



Microsoft Partner
Silver Learning



Basis for Java

Базові конструкції в програмуванні

Тема курсу

Тема уроку

Базові конструкції в програмуванні

Тема курсу

План уроку

1. Поняття синтаксису, семантики та типізації
2. Змінні, типи даних
3. Масиви
4. Операції над даними
5. Умовні конструкції
6. Циклічні конструкції
7. Алгоритми

Тема курсу

Поняