МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ

Кафедра інформаційних технологій факультету № 4

ТЕКСТ ЛЕКЦІЇ

з навчальної дисципліни «Алгоритмізація та програмування»
обов'язкових компонент
освітньої програми першого рівня вищої освіти
125 «Кібербезпека»
(«Поліцейська діяльність у кіберсфері», «Безпека інформаційних та комунікаційних систем»)

за темою – ««Вступ у C#. Типи даних»

Харків 2019

ЗАТВЕРДЖЕНО
Науково-методичною радою
Харківського національного
університету внутрішніх справ
Протокол від _*24.01.19*_ № _*1*____

СХВАЛЕНО
Вченою радою факультету № 4
Протокол від _*16.01.19*_ № _*1*____

ПОГОДЖЕНО
Секцією Науково-методичної ради
ХНУВС з технічних дисциплін

Протокол від _*17.01.19*_ № _*1*____

Розглянуто на засіданні кафедри інформаційних технологій *(протокол від 15.01.19 № 1)*

Розробники:
1. *Зав. кафедри,к.т.н., доцент Струков В.М.*

Рецензенти:
1. *д.т.н., професор Зацеркляний М.М.*

План лекції

1. Вступ у C#.
2. Елементи мови, типи даних, константи, змінні.

Література:

Основна:

1. Л.В. Соловей, Н.Н. Мирошниченко, Н.Г. Пономарёва. Программирование на языке C#. – Х. : НТУ «ХПИ», 2016. – 356 с. – На рус. яз.
2. Конспект лекцій.

Додаткова:

3. Э. Троелсен. Язык программирования C# 5.0 и платформа .NET 4.5. , 6-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильяме", 2013. — 1312 с. : ил. — Парал. тит. англ.

Текст лекції

1. Вступ у C#.

C# є спадкоємцем безпосередньо двох найбільш вдалих в області програмування мов: C і C ++. Від мови C він успадкував синтаксис, більшість ключових слів і операторів, а від C ++ - вдосконалену об'єктну модель. Крім того, C # тісно пов'язаний з Java - іншою не менш вдалою мовою.

2. Елементи мови, типи даних, константи, змінні.

2.1. Елементи мови C#.

Множина символів мови C # включає великі та малі букви латинського алфавіту, цифри і знаки. Букви і цифри використовуються для формування ідентифікаторів, констант, ключових слів. ***У C # великі і маленькі букви відрізняються, як в іменах змінних, так і при написанні службових слів***. Ця особливість часто є джерелом помилок при написанні програми.

Лексема - це мінімальна одиниця мови, що має самостійний сенс. Існують наступні види лексем: імена (ідентфікатори), ключові слова, знаки операцій, роздільники, константи.

Сукупність ключових слів становить словник мови. Всі зарезервовані

ключові слова містять тільки рядкові літери (символи нижнього регістра) і написані англійською мовою. Нелатинський алфавіт допускається тільки в коментарях, всередині символьних рядків між подвійними лапками і для символьних змінних між одинарними лапками.

Прогалина використовується не тільки як роздільник лексем мови в тексті програми, а й як символьний знак. Компілятор не реагує на додаткові, «зайві», прогалини в тексті програми, тому їх можна вводити для надання тексту наочності.

2.2. Типи даних.

Як і в багатьох мовах програмування, в C # є своя система типів даних, яка використовується для створення констант і змінних. Тип даних визначає:

1) внутрішнє представлення даних,
2) множину значень, які може приймати об'єкт,
3) а також допустимі дії, які можна застосовувати над об'єктом.

У мові C# всі типи даних згруповані у дві групи: прості і структуровані. Кожний із структурованих типів є окремою темою для розгляду і вони будуть розглянуті у подальшому.

Прості типи подані у наступній таблиці.

Таблиця 1. Прості типи даних.

