

Конспект

У С# ідентифікатор - це ім'я, що привласнюється методу, змінній або будь-якому іншому, визначеному користувачем, елементу програми. Ідентифікатори можуть складатися з одного або декількох символів. Імена змінних можуть починатися з будь-якої літери алфавіту або знаку підкреслення.

Ключові слова - це визначені ідентифікатори, які мають спеціальне значення для компілятора. Їх використання строго регламентовано.

Операція - конструкція в С#, подібна за записом до математичної, тобто спеціальний спосіб запису деяких дій. Операції поділяються на:

унарні - один аргумент (заперечення, унарний мінус і т. д.);

бінарні - два аргументи (додавання, віднімання, множення і т. д.);

тернарні - три аргументи

Літерали дозволяють задати в програмі значення для числових, символьних і рядкових виразів, а також null-літералів. В С# визначені наступні види літералів:

- цілочисельний (integer);

- дробовий (floating-point);

- булевий (bool);

- символьний (character);

- строковий (string);

- null-літерал (null-literal).

Роздільники - спеціальні символи, що використовуються в конструкціях мови: () [] { } ; , .

Коментарі - пояснення до тексту програми, що знаходяться безпосередньо усередині коду. Коментарі - частина тексту програми, що не впливає на її семантику.

У С# можна виконувати наступні **види перетворень**:

- неявні перетворення;

- явні перетворення (приведення);

- задані користувачем перетворення;

- перетворення за допомогою допоміжних класів.

Неявні перетворення: ніякій спеціальний синтаксис не потрібен, оскільки перетворення є типізованими і дані не будуть втрачені. Приклади включають перетворення з менших у великі цілочисельні типи і перетворення з похідних класів в базові класи.

Явні перетворення вимагають наявності оператора приведення. Приведення потрібно, якщо в ході перетворення дані можуть бути втрачені або перетворення може завершитися помилкою. Приклад - числове перетворення в тип з меншою точністю (меншим діапазоном).

Побітові операції - виконуються спеціальними методами, які можна визначити для включення явних і неявних перетворень між призначеними для користувача типами, що не мають відношення до "базовий клас-похідний клас".

Мій репозиторій у GitHub - <https://github.com/Lamer3011/univer>

Папка з лабораторною - лаба2