

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Лабораторная работа 6

Классы данных в Python

Выполнил студент группы ИВТ-б-о-20-1

Пентухов С. А. « » _____ 2022г.

Подпись студента _____

Работа защищена « » _____ 2022г.

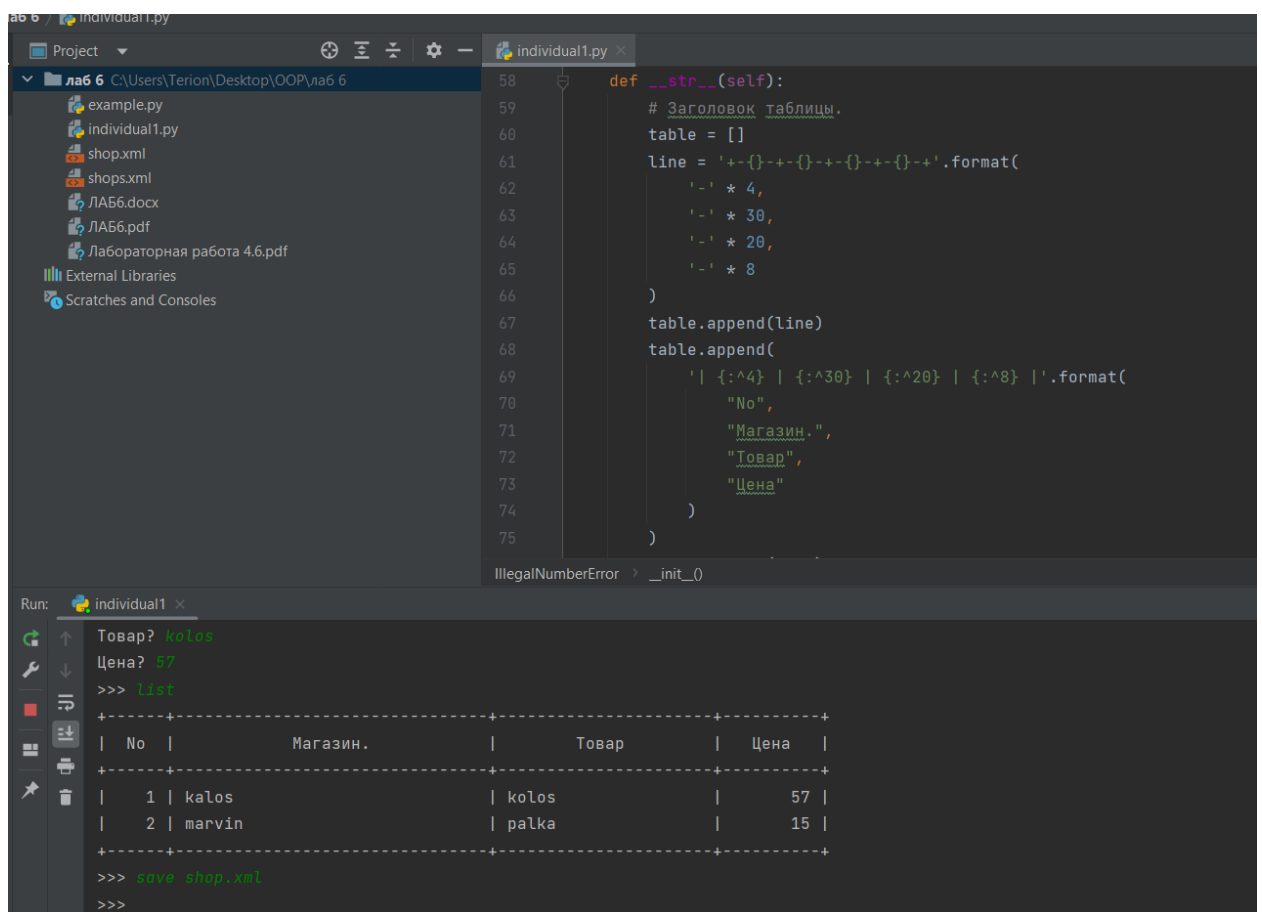
Проверил Воронкин Р.А. _____

(подпись)

