

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Информатика»
Тема: Парадигмы программирования

Студент гр. 3344

Мурдасов М.К.

Преподаватель

Глазунов С.А.

Санкт-Петербург

2024

Цель работы

Получить базовые навыки работы с ООП на языке Python.

Задание

Базовый класс — печатное издание Edition:

class Edition:

Поля объекта класса Edition:

- название (строка)
- цена (в руб., целое положительное число)
- возрастное ограничение (в годах, целое положительное число)
- стиль(значение может быть одной из строк: c (color), b (black))
- При создании экземпляра класса Edition необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

Книга - Book:

class Book: #Наследуется от класса Edition

Поля объекта класс Book:

- название (строка)
- цена (в руб., целое положительное число)
- возрастное ограничение (в годах, целое положительное число)
- стиль(значение может быть одной из строк: c (color), b (black))
- автор (фамилия, в виде строки)
- твердый переплет (значениями могут быть или True, или False)
- количество страниц (целое положительное число)
- При создании экземпляра класса Book необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

В данном классе необходимо реализовать следующие методы:

Метод `__str__()`:

Преобразование к строке вида: Book: название <название>, цена <цена>, возрастное ограничение <возрастное ограничение>, стиль <стиль>, автор

<автор>, твердый переплет <твердый переплет>, количество страниц <количество страниц>.

Метод `__eq__()`:

Метод возвращает True, если два объекта класса равны и False иначе. Два объекта типа Book равны, если равны их название и автор.

Газета - Newspaper:

class Newspaper: #Наследуется от класса Edition

Поля объекта класс Newspaper:

- название (строка)
- цена (в руб., целое положительное число)
- возрастное ограничение (в годах, целое положительное число)
- стиль(значение может быть одной из строк: c (color), b (black))
- интернет издание (значениями могут быть или True, или False)
- страна (строка)
- периодичность (период выпуска газеты в днях, целое положительное число)
- При создании экземпляра класса Newspaper необходимо убедиться, что переданные в конструктор параметры удовлетворяют требованиям, иначе выбросить исключение ValueError с текстом 'Invalid value'.

В данном классе необходимо реализовать следующие методы:

Метод `__str__()`:

Преобразование к строке вида: Newspaper: название <название>, цена <цена>, возрастное ограничение <возрастное ограничение>, стиль <стиль>, интернет издание <интернет издание>, страна <страна>, периодичность <периодичность>.

Метод `__eq__()`:

Метод возвращает True, если два объекта класса равны и False иначе. Два объекта типа Newspaper равны, если равны их название и страна.

Необходимо определить список list для работы с печатным изданием:

Книги:

class BookList – список книг - наследуется от класса list.

Конструктор:

1. Вызвать конструктор базового класса.
2. Передать в конструктор строку name и присвоить её полю name созданного объекта

Необходимо реализовать следующие методы:

Метод append(p_object): Переопределение метода append() списка. В случае, если p_object - книга, элемент добавляется в список, иначе выбрасывается исключение TypeError с текстом: Invalid type <тип_объекта p_object> (результат вызова функции type)

Метод total_pages(): Метод возвращает сумму всех страниц всех имеющихся книг.

Метод print_count(): Вывести количество книг.

Газеты:

class NewspaperList – список газет - наследуется от класса list.

Конструктор:

1. Вызвать конструктор базового класса.
2. Передать в конструктор строку name и присвоить её полю name созданного объекта

Необходимо реализовать следующие методы:

Метод extend(iterable): Переопределение метода extend() списка. В случае, если элемент iterable - объект класса Newspaper, этот элемент добавляется в список, иначе не добавляется.

Метод `print_age()`: Вывести самое низкое возрастное ограничение среди всех газет.

Метод `print_total_price()`: Посчитать и вывести общую цену всех газет.

В отчете укажите:

1. Изображение иерархии описанных вами классов.
2. Методы, которые вы переопределили (в том числе методы класса `object`).
3. В каких случаях будут использованы методы `__str__()` и `__eq__()`.
4. Будут ли работать переопределенные методы класса `list` для `BookList` и `NewspaperList`? Объясните почему и приведите примеры.

