### Задание 1: Создание класса и объектов

\*\*Описание задания:\*\* Создайте класс `Car`, который будет представлять автомобиль. В классе должны быть следующие поля:

- Марка автомобиля (`brand`)

- Модель автомобиля (`model`)

- Год выпуска (`year`)

Создайте три объекта этого класса, задайте значения для их полей и выведите на консоль информацию об автомобилях.

\*\*Условие:\*\*

1. Определите конструктор по умолчанию и перегруженный конструктор с параметрами для инициализации полей.

2. Реализуйте метод `Print()`, который будет выводить информацию об автомобиле в формате: `"Марка: {brand}, Модель: {model}, Год выпуска: {year}"`.

---

### Задание 2: Использование модификаторов доступа

\*\*Описание задания:\*\* Создайте класс `BankAccount`, который будет представлять банковский счёт. В классе должны быть следующие поля:

- Номер счёта (`accountNumber`) — доступно только для чтения извне.

- Баланс (`balance`) — доступно только внутри класса и через специальные методы.

Создайте методы:

- `Deposit` — метод, который добавляет средства на счёт.

- `Withdraw` — метод, который снимает средства со счёта, если хватает средств.

\*\*Условие:\*\*

1. Используйте модификаторы доступа `private` и `public`.

2. Реализуйте метод для вывода текущего баланса с помощью метода `PrintBalance()`.

---

### Задание 3: Цепочка конструкторов

\*\*Описание задания:\*\* Создайте класс `Book`, который будет представлять книгу. В классе должно быть три конструктора:

1. Конструктор по умолчанию, который задает название книги как `"Неизвестно"` и год издания как `2024`.

2. Конструктор с параметром для названия книги.

3. Конструктор с параметрами для названия книги и года издания.

Создайте объекты класса с разными конструкторами и выведите их данные.

\*\*Условие:\*\*

1. Используйте цепочку вызова конструкторов через ключевое слово `this`.

2. Реализуйте метод `Print()`, который будет выводить информацию о книге.

---

### Задание 4: Деконструктор и декомпозиция

\*\*Описание задания:\*\* Создайте класс `Student`, который будет представлять студента. В классе должны быть следующие поля:

- Имя студента (`name`)

- Возраст (`age`)

- Курс обучения (`course`)

Определите метод `Deconstruct()`, который позволяет получить имя, возраст и курс студента через декомпозицию объекта.

\*\*Условие:\*\*

1. Создайте объект класса `Student`.

2. С помощью декомпозиции получите и выведите на консоль отдельные значения имени, возраста и курса.

---

### Задание 5: Работа с инициализаторами объектов

\*\*Описание задания:\*\* Создайте класс `House`, который будет представлять дом. В классе должны быть следующие поля:

- Адрес дома (`address`)

- Площадь дома (`area`)

- Год постройки (`yearBuilt`)

Используйте инициализатор объектов для создания нескольких объектов и выведите их информацию.

\*\*Условие:\*\*

1. Определите свойства для каждого поля с модификатором `public`.

2. Создайте объекты с разными значениями полей через инициализатор объекта и выведите информацию.

---

### Задание 6: Работа с контекстом переменных

\*\*Описание задания:\*\* Создайте класс `Rectangle`, который будет представлять прямоугольник. В классе определите:

- Поля ширины и высоты.

- Метод для вычисления площади.

- Метод для вычисления периметра.

Создайте объект класса, вычислите его площадь и периметр, используя переменные на уровне класса, метода и блока кода.

\*\*Условие:\*\*

1. Покажите пример использования переменных с разной областью видимости: на уровне класса, метода и внутри блока кода.

2. Выведите результаты на консоль.

---

### Задание 7: Модификаторы доступа и наследование

\*\*Описание задания:\*\* Создайте класс `Employee` (Сотрудник) с полями:

- Имя (`name`) — публичное поле.

- Должность (`position`) — поле с модификатором доступа `protected`.

- Зарплата (`salary`) — поле с модификатором `private`.

Создайте наследника класса `Manager`, который будет иметь доступ к полям класса `Employee`. В классе `Manager` добавьте метод для вывода информации о должности и зарплате.

\*\*Условие:\*\*

1. Используйте модификаторы доступа `private`, `protected` и `public`.

2. Реализуйте метод для вывода информации о сотруднике в классе `Manager`.

---

### Задание 8: Наследование и полиморфизм

\*\*Описание задания:\*\* Создайте классы `Animal` и его потомков `Cat` и `Dog`. В классе `Animal` определите метод `Speak()` (говорить), который должен быть переопределен в классах-наследниках.

Создайте массив объектов типа `Animal` и вызовите для каждого метода `Speak()`, демонстрируя работу полиморфизма.

\*\*Условие:\*\*

1. Используйте ключевое слово `virtual` для метода `Speak()` в классе `Animal` и ключевое слово `override` в классах-наследниках.

2. Реализуйте полиморфное поведение через массив объектов.