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

Индивидуальное задание

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



The screenshot shows a Python IDE with a project named 'лаб 6'. The file explorer on the left lists files: example.py, individual1.py, shop.xml, shops.xml, ЛАБ6.docx, ЛАБ6.pdf, and Лабораторная работа 4.6.pdf. The main editor shows the code for 'individual1.py'.

```
58 def __str__(self):
59     # Заголовок таблицы.
60     table = []
61     line = '+-{}-+-{}-+-{}-+-{}-+'.format(
62         '- ' * 4,
63         '- ' * 30,
64         '- ' * 20,
65         '- ' * 8
66     )
67     table.append(line)
68     table.append(
69         '| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
70             "No",
71             "Магазин.",
72             "Товар",
73             "Цена"
74         )
75     )
```

The terminal window shows the execution of the program. It prompts for 'Товар?' (kolos) and 'Цена?' (57), then displays a table with two rows of data. Finally, it prompts to 'save shop.xml'.

```
Run: individual1
Товар? kolos
Цена? 57
>>> list
+-----+-----+-----+-----+
| No |          Магазин.          | Товар | Цена |
+-----+-----+-----+-----+
| 1 | kalos | kolos | 57 |
| 2 | marvin | palka | 15 |
+-----+-----+-----+-----+
>>> save shop.xml
>>>
```

Рисунок 1 – Полученная XML-таблица

На рисунке 2 изображен созданный xml-файл.




Имя	Дата изменения	Тип	Размер
 individual1.py	22.10.2022 11:24	JetBrains PyCharm	7 КБ
 shop.xml	22.10.2022 11:25	Документ XML	1 КБ
 ЛАБ6.docx	22.10.2022 11:26	Документ Microso...	123 КБ

Рисунок 2 – Файл с данными

Ответы на контрольные вопросы

1. Как осуществляется объявление класса в языке Python? - Классы объявляются с помощью ключевого слова `class` и имени класса.
2. Чем атрибуты класса отличаются от атрибутов экземпляра? - Атрибуты класса определены внутри класса, но вне каких-либо методов. Их значения одинаковы для всех экземпляров этого класса. Так что вы можете рассматривать их как тип значений по умолчанию для всех наших объектов. Что касается переменных экземпляра, они хранят данные, уникальные для каждого объекта класса. В этой теме мы рассмотрим только атрибуты класса, но не волнуйтесь, у вас будет достаточно времени, чтобы узнать больше и об атрибутах экземпляра.
3. Каково назначение методов класса? - Методы определяют функциональность объектов, принадлежащих конкретному классу.
4. Для чего предназначен метод `__init__()` класса? – Для объявления конструктора класса.
5. Каково назначение `self` ? – Указание объекта на самого себя.
6. Как добавить атрибуты в класс? - Атрибуты экземпляра - это как раз те, которые мы определяем в методах, поэтому по определению мы можем создавать новые атрибуты внутри наших пользовательских методов.

Как осуществляется управление доступом к методам и атрибутам в языке Python? - т.д.) - В Python таких возможностей нет, и любой может обратиться к атрибутам и методам вашего класса, если возникнет такая необходимость. Это существенный недостаток этого языка, т.к. нарушается один из ключевых принципов ООП – инкапсуляция. Хорошим тоном считается, что для чтения/изменения какого-то атрибута должны использоваться специальные методы, которые называются `getter/setter`, их можно реализовать, но ничего не мешает изменить атрибут напрямую. При этом есть соглашение, что метод или атрибут, который начинается с нижнего подчеркивания, является скрытым, и снаружи класса трогать его

не нужно (хотя сделать это можно).

7. В Python есть встроенная функция `instance ()`, которая сравнивает значение с указанным типом. Если данное значение и тип соответствуют, он вернет `true`, иначе `false`. Используя `isinstance ()`, вы можете проверить строку, число с плавающей точкой, `int`, список, кортеж, `dict`, `set`, `class` и т. Д.