Позначення в C#	Системний тип	Діапазон	Опис
bool	System.Boolean	true или false	істина або брехня
sbyte	System.SByte	-128 ÷ 127	8-бітне число із знаком
byte	System.Byte	0 ÷ 255	8-бітне число без знака
short	System.Intl6	-32768 ÷ 32767	16-бітне число із знаком
ushort	System.UIntl6	0 ÷ 65535	16-бітне число без знака
int	System.Int32	-2147483648 ÷ 2147483647	32-битне число із знаком

uint	System.UInt32	0 ÷ 4294967295	32-битне число без знака
long	System.Int64	-9223372036854775808 ÷ 9223372036854775807	64-битне число із знаком
ulong	System.UInt64	0 ÷ 18446744073709551615	64-битне число без знака
char	System.Char		Поодинокий 16-бітный символ Unicode
float	System.Single	±1.5E–45 ÷ ±3.4E+38	32-бітне число з плаваючою комою
double	System.Double	±5,0e–324 ÷ ±1.7e+308	64-бітне число з плаваючою комою
decimal	System.Decimal	від 0 до ±79 228 162 514 264 337 593 543 950 335 – число без коми; від 0 до ±7,922816251426433759354395033 5 – число з комою	16 байт, з яких 28 розрядів після коми

2.3. Константи.

Константами називаються дані, значення яких в ході виконання програми не змінюються.

Тільки вбудовані типи C # (за винятком Object) можуть бути оголошені як const. Визначені користувачем типи, включаючи класи, структури і масиви, не можуть бути const.

Константи в C# можуть бути неіменовані та іменовані.

Неіменовані константи визначаються своєю появою в тексті програми, наприклад, в операторі присвоювання x=1; число 1 є константою цілого типу. Таким чином, в даному випадку поява одиниці в операторі присвоювання повністю визначила (тобто зафіксувала конкретне значення і тип цього значення) неіменовану константи цілого типу. Тип константи визначається формою її запису.

Іменована константа повністю визначається в блоці const - тут задається її значення, тип (неявно - по формі запису значення) і ім'я, яке можна

використовувати в тексті програми з одним обмеженням: цьому імені не можна присвоювати інше значення.

ПРИКЛАД.

const Max=55, Min=0.1, Word=”Best”;

У цьому прикладі визначені три іменовані константи: Max – константа цілого типу; Min - константа дійсного типу і Word – константа рядкового типу. Зазначимо, що іменам Max, Min, Word у програмі не можна присвоювати інші значення, оскільки це імена констант, а не змінних, тобто оператор виду Min=Min+0.1; буде некоректним.

З простих констант в C# використовуються числові (цілі і дійсні), рядкові, символьні й логічні.

Числові константи.

Цілочисельні константи можуть бути виражені в десятковій, шістнадцятковій і двійковій формі.

12 – константа типа int

12L – константа типа long

Перед числом в двійковій формі передують символи 0b, після яких йде набір з нулів і одиниць: 0b11 – число 3.

Для запису числа в шістнадцятковій формі застосовуються символи 0x, після яких йде набір символів від 0 до 9 і від A до F, які власне представляють число: 0xFF – число 255.

Числові константи дійсного типу можуть записуватися у двох формах:

1) з фіксованою точкою (звичайний десятковий дріб) і

2) з плаваючою точкою (в експоненційній формі).

У другому випадку для записування числа використовується символ ‘E’ (або ‘e’) : перед E записується десятковий дріб або ціле десяткове число, а після E записується степінь числа 10, наприклад: 2.35E-2. Обчислюється таке число наступним чином: число, записане перед E, множиться на десять у степені, який вказаний після E. Таким чином, 2.35E-2 = 2.35/100=0.0235. За замовчуванням дійсні константи відносяться до типу double. Тому у випадку, коли дійсне

значення повинно бути типу float необхідно застосовувати суфікс F.

У цілочисельних констант за замовчуванням може бути один з наступних типів: int, uint, long, ulong в залежності від значення константи. Явно вказати тип константи можна за допомогою суфікса:

12 – константа типу int

12L – константа типу long

10.23F –константа типу float

Рядкові константи представляють собою послідовність, з нуль або більше символів, використовуваних у даному комп'ютері, і укладену в подвійні лапки - ”Best”. **Символьні константи** записуються як будь-який символ в апострофах – 'a'.

