#### Статична і динамічна типизація

Нормальним для C# (і для багатьох інших мов програмування) є випадок, коли при оголошенні змінної вказують її тип. Він дозволяє компілятору розрахувати необхідний для збереження даних обсяг пам'яті і спланувати дії по її виділенню (у стеку чи в купі). Тип змінної не можна міняти під час виконання програми. Такий підхід називають *статичною типизацією*.

Але не всі мови підтримують статичну типизацію. Багато скриптових мов, (зокрема, IronPython та IronRuby) підтримують динамічну типизацію. При цьому тип змінної визначається тільки під час виконання. Для підтримки взаємодії програми на C# з бібліотеками, написаними на мовах з динамічною типизацією, використовують спеціальний тип даних dynamic, який забезпечує роботу зі змінною динамічного типу.

Під час компіляції компілятор не перевіряє тип змінної dynamic. Він аналізує всю інформацію про використання такої змінної і розміщує її разом зі змінною. Під час виконання ця інформація порівнюється з умовами певного типу, до якого ця змінна приводиться. Якщо ці умови не виконуються, генерується виняткова ситуація (помилка).

Ось приклад використання змінної динамічного типу:

dynamic var1 = 10; var1 = var1 \* 2.2; Console.WriteLine("var1 = {0}", var1);

#### Значення null та типи, які його підтримують

При роботі з інформацією, яка представляє реальні дані, часто виникає необхідність показати, що змінна не містить коректних даних. Особливо часто така потреба виникає під час роботи з базами даних. Для посилальних типів з цією метою використовують спеціальне значення null. Це значення не тотожне значенню 0 (нуль). Воно означає відсутність даних.

Типам значень не можна присвоювати значення null. Але C# підтримує модифікатор "?", який використовують для оголошення змінної, що може приймати значення null.

Цей модифікатор допустимий при оголошенні змінних всіх простих попередньо визначених типів. Змінна такого типу з модифікатором "?" може приймати будь-яке значення з діапазону базового типу, і крім цього – значення null.

Наступний код демонструє використання змінної цілого типу, яка може містити значення null:

Модифікатор "?"

int? i = null; i = 10; Console.WriteLine("i = {0}", i);

При оголошенні змінної з модифікатором типу "?" насправді створюється змінна зовсім іншого типу – екземпляр структури System.Nullable<>, яка має свої властивості та методи.

#### Оголошення using

Цей спосіб оголошення змінних було додано починаючи з C# 8.0. Якщо при оголошенні змінної на початку додати ключове слово using, то компілятор вивільнить ресурси, зайняті змінною, при виході з програмного блоку, де її оголошено. Це зручно використовувати, наприклад, при роботі з файлами, які потрібно закривати по закінченню сеансу роботи з ними: using var file = new System.IO.StreamWriter("WriteLines2.txt");