Ключевые слова C# и их назначения

**1. Типы данных**

* **bool**: Логический тип, может принимать значения true или false.
* **byte**: Целочисленный тип, занимающий 1 байт (0 до 255).
* **sbyte**: Целочисленный тип, занимающий 1 байт (-128 до 127).
* **short**: Целочисленный тип, занимающий 2 байта (-32,768 до 32,767).
* **ushort**: Целочисленный тип, занимающий 2 байта (0 до 65,535).
* **int**: Целочисленный тип, занимающий 4 байта (-2,147,483,648 до 2,147,483,647).
* **uint**: Целочисленный тип, занимающий 4 байта (0 до 4,294,967,295).
* **long**: Целочисленный тип, занимающий 8 байт (-9,223,372,036,854,775,808 до 9,223,372,036,854,775,807).
* **ulong**: Целочисленный тип, занимающий 8 байт (0 до 18,446,744,073,709,551,615).
* **float**: Тип с плавающей точкой одинарной точности (4 байта).
* **double**: Тип с плавающей точкой двойной точности (8 байт).
* **decimal**: Тип для работы с денежными значениями (128 бит).
* **char**: Символьный тип, занимающий 2 байта, представляет один символ Unicode.
* **string**: Тип для работы с текстовыми строками.

**2. Условные конструкции**

* **if**: Условная конструкция, выполняет блок кода, если условие истинно.
* **else**: Выполняет блок кода, если предшествующее условие if ложно.
* **else if**: Дополнительное условие для проверки, если предшествующее условие if ложно.
* **switch**: Выбор конструкции, позволяющая выполнять разные блоки кода на основе значения переменной.
* **case**: Определяет блок кода, который будет выполнен в случае совпадения значения с switch.
* **default**: Блок кода, который выполняется, если ни одно из условий в switch не выполнено.

**3. Циклы**

* **for**: Цикл, который выполняет блок кода определенное количество раз.
* **foreach**: Цикл, который перебирает элементы коллекции или массива.
* **while**: Цикл, который выполняет блок кода, пока условие истинно.
* **do**: Цикл, который выполняет блок кода хотя бы один раз, а затем продолжает, пока условие истинно.

**4. Управление потоком выполнения**

* **break**: Прерывает выполнение цикла или switch.
* **continue**: Пропускает текущую итерацию цикла и переходит к следующей.
* **return**: Завершает выполнение метода и возвращает значение (если указано).

**5. Объявление классов и интерфейсов**

* **class**: Объявляет новый класс.
* **interface**: Объявляет новый интерфейс.
* **struct**: Объявляет новую структуру (значимый тип).
* **enum**: Объявляет перечисление.

**6. Модификаторы доступа**

* **public**: Доступно отовсюду.
* **private**: Доступно только внутри того класса, где объявлено.
* **protected**: Доступно в классе и его производных.
* **internal**: Доступно в пределах одной сборки.
* **protected internal**: Доступно внутри сборки и в производных классах.

**7. Асинхронность**

* **async**: Указывает, что метод может содержать асинхронные операции.
* **await**: Ожидает завершения асинхронной операции.

**8. Директивы препроцессора**

* **#define**: Определяет символ для использования в условной компиляции.
* **#undef**: Удаляет определение символа.
* **#if**: Начинает условную компиляцию, если символ определен.
* **#else**: Иначе в условной компиляции.
* **#elif**: Дополнительное условие в условной компиляции.
* **#endif**: Завершает условную компиляцию.

**9. Другие ключевые слова**

* **abstract**: Определяет абстрактный класс или метод.
* **sealed**: Запрещает наследование от класса.
* **static**: Определяет статический класс или метод, который принадлежит классу, а не экземпляру.
* **const**: Определяет константу, значение которой не может изменяться.
* **readonly**: Определяет поле, которое может быть присвоено только в конструкторе или в месте объявления.
* **volatile**: Указывает, что поле может изменяться в многопоточной среде.