Лабораторные работы 6

Присылать на почту daria.zubkova@spbpu.com

Дедлайн: 11.05.2025 до 23:59

Формат (выбирайте один из):

1. архив с задачами в формате *.py/*.ipynb, архив подписывать своей фамилией и именем
2. один файл *.txt, задачи с пометкой #задача 1, #задача 2 и т.д., файл подписывать своей фамилией и именем

Важно: Можно не запрашивать данные через input(), но привести примеры вызова ваших классов и методов.

Задача 1:

Создайте иерархию классов для моделирования транспортных средств:

- Базовый класс `Transport` с методами:
  - `start_engine()` (возвращает «Двигатель запущен»)
  - `stop_engine()` (возвращает «Двигатель остановлен»)
  - `move()` (базовый метод -> «Производится движение»)
- Наследники:
  - `Car` (переопределяет `move()` -> «Едет по дороге»)
  - `Airplane` (переопределяет `move()` -> «Летит по воздуху»)
  - `Boat` (переопределяет `move()` -> «Плывет по воде»)

Требования:

- Использовать наследование
- Каждый дочерний класс должен переопределять метод `move()`

Задача 2:

Создайте иерархию классов животных с разными звуками:

- Базовый класс `Animal` с методом `make_sound()` (возвращает "Неизвестный звук").
- Наследники:
  - `Dog` (переопределяет `make_sound()` -> «Гав!»)
  - `Cat` (переопределяет `make_sound()` -> «Мяу!»)
  - `Bird` (переопределяет `make_sound()` -> «Чирик!»)
  - Добавьте на свое усмотрение еще 2 наследника
- Требования:

- Использовать наследование
- Каждый дочерний класс должен переопределять метод `make_sound()`
- Реализовать функцию `animal_concert()`, которая принимает список животных и вызывает их звуки (функция вне классов)

Задача 3:

Реализуйте класс `User` с защищенными данными профиля:

- Приватные атрибуты:
  - `__username` (логин, минимум 5 символов),
  - `__password` (пароль, минимум 8 символов).
- Публичные методы:
  - `update_password(old_pass, new_pass)` (смена пароля при условии правильного старого).
  - `show_info()` (возвращает логин и пароль, но скрывает пароль: логин: `user123`, пароль: `********`)
- Ошибки:
  - Неверный старый пароль -> `ValueError("Неверный пароль")`
  - Слабый новый пароль (меньше 8 символов) -> `ValueError("Пароль слишком короткий")`
  - Слабый логин (меньше 5 символов) -> `ValueError("Логин слишком короткий")`
- Требования:
  - Инкапсуляция через приватные атрибуты
  - Валидация входных данных в методах
  - Запретить прямой доступ к `__username`, `__password`

Задача 4:

Реализуйте класс `Vector` для работы с векторами:

- Методы:
  - `__add__` (сложение векторов)
  - `__sub__` (вычитание векторов)
  - `__mul__` (умножение на число)
  - `__eq__` (сравнение векторов)

Пример: `Vector(1, 2) + Vector(3, 4) -> Vector(4, 6)`

- Требования:
  - Поддержка операций с числами и другими векторами

Задача 5:

Создайте систему для университета:

- Класс `Student` (свойства: имя, группа)
- Класс `Course` (свойства: название, преподаватель)
- Класс `University` (композиция: содержит список студентов и курсов)
- Методы:
    - `Student.add_student(name, group)` - добавление студентов
    - `Course.add_course (name, teacher)` – добавление курсов
    - `University.sign_up(student, course)` – запись студента на курс
    - `University.get_participants(course)` - вывод списка студентов курса
- Требования:
    - Использовать композицию (университет содержит студентов и курсы)
    - Реализовать взаимодействие между объектами