Выполнение работы

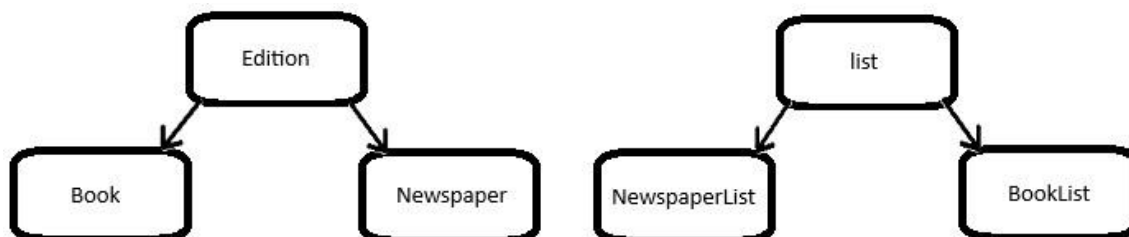


Рисунок 1 – Изображение иерархии классов

1. Методы, унаследованные у *Edition*:

`__init__()` – Принимает параметры издания и проверяет на корректность

`__str__()` – Преобразовывает данные об издании в строку и возвращает ее

`__eq__()` – Сравнивает два объекта одного класса, проверяет издания на идентичность

2. Методы, унаследованные у *list*:

`__init__()` – Принимает параметры издания и проверяет на корректность

`append()` – Добавляет объект в список, если он является книгой, в противном случае выводит ошибку

`total_pages()` – Возвращает сумму всех страниц всех имеющихся книг

`print_count()` – Выводит количество книг

`extend()` – Добавляет в список только элементы класса *Newspaper*

`print_age()` – Выводит минимальное возрастное ограничение среди всех газет

`print_total_price()` – Выводит суммарную стоимость всех газет

3. Метод `__str__()` будет применяться при вызове `str(<издание>)`

Метод `__eq__()` будет применяться при сравнении изданий одного класса.

4. Переопределенные методы класса *list* будут работать для *BookList* и *NewspaperList*, т.к. к ним лишь были добавлены условия, а в остальном функции методов остались прежними.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	<pre>print('----- BOOK-----') book = Book("ахахаха", 999, 1, "b", "Я", True, 1889) book2 = Book("ахахаха", 111, 56, "c", "Я", False, 1945) book3 = Book("хох", 111, 9, "c", "НЕ Я", False, 1945) book_str = Book.__str__(book) book2_str = Book.__str__(book2) print(f'book1 - {book_str}\nbook2 - {book2_str}\nbook == book2: {book == book2}\nbook == book3: {book == book3}')</pre> <pre>print('----- NEWSPAPER----- -----') newspaper = Newspaper("Ниче такая газета", 1, 8, "c", True, "Russia", 7) newspaper2 = Newspaper("Ниче такая газета", 500, 99, "b", True, "Russia", 7) newspaper3 = Newspaper("Че такая газета", 500, 99, "b", True, "He Russia", 7) newspaper_str = Newspaper.__str__(newspap er) newspaper2_str = Newspaper.__str__(newspap er2) print(f'newspaper1 - {newspaper_str}\nnewspaper 2 - {newspaper2_str}\nnewspap er == newspaper2:</pre>	<pre>-----BOOK----- ----- book1 - Book: название ахахаха, цена 999, возрастное ограничение 1, стиль b, автор Я, твердый переплет True, количество страниц 1889. book2 - Book: название ахахаха, цена 111, возрастное ограничение 56, стиль c, автор Я, твердый переплет False, количество страниц 1945. book == book2: True book == book3: False ----- NEWSPAPER----- ----- newspaper1 - Newspaper: название Ниче такая газета, цена 1, возрастное ограничение 8, стиль c, интернет издание True, страна Russia, периодичность 7. newspaper2 - Newspaper: название Ниче такая газета, цена 500, возрастное ограничение 99, стиль b, интернет издание True, страна Russia, периодичность 7. newspaper == newspaper2: True newspaper == newspaper3: False ----- BOOKLIST----- -- total_pages: 5779 print_count(): 3 ----- NEWSPAPERLIST----- -----</pre>	Корректно

