>python Prim.py
>>> add
Фамилия и инициалы? Говоров Егор Юрьевич
Должность? Руководитель
Год поступления? 2004
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Говоров Егор Юрьевич | Руководитель | 2004 |
+-----+-----+-----+-----+
>>> save Prim.xml
>>> load Prim.xml
>>> list
+-----+-----+-----+-----+
| № | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Говоров Егор Юрьевич | Руководитель | 2004 |
+-----+-----+-----+-----+
```

Рис 1. Результат работы примера

2. Выполнение индивидуального задания:

Задача: Выполнить индивидуальное задание лабораторной работы 4.5, используя классы данных, а также загрузку и сохранение данных в формат XML.

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_6\Prog>python Ind_1.py -f -s -o Ind.xml
├── Ind_1.py (5084 bytes)
├── Prim.py (6100 bytes)
└── prim.xml (179 bytes)
Структура каталога сохранена в Ind.xml.

D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_6\Prog>python Ind_1.py -i Ind.xml
Структура каталога, загруженная из Ind.xml:
├── Ind_1.py (5084 bytes)
├── Prim.py (6100 bytes)
└── prim.xml (179 bytes)
```

Рис 2. Результат выполнения индивидуального задания

```
D:\Users\Admin\Рабочий стол\ООП\object-oriented-programming_6\Prog>mypy Ind_1.py
Success: no issues found in 1 source file
```

Рис 3. Проверка типов данных в коде с помощью утилиты mypy

```
D:\Users\Admin\Рабочий стол\BACKAPP\ООП\object-oriented-programming_6\Test>pytest
===== test session starts =====
platform win32 -- Python 3.11.7, pytest-7.4.0, pluggy-1.0.0
rootdir: D:\Users\Admin\Рабочий стол\BACKAPP\ООП\object-oriented-programming_6\Test
plugins: anyio-4.2.0
collected 3 items

test_Ind_1.py ... [100%]

===== 3 passed in 0.16s =====
```

Рис 4. Проверка корректности написания кода с помощью получения ожидаемых результатов на основе использования pytest

Ответы на контрольные вопросы:

1. Как создать класс данных в языке Python?

В Python для создания класса данных используется модуль `dataclasses`, который позволяет автоматически генерировать специальные методы, такие как `__init__()` и `__repr__()`. Чтобы создать класс данных, необходимо использовать декоратор `@dataclass`

2. Какие методы по умолчанию реализует класс данных?

Классы данных автоматически реализуют несколько специальных методов, включая:

- 1) `__init__()`: для инициализации экземпляров класса,
- 2) `__repr__()`: для представления экземпляров класса в виде строки,
- 3) `__eq__()`: для сравнения экземпляров класса на равенство,
- 4) `__hash__()`: если класс данных является неизменяемым, этот метод также будет сгенерирован, Эти методы позволяют избежать написания шаблонного кода и делают работу с классами данных более удобной.

3. Как создать неизменяемый класс данных?

Чтобы создать неизменяемый класс данных, нужно использовать параметр `frozen=True` в декораторе `@dataclass`. Это сделает все атрибуты класса неизменяемыми после их инициализации. Вот пример: `@dataclass(frozen=True)`

Вывод: в ходе выполнения работы были приобретены навыки по работе с классами данных при написании программ с помощью языка программирования Python версии 3.x.