Крім описаних строкових літералів, існують буквальні рядкові літерали. Такий літерал починається з символу @, після якого слідкує рядок в лапках. Буквальні рядкові літерали виводяться в тому ж вигляді, в якому вони введені в початковому тексті програми. Перевага буквальних строкових констант полягає в тому, що вони дозволяють вказати в програмі виведений результат сааме так, як він повинен виглядати на екрані.

В C# є дві **логічні константи** true і false ("істина" і "брехня", відповідно).

Окремим літералом є ключове слово null.

2.4. Змінні.

Змінними називаються дані, які у ході виконання програми можуть змінювати своє значення. Всі змінні, використовувані в програмі, повинні бути описані явно. При описі для кожної змінної задаються її ім'я і тип.

ПРИКЛАД.

```
int Sum1,Sum2;
double p;
```

Під ім'ям розуміється символічне позначення змінної. Під час виконання програми ім'я перетворюється у фізичну адресу змінної. Імена змінних називають ідентифікаторами.

Особливості використання ідентифікаторів:

1. У ідентифікаторі можуть використовуватися літери, цифри і символ підкреслення.
2. Першим символом ідентифікатора може бути буква або знак підкреслення, але не цифра.
3. Довжина ідентифікатора не обмежена.
4. Прогалини всередині імен не допускаються.
5. Ідентифікатор не повинен збігатися з ключовими словами.
6. У ідентифікаторів C # можна використовувати букви різних абеток, наприклад, російської чи грецької.
7. Великі та малі літери розрізняються, наприклад, A і a - два різних імені.

Задати значення змінної можна за допомогою оператора присвоювання. Крім того, задати початкове значення змінної можна при її оголошенні. Цей прийом називається ініціалізацією.

Приклад: int i=10;

При присвоєнні числового значення (константи) змінної типу float слід використовувати суфікс F(f). При присвоєнні змінної типу double або float значення слід ставити після десяткового дробу 0, навіть якщо число не має значущих десяткових знаків. В іншому випадку число може інтерпретуватися компілятором як ціле.

Неявно типізовані змінні.

Неявно типізована змінна оголошується за допомогою ключового слова var і повинна бути неодмінно ініціалізована. Компілятору надається можливість самому визначити тип змінної, виходячи із значення, яким вона ініціалізується.

Приклад.

var x = 2.4578;

В даному прикладі змінна x ініціалізується дійсною константою, яка за замовчуванням має тип double, і тому змінна відноситься до типу double.

Приклад.

var y = 4.5783F; // змінна віднесена до типу float

Явне і неявне перетворення даних.

Мова C# є строго типізованою мовою, тобто контролює типи даних і визначає дозволені для них операції. Арифметичні дії виконуються над змінними одного типу, тому вихідні складові арифметичного виразу повинні бути приведені до одного типу.

Загальне правило: неявно можна виконувати всі перетворення, які не призведуть до втрати інформації. Тому дані типів bool, double, decimal не можуть бути неявно перетворені ні в які типи даних. Тип float може бути перетворений в double; int може бути перетворений в long, float, double, decimal; long може бути перетворений в float, double, decimal.

Явне перетворення виконується наступним чином:

(Новий тип даних) змінна

Приклад.

float x; int i;

x = (float) 56.3; // константа типу double перетворюється в float

i = (int) 8.6; // результат i = 8

У мові програмування C# всім змінним до їх використання в виразах мають бути присвоєні значення, в тому числі нулі і порожні рядки. Використання в виразах неініціалізуємих змінних вважається помилкою!

Мітки.

Під міткою розуміється символічна адреса оператора, призначена для передачі йому управління за допомогою оператора безумовного переходу. В якості мітки використовується довільно обраний програмістом ідентифікатор. В програмі після імені мітки ставиться двокрапка.

ПРИКЛАД.

met1: a=1;

…

goto met1;

Перехід до мітки можливий не з усіх точок програми. Мітки всередині блоку недосяжні для операторів переходу, розміщених поза блоком. У той же час для оператора переходу, розміщеного всередині блоку, дозволений перехід до мітки поза блоком.

Блок - укладена в фігурні дужки послідовність операторів.