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

ОТЧЕТ

по лабораторной работе №1
по дисциплине «Информационные технологии»
Тема: Парадигмы программирования

Студент гр. 3343 _____ Пивоев Н. М.

Преподаватель _____ Иванов Д.В.

Санкт-Петербург
2024

Цель работы

Ознакомиться с классами в Python и, применив полученные знания на практике, написать программу с их реализацией.

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.

Выполнение работы

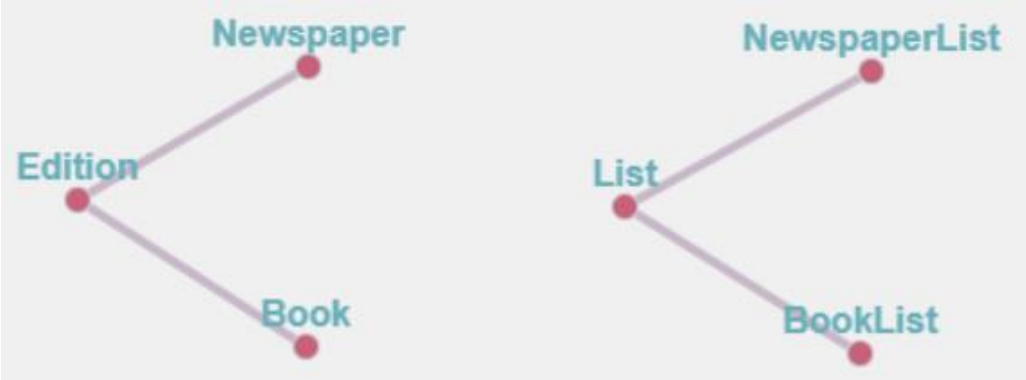

Рисунок 1. Иерархия классов

Методы созданных классов:

1) Методы Edition

__init__ - конструктор класса, принимающий следующие аргументы *name*, *price*, *age_limit* и *style*, если их тип совпадает с определённым в задании.

2) Методы Book и Newspaper

__init__ - конструктор класса, наследующий поля класса Edition и добавляющий новые поля *author*, *hardcover* и *pages* в случае класса Book, и методы *online_edition*, *country* и *frequency*, в случае.

__str__ - метод класса, выводящий информацию о созданном объекте.

__eq__ - метод класса, проводящий сравнение двух объектов по имени и автору для класса Book, по имени и стране для класса Newspaper.

3) Методы BookList

__init__ - конструктор класса, заполняющий поле *name*.

append – метод класса, добавляющий объект класса Book в конец списка. При несовпадении типа объекта вызывает ошибку.

total_pages – метод класса, возвращающий число страниц во всех книгах списка.

print_count – метод класса, возвращающий длину списка.

4) Методы NewspaperList

__init__ - конструктор класса, заполняющий поле *name*.

extend – метод класса, добавляющий несколько элементов в список, в случае совпадения типа объекта с классом Newspaper.

print_age – метод класса, выводящий минимальный возраст среди объектов в списке.

print_total_price – метод класса, выводящий суммарную цену всех объектов списка.

Вопросы:

1) В каких случаях будут использованы методы *__str__()* и *__eq__()*.

Метод *__str__* можно использовать, чтобы получить полную информацию об объекте класса. Метод *__eq__* можно использовать для сравнения, например, во время сортировки.

2) Будет ли работать переопределенные методы класса list для BookList и NewspaperList? Объясните почему и приведите примеры.

Методы будут работать, потому что в них используются функция *super()*, вызывающая методы класса-родителя.

В качестве примера можно рассмотреть метод *extend*, в котором через функцию *super()* осуществляется работа метода *extend*, как у класса-родителя *list*.

Тестирование

Результаты тестирования представлены в табл. 1.

Таблица 1.

№	Входные данные	Выходные данные	Комментарий
1.	a = Book("Book", 10, 16, 'c', "Writer", False, 30) b = Newspaper("Newspaper", 10, 16, 'c', True, "Russia", 10) print(a.__str__()) print(b.__str__())	Book: название Book, цена 10, возрастное ограничение 16, стиль c, автор Writer, твердый переплет False, количество страниц 30. Newspaper: название Newspaper, цена 10, возрастное ограничение 16, стиль c, интернет издание True, страна Russia, периодичность 10.	Вывод соответствует ожиданию.

Выводы

В ходе выполнения лабораторной работы была написана программа с использованием классов, изучены особенности их реализации на языке Python.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.py

```
class Edition:
    def __init__(self, name, price, age_limit, style):
        if type(name) != str or type(price) != int or type(age_limit) != int or price <= 0 or age_limit <= 0 or (style != 'c' and style != 'b'):
            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):
        if type(author) != str or type(hardcover) != bool or type(pages) != int or pages <= 0:
            raise ValueError('Invalid value')
        super().__init__(name, price, age_limit, style)
        self.author = author
        self.hardcover = hardcover
        self.pages = pages
    def __str__(self):
        return f'Book: название {self.name}, цена {self.price}, возрастное 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):
        if type(online_edition) != bool or type(country) != str or type(frequency) != int or frequency <= 0:
            raise ValueError('Invalid value')
```

```
        super().__init__(name, price, age_limit, style)
        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, p_object):
        if isinstance(p_object,Book):
            super().append(p_object)
        else:
            raise TypeError('Invalid type <тип_объекта p_object> (результат вызова функции type)')

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

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


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

```
def extend(self, iterable):
    for i in iterable:
        if isinstance(i, Newspaper):
            super().append(i)

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

def print_total_price(self):
    print(sum([newspaper.price for newspaper in list(self)]))
```