#### Змінні

В загальному випадку будь яка програма має дані, з якими вона працює. Типи є шаблонами, за якими створюють конкретні екземпляри даних. Ці екземпляри у програмі представлені змінними. *Змінна* – це ім'я, яке під час виконання програми представляє конкретні дані у пам'яті. Кожна змінна має тип, який вказують при її оголошенні.

Якщо змінна належить до одного із значущих типів, вона безпосередньо представляє дані. Змінна посилального типу представляє лише посилання (вказівник) на дані. Ця різниця спричиняє певну відмінність у роботі зі змінними, які належать до різних категорій типів даних.

Існують такі види змінних у C#:

► Локальні змінні. Вони містять тимчасові дані, доступні у певній ділянці програми. Такі змінні не є елементами типу.

► Поля. Вони містять дані, пов'язані з типом або з екземпляром типу, і є елементами типу.

► Параметри. Це тимчасові змінні, які використовують для передачі даних від одного методу до іншого. Такі змінні не є елементами типу.

Перед тим як використовувати змінну, її слід оголосити. При оголошенні змінної їй дають назву і пов'язують з певним типом. Виконання рядка програми, у якому оголошено змінну, приводить до виділення для неї пам'яті. Найпростіше оголошення змінної потребує задання її назви і типу:

Тип

Назва

int VariableName;

Наприклад, у приведеному фрагменті коду оголошено змінні цілого типу (тип int) Population та Count, дійсну змінну Summ (тип double), і змінну town типу Town (цей тип не належить до переліку типів C# чи .NET, тому має бути описаний до оголошення відповідної змінної):

int Population; int Count; double Summ; Town town;

Припустимо, що Town є класом – посилальним типом. Тоді виконання четвертого рядка цього коду приведе до виділення у пам'яті місця для посилання, а не для даних об'єкта. Посилання поки-що нікуди не вказує, але в подальшому вказуватиме на дані об'єкта (проте для них також потрібно буде виділити пам'ять). У той же час виконання перших трьох рядків спричинить до виділення у пам'яті місць, куди безпосередньо записуватимуться значення змінних Population, Count та Summ (хоча на даному етапі значення змінних ще не задані).

При оголошенні змінної можна відразу присвоїти їй значення. Присвоєння змінній початкового значення називають її *ініціалізацією*. Для ініціалізації змінної під час оголошення дописують знак "=", після якого вказують потрібне значення змінної:

Ініціалізація

int Percent = 100;

Виконання цієї інструкції приведе до виділення ділянки пам'яті для змінної Percent цілого типу та запису у неї значення 100.

Локальна змінна не може бути використана у програмі, доки їй не буде присвоєно якесь значення. Спроба використання змінної, якій не було присвоєно значення, приведе до помилки компіляції "Use of unassigned local variable…'".

Змінні деяких типів при оголошенні ініціалізуються автоматично. До них належать поля класів, структур та елементи масивів. Локальні змінні та параметри методів автоматично не ініціалізуються. Автоматична ініціалізація полягає у тому, що відразу після створення змінної в неї записується значення за замовчуванням. Всі числові типи ініціалізуються значенням 0, тип char – символом з кодом U+0000, тип bool значенням false. Непрості типи ініціалізуються значенням null (його розглянемо далі).

В одній інструкції можна оголошувати кілька змінних одного і того ж типу, і відразу їх вибірково ініціалізувати. При цьому опис окремих змінних розділяють комами:

int var1 = 10, var2, var3 = 30; double var4, var5 = 55.55;

Тут оголошено три змінних цілого типу (з них перша і остання ініціалізовані), і дві змінних дійсного типу (остання з них ініціалізована).

Оголошувати в одній інструкції змінні різних типів не можна.

Один тип

Інший тип

int Count, float Summ; // Помилка! Так не можна

Назва змінної безпосередньо представляє значення, яке у ній зберігається. Щоб використати значення змінної у програмі, використовують її назву. Наприклад, виконання наступного коду приведе до виводу на екран числа 10 – значення змінної var1:

int var1 = 10; Console.WriteLine("var1 = {0}", var1);