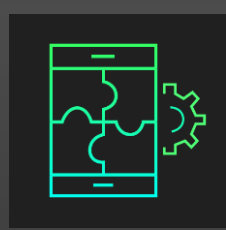


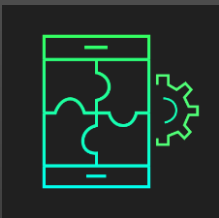


Типы переменных

Определение



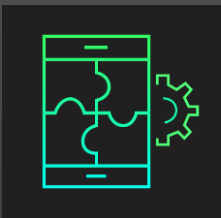
Типы переменных — это категории данных, которые определяют, какие значения переменные могут хранить и какие операции могут быть выполнены над этими значениями. В языках программирования типы переменных определяют не только структуру данных, но и правила обработки и хранения.



Типы переменных

Примитивные типы

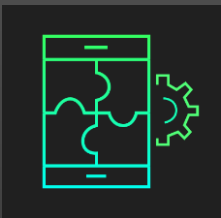
Тип данных	Диапазон значений	Использование	Примеры
int	От -2,147,483,648 до 2,147,483,647	Счетчики и целые числа. Когда необходим подсчет	<code>int userCount = 150; // Количество пользователей</code> <code>int pageCount = 300; // Количество страниц в книге</code> <code>int failedAttempts = 3; // Количество попыток входа</code> <code>int productStock = 50; // Количество товара на складе</code>
long	От -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807	Идентификаторы и большие счетчики. Полезен для больших счетчиков или идентификаторов.	<code>long totalSongs = 1000000000L; // Общее количество песен на платформе</code> <code>long largeTransactionId = 1234567890123L; // Идентификатор транзакции</code> <code>long worldPopulation = 7800000000L; // Население мира</code> <code>long bytesDownloaded = 55000000000L; // Объем загруженных данных</code>
short	От -32,768 до 32,767	Небольшие счетчики. Применяется для небольших счетчиков или значений	<code>short temperature = -5; // Температура в градусах Цельсия</code> <code>short age = 25; // Возраст человека</code> <code>short score = 85; // Оценка в тесте</code> <code>short maxAttempts = 5; // Максимальное количество попыток</code>
byte	От 0 до 255	Небольшие целые значения. Подходит для значений цвета или небольших параметров.	<code>byte redValue = 255; // Значение красного цвета в RGB</code> <code>byte greenValue = 128; // Значение зеленого цвета в RGB</code> <code>byte blueValue = 64; // Значение синего цвета в RGB</code> <code>byte ageInMonths = 24; // Возраст в месяцах</code>



Типы переменных

Числа с плавающей точкой

Тип данных	Диапазон значений	Использование	Примеры
float	$\pm 1.5e-45$ до $\pm 3.4e+38$	Числа с плавающей точкой (одинарная точность). Удобно для не слишком точных расчетов	<code>float speed = 54.2f; // Скорость автомобиля в км/ч</code> <code>float height = 1.75f; // Рост человека в метрах</code> <code>float weight = 68.5f; // Вес человека в килограммах</code> <code>float temperature = 36.6f; // Температура тела</code>
double	$\pm 5.0e-324$ до $\pm 1.7e+308$	Числа с плавающей точкой (двойная точность). Применяется в точных расчетах, таких как финансовые операции.	<code>double bankBalance = 10500.75; // Баланс банковского счета</code> <code>double earthGravity = 9.81; // Ускорение свободного падения на Земле</code> <code>double price = 1999.99; // Цена товара</code> <code>double distance = 150.5; // Расстояние в километрах</code>
decimal	$\pm 1.0 \times 10^{-28}$ до $\pm 7.9 \times 10^{28}$	Высокоточные десятичные вычисления. Применяется в финансовых и бухгалтерских расчетах	<code>decimal taxRate = 0.07m; // Налоговая ставка</code> <code>decimal itemPrice = 19.99m; // Цена товара с учетом десятичных</code> <code>decimal totalAmount = 1500.15m; // Общая сумма покупки</code> <code>decimal interestRate = 0.05m; // Процентная ставка по кредиту</code>



Типы переменных

Символы и строки

Тип данных	Диапазон значений	Использование	Примеры
char	Один символ	Хранение одиночных символов. Применяется для представления статусов или параметров	<code>char grade = 'A';</code> // Оценка студента <code>char currencySymbol = '\$';</code> // Символ валюты <code>char statusFlag = 'Y';</code> // Статус активности <code>char initial = 'J';</code> // Инициалы пользователя
string	Неограниченная длина текста	Хранение строковой информации. Применяется в большинстве случаев, когда нужны строки	<code>string email = "user@example.com";</code> // Электронная почта пользователя <code>string fullName = "Александр Иванов";</code> // Полное имя <code>string address = "Москва, Красная площадь, 1";</code> // Адрес <code>string welcomeMessage = "Добро пожаловать!";</code> // Приветствие

Логический тип

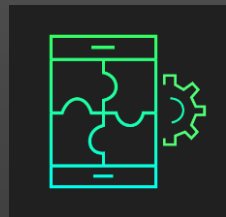
Тип данных	Диапазон значений	Использование	Примеры
bool	true или false	Логические значения. Применяется для проверки условий, состояний и флагов.	<code>bool isLoggedIn = false;</code> // Статус входа в систему <code>bool isActive = true;</code> // Статус активности пользователя <code>bool hasPermission = true;</code> // Допуск к определенным действиям <code>bool isCompleted = false;</code> // Статус завершения задачи



Типы переменных

Ссылочные типы

Тип данных	Диапазон значений	Использование	Примеры
Классы	Пользовательские типы с состоянием и поведением	Моделирование объектов. Используются для создания пользовательских типов с состоянием и поведением	<pre>class Product { public string Name { get; set; } public decimal Price { get; set; } } class Student { public string Name { get; set; } public int Age { get; set; } }</pre>
Массивы	Коллекции значений одного типа	Хранение групп значений (одного типа)	<pre>int[] studentScores = { 90, 85, 78, 92 }; // Оценки студентов string[] bookTitles = { "1984", "Brave New World", "Fahrenheit 451" }; // Названия книг char[] grades = { 'A', 'B', 'C', 'D' }; // Оценки string[] colors = { "Red", "Green", "Blue" }; // Цвета</pre>
Интерфейсы	Определение контрактов	Упрощение взаимодействия между классами. Используются для определения контрактов, которые могут реализовывать классы	<pre>interface IPrintable { void Print(); } interface IMovable { void Move(int x, int y); }</pre>
Делегаты	Ссылки на методы	Обработка событий	<pre>delegate void ButtonClickHandler(); // Делегат для обработки нажатий кнопок delegate int MathOperation(int a, int b); // Делегат для математических операций delegate void StatusChangedHandler(string newStatus); // Делегат для изменения статуса delegate void Notify(); // Делегат для уведомления</pre>



Типы переменных

Ключевое слово `var`

`var` — это ключевое слово в C#, которое используется для неявного объявления переменных. Это означает, что компилятор сам определяет тип переменной на основе присваиваемого значения. Ключевое слово `var` упрощает код, поскольку не требуется явно указывать тип.

Без слова `var`

```
User user = new User();  
BankAccount bankAccount = new BankAccount();
```

Со словом `var`

```
var user = new User();  
var bankAccount = new BankAccount();
```

Плюсы

Сокращение кода, улучшение читаемости, работа с анонимными типами, поддержка LINQ, избежание ошибок при изменении типов