

Каждый аргумент метода должен быть аннотирован, должно быть известно какой тип данных возвращает метод.

Задание 1:

Создайте класс Shape, который будет представлять абстрактную геометрическую фигуру. Класс должен иметь метод area, который возвращает площадь фигуры, и метод perimeter, который возвращает периметр фигуры. Затем создайте классы Circle, Rectangle и Triangle, которые будут наследовать от класса Shape и переопределять методы area и perimeter согласно формулам для соответствующих фигур. Также добавьте инициализаторы для каждого класса, которые будут принимать параметры, необходимые для определения фигуры (например, радиус для круга, длина и ширина для прямоугольника, три стороны для треугольника).

Задание 2:

Создайте класс Employee, который будет представлять сотрудника компании. Класс должен иметь атрибуты name, position, salary и метод get_info, который возвращает строку с информацией о сотруднике. Затем создайте класс Manager, который будет наследовать от класса Employee и добавлять атрибут subordinates, который будет хранить список подчиненных менеджера. Также переопределите метод get_info, чтобы он включал в строку количество подчиненных. Кроме того, создайте метод add_subordinate, который будет добавлять сотрудника в список подчиненных, и метод remove_subordinate, который будет удалять сотрудника из списка подчиненных.

Задание 3:

Создайте класс Vehicle, который будет представлять транспортное средство. Класс должен иметь атрибуты brand, model, color и метод start, который выводит сообщение "The vehicle is starting". Затем создайте классы Car, Bike и Plane, которые будут наследовать от класса Vehicle и добавлять свои специфические атрибуты и методы. Например, класс Car может иметь атрибут doors, который указывает количество дверей, и метод honk, который выводит сообщение "The car is honking". Класс Bike может иметь атрибут gears, который указывает количество скоростей, и метод pedal, который выводит сообщение "The bike is pedaling". Класс Plane может иметь атрибут wingspan, который указывает размах крыльев, и метод fly, который выводит сообщение "The plane is flying".

Задание 4:

Создайте класс BankAccount, который будет представлять банковский счет. Класс должен иметь атрибуты number, balance и owner, которые будут хранить номер счета, баланс и владельца счета. Также создайте методы deposit, withdraw и transfer, которые будут позволять вносить, снимать и переводить деньги с одного счета на другой. Затем создайте класс Person, который будет представлять человека. Класс должен иметь атрибуты name, age и accounts,

которые будут хранить имя, возраст и список счетов человека. Также создайте метод `get_total_balance`, который будет возвращать общий баланс по всем счетам человека.

Задание 5:

Создайте класс `Book`, который будет представлять книгу. Класс должен иметь атрибуты `title`, `author` и `pages`, которые будут хранить название, автора и количество страниц книги. Также создайте метод `read`, который будет принимать количество страниц, которые были прочитаны, и возвращать процент прочтения книги. Затем создайте класс `Library`, который будет представлять библиотеку. Класс должен иметь атрибут `books`, который будет хранить список объектов класса `Book`, и метод `add_book`, который будет добавлять книгу в список.

Задание 6:

Создайте класс `Product`, который будет представлять товар в интернет-магазине. Класс должен иметь атрибуты `name`, `price` и `description`, которые будут хранить название, цену и описание товара. Затем создайте класс `Cart`, который будет представлять корзину покупателя. Класс должен иметь атрибут `items`, который будет хранить список объектов класса `Product`, и методы `add_item`, `remove_item` и `get_total`, которые будут добавлять, удалять и возвращать общую стоимость товаров в корзине.

Задание 7:

Создайте класс `Student`, который будет представлять студента университета. Класс должен иметь атрибуты `name`, `age` и `courses`, которые будут хранить имя, возраст и список курсов студента. Затем создайте класс `Course`, который будет представлять учебный курс. Класс должен иметь атрибуты `title`, `instructor` и `students`, которые будут хранить название, преподавателя и список студентов, записанных на курс. Также создайте метод `enroll`, который будет принимать объект класса `Student` и добавлять его в список студентов курса, а также добавлять курс в список курсов студента.