**ЛАБОРАТОРНАЯ РАБОТА N2**

**ВВЕДЕНИЕ В ООП**

*Цель: Ознакомление с основами объектно-ориентированного программирования (ООП) на языке Python и изучение основных концепций ООП.Теория:*

*Ход работы:*

1. **Изучить теорию**
2. Решить задачи
3. По окончанию работы выложить на github
4. Защитить работы перед преподавателем
5. Выложить отчёт

*Защита происходит следующим образом*: как только работа готова вся, а не 1-2 задачи, зовём преподавателя. На выбор несколько задач будут спрашиваться как вы их решили и как они работают.

**НАПОМИНАНИЕ: Списанная работа, любым способом, либо не будет проверяться, либо будет даваться аналогичная в трое сложнее.**

Работа считается выполненная, если вы сдали и защитили её на паре.

*Теория***:**

Объектно-ориентированное программирование (ООП) - это парадигма программирования, которая строится на концепциях объектов и классов. В Python, как и во многих других языках, ООП является важной составляющей. Основными концепциями ООП являются:

1. **Классы и объекты:** Класс - это шаблон для создания объектов. Объект - это экземпляр класса, обладающий свойствами и методами, определенными в классе.
2. **Атрибуты и методы:** Атрибуты - это переменные, хранящие данные объекта. Методы - это функции, выполняющие операции над объектами.
3. **Инкапсуляция:** Инкапсуляция означает упаковку данных (атрибутов) и методов, работающих с ними, в одном классе. Она позволяет скрыть детали реализации и предоставить интерфейс для работы с объектом.
4. **Наследование:** Наследование позволяет создавать новый класс на основе существующего (родительского) класса. Новый класс наследует атрибуты и методы родительского класса и может их расширять или изменять.
5. **Полиморфизм:** Полиморфизм позволяет объектам разных классов обладать схожим интерфейсом. Это позволяет использовать объекты разных классов с одинаковыми методами без необходимости знать их типы.

*Практика:*

[*https://github.com/ChePchik/book\_python\_oop\_practic*](https://github.com/ChePchik/book_python_oop_practic)выложены 60 заданий, в каждом задании находятся ещё подзадачи. Выполнить всё.

Для выполнения задач создаётся данная структура

* 1 задание (папка)
  + 1.1 задание (файл.py)
  + 1.2 задание

ИТД

После выполнения выложить на github.com

Если нет аккаунта, создать.

Создать репозиторий с именем *python\_oop\_laba2* в который выложить все сделанные задачи.

Выкладываются задачи через консоль.

*Ссылки:*

<https://docs.github.com/en/migrations/importing-source-code/using-the-command-line-to-import-source-code/adding-locally-hosted-code-to-github>

<https://oiplug.com/blog/git/5049/>

*Оценка:*

*Будет 3 оценки, оценивается качество кода и способ выкладывания задания.*

*За несоответствие снимаются баллы.*

*Срок выполнения*: 3 пары с момента начала