МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И. С. ТУРГЕНЕВА»

Кафедра программной инженерии

ОТЧЕТ

по лабораторной работе № 3
на тему: «Объектно-ориентированное программирование»
по дисциплине: «Программирование на языке Python»

Выполнил: Евдокимов Н.А.
Институт приборостроения, автоматизации и информационных технологий
Направление: 09.03.04 «Программная инженерия»
Группа: 71-ПГ
Проверила: Захарова О.В.

Отметка о зачете:

Дата: «____» _____ 2019 г.

Орёл, 2019

Задание

Вариант 4

В программе разработать класс «Книга» с обязательными атрибутами: ISBN, название, авторы, издательство, год выпуска, количество страниц, стоимость. Вводимую информацию о книгах хранить в списке объектов класса «Книга».

Меню программы: 1) добавление информации в список (разработать конструктор с произвольным количеством параметром); 2) удаление информации о выбранном объекте списка; 3) отображение информации обо всех объектах списка в удобном виде; 4) поиск книг указанного пользователем года выпуска.

Программа должна работать до тех пор (выполнять выбранные пользователем пункты меню программы), пока пользователь не решит из неё выйти.

В лабораторной работе не разрабатывать GUI.

Код

```
import textwrap as tw

from lab3.Book import Book


def print_menu():
    menu = """
        1) Добавить элемент в список
        2) Удалить элемент из списка
        3) Вывести список на экран
        4) Поиск по году
        0) Выйти
    """
    print(tw.dedent(menu))


def add_item(target_list):
    if type(target_list) != list:
        raise ValueError

    name = None
    while name is None:
        name = input('Введите название: ')
        if len(name) <= 0:
            name = None

    publication_year = None
    while publication_year is None:
        try:
            publication_year = int(input('Введите год издания: '))
        except ValueError:
            publication_year = None

    n_pages = None
    while n_pages is None:
        try:
            n_pages = int(input('Введите кол-во страниц: '))
        except ValueError:
            n_pages = None

    isbn = input('Введите ISBN: ')
    if len(isbn) <= 0:
        isbn = None
```

```
    authors = input('Введите авторов через запятую: ')
    authors = authors.lstrip().rstrip().replace(' ', '').split(',')
    authors = [author for author in authors if len(author) > 0]
    if len(authors) <= 0:
        authors = None

    publisher = input('Введите издательство: ')
    if len(publisher) <= 0:
        publisher = None

    try:
        price = float(input('Введите цену: '))
    except ValueError:
        price = None

    book = Book(name, publication_year,
                n_pages, isbn, authors,
                publisher, price)
    target_list.append(book)


def print_book_list(book_list):
    if type(book_list) != list:
        raise ValueError

    if len(book_list) <= 0:
        return

    if type(book_list[0]) != Book:
        raise ValueError

    for index, book in enumerate(book_list):
        book_str = '{})\n\tНазвание: {}\n\tГод издание: {}\n\tКол-во страниц: {}\n'\
            .format(index, book.name, book.publication_year, book.n_pages)

        if book.isbn is not None:
            book_str += '\tISBN: {}\n'.format(book.isbn)

        if book.authors is not None and len(book.authors) > 0:
            book_str += '\tАвторы: {}\n'.format(book.authors)

        if book.publisher is not None:
            book_str += '\tИздательство: {}\n'.format(book.publisher)

        if book.price is not None:
            book_str += '\tЦена: {}'.format(book.price)

        book_str = tw.dedent(book_str)
        print(book_str)


def remove_item(target_list):
    if type(target_list) != list:
        raise ValueError

    print_book_list(target_list)

    index = None
    while index is None:
        try:
            index = int(input('Введите номер элемента: '))
        except ValueError:
            index = None
```

```
    target_list.pop(index)


def search_by_year(book_list):
    if type(book_list) != list or type(book_list[0]) != Book:
        raise ValueError

    menu = """
        Год искомой книги должен быть:
        1) Меньше
        2) Меньше или равен
        3) Равен
        4) Больше или равен
        5) Больше
    """
    menu = tw.dedent(menu)

    condition = None
    while condition is None:
        print(menu)
        try:
            condition = int(input())
        except ValueError:
            condition = None

    year = None
    while year is None:
        print('Введите год')
        try:
            year = int(input())
        except ValueError:
            year = None

        if year < 0:
            year = None

    books = []
    for book in book_list:
        if condition == 1:  # <
            if book.publication_year < year:
                books.append(book)

        elif condition == 2:  # <=
            if book.publication_year <= year:
                books.append(book)

        elif condition == 3:  # ==
            if book.publication_year == year:
                books.append(book)

        elif condition == 4:  # >=
            if book.publication_year >= year:
                books.append(book)

        elif condition == 5:  # >
            if book.publication_year > year:
                books.append(book)

    return books


def run():
```

```
    should_run = True
    book_list = []

    while should_run:
        print_menu()
        try:
            user_choice = int(input())
        except ValueError:
            user_choice = None

        if user_choice is None:
            print("Некоректный ввод")
            continue

        if user_choice == 0:
            should_run = False

        elif user_choice == 1:
            add_item(book_list)

        elif user_choice == 2:
            remove_item(book_list)

        elif user_choice == 3:
            print_book_list(book_list)

        elif user_choice == 4:
            print_book_list(search_by_year(book_list))


if __name__ == '__main__':
    run()
```

Book.py

```
class Book:
    def __init__(
        self, name, publication_year, n_pages,
        isbn=None, authors=None, publisher=None, price=None
    ):
        self.price = price
        self.publisher = publisher
        self.isbn = isbn
        self.n_pages = n_pages
        self.publication_year = publication_year
        self.name = name
        if authors is None:
            authors = []
        self.authors = authors
```