

Homework_Lesson26_Report

Задача: сделать 7 заданий из основного блока на выбор. Усложнённые задания делаются по желанию. Все задания необходимо выполнить, соблюдая принципы ООП, а также обеспечить читаемость кода и покрытие проверками (if checks) там, где это возможно.

1. Создать класс Person с атрибутами name и age. Реализовать метод introduce(), который выводит информацию о человеке.
2. Создать класс Car с атрибутами brand, model, year. Реализовать метод get_info(), который возвращает строку с описанием машины.
3. Наследование: Создать класс Animal с методом make_sound(). От него унаследовать классы Dog и Cat, переопределяя метод make_sound().
4. Инкапсуляция: Создать класс BankAccount с приватным атрибутом _balance и методами deposit(amount), withdraw(amount) и get_balance().
5. Полиморфизм: Написать классы Circle и Rectangle, оба должны иметь метод area(), который возвращает площадь.
6. Классовые атрибуты: Создать класс Employee с атрибутами name, position, а также company_name, который общий для всех сотрудников.
7. Методы класса и статические методы: Добавить в класс Employee метод класса change_company(new_name), который изменяет company_name, и статический метод is_adult(age), который возвращает True, если возраст больше 18.
8. Перегрузка операторов: Реализовать класс Vector с методами __add__, __sub__ для сложения и вычитания векторов.
9. Использование @property: Реализовать класс Temperature, где атрибут celsius автоматически пересчитывает значение в fahrenheit и обратно.
10. Множественное наследование: Создать классы Flyable и Swimmable, затем класс Duck, который наследуется от обоих.
11. Абстрактные классы: Создать абстрактный класс Shape с абстрактным методом area(), затем унаследовать Triangle и Square, реализовав метод area().
12. Singleton: Реализовать класс Logger по паттерну Singleton.
13. Фабричный метод: Реализовать фабрику, которая создает разные типы автомобилей (ElectricCar, GasCar) в зависимости от параметра.

14. Декораторы классов: Реализовать класс Debuggable, который автоматически логирует вызовы методов класса.

15. Контекстные менеджеры: Реализовать класс FileManager, который открывает и закрывает файл через with.

Задание 1

Создать класс Person с атрибутами name и age. Реализовать метод introduce(), который выводит информацию о человеке.

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_1_Person.py
Введите имя: Alex
Введите возраст: 33
Привет, меня зовут Alex, мне 33.
```

Задание 2

Создать класс Car с атрибутами brand, model, year. Реализовать метод get_info(), который возвращает строку с описанием машины.

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_2_Cars.py
2020 BMW X6 127 km Blue
2022 Mercedes GLN 50 km Silver
```

Задание 3

Наследование: Создать класс Animal с методом make_sound(). От него унаследовать классы Dog и Cat, переопределяя метод make_sound().

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_3_Animals.py
Животное издаёт звуки:
Собака: Гав-гав-гав!
Кот: Мяу-мяу-мяу!
```

Задание 4

Инкапсуляция: Создать класс BankAccount с приватным атрибутом _balance и методами deposit(amount), withdraw(amount) и get_balance().

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_4_BankAccount.py
Вы внесли 200. Новый баланс: 1200
Вы сняли 100. Новый баланс: 1100
Текущий баланс: 1100
```

Задание 5

Полиморфизм: Написать классы Circle и Rectangle, оба должны иметь метод area(), который возвращает площадь.

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_5_Area.py
Площадь круга: 201.06
Площадь прямоугольника: 35
```

Задание 15

Контекстные менеджеры: Реализовать класс FileManager, который открывает и закрывает файл через with.

```
avl@ubuntu-s24:~/git/Alexey_Levchenko_D0S24/Alexey_Levchenko_D0S24/Homework_Lesson26_Python4$ python3 Task_15_FileManager.py
Введите имя файла для последующей записи в него текста: file
Введите текст для записи в файл: test tekst
Содержимое файла:
test tekst
```