<pre> {newspaper == newspaper2}\nnewspaper == newspaper3: {newspaper == newspaper3}') print('----- BOOKLIST----- --') booklist = BookList(Book) booklist.append(book) booklist.append(book2) booklist.append(book3) print(f'total_pages(): {booklist.total_pages()}\npri nt_count():') booklist.print_count() print('----- NEWSPAPERLIST----- -----') news_list = NewspaperList(Newspaper) news_list.extend([newspaper , newspaper2, book, book2, newspaper3]) print('Added elements:', news_list) print('print_age:') news_list.print_age() print('print_total_price:') news_list.print_total_price() </pre>	<pre> Added elements: [<__main__.Newspaper object at 0x00000176174ED690>, <__main__.Newspaper object at 0x00000176174EF790>, <__main__.Newspaper object at 0x00000176174EF7D0>] print_age: 8 print_total_price: 1001 </pre>
---	---

Выводы

Были получены базовые навыки работы с ООП. С использованием ООП была написана программа для работы с информацией о печатных изданиях. Также были изучены наследование классов и переопределение методов.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: Murdasov_Mikhail_lb1.py

```
class Edition:

    def __init__(self, name, price, age_limit, style):

        if not(isinstance(price, int) and price > 0 and
isinstance(age_limit, int) and age_limit > 0 and (style == "c" or style ==
"b") and isinstance(name, str)):
            raise ValueError("Invalid value")

        self.name = name
        self.price = price
        self.age_limit = age_limit
        self.style = style

class Book(Edition):

    def __init__(self, name, price, age_limit, style, author,
hardcover, pages):

        super().__init__(name, price, age_limit, style)
        if not(isinstance(author, str) and isinstance(hardcover,
bool) and isinstance(pages, int) and pages > 0):
            raise ValueError("Invalid value")

        self.author = author
        self.hardcover = hardcover
        self.pages = pages

    def __str__(self):
        return f"Book: название {self.name}, цена {self.price}, возра
ственное ограничение {self.age_limit}, стиль {self.style}, автор {self.author},
твердый переплет {self.hardcover}, количество страниц {self.pages}."

    def __eq__(self, other):
        return self.name == other.name and self.author ==
other.author

class Newspaper(Edition):

    def __init__(self, name, price, age_limit, style, online_edition,
country, frequency):
        super().__init__(name, price, age_limit, style)
        if not(isinstance(online_edition, bool) and
isinstance(country, str) and isinstance(frequency, int) and frequency >
0):
            raise ValueError("Invalid value")

        self.online_edition = online_edition
```

```

        self.country = country
        self.frequency = frequency

    def __str__(self):
        return f"Newspaper: название {self.name}, цена {self.price},
возрастное ограничение {self.age_limit}, стиль {self.style}, интернет издание
{self.online_edition}, страна {self.country}, периодичность
{self.frequency}."

    def __eq__(self, other):
        return self.name == other.name and self.country ==
other.country

class BookList(list):
    def __init__(self, name):
        super().__init__()
        self.name = name

    def append(self, object):
        if isinstance(object, Book):
            super().append(object)
        else:
            raise TypeError(f"Invalid type <тип_объекта
{type(object)}>")

    def total_pages(self):
        return sum([x.pages for x in self])

    def print_count(self):
        print(len(self))

class NewspaperList(list):

    def __init__(self, name):
        super().__init__()
        self.name = name

    def extend(self, iterable):
        super().extend([x for x in iterable if isinstance(x,
Newspaper)])

    def print_age(self):
        print(min([x.age_limit for x in self]))

    def print_total_price(self):
        print(sum([x.price for x in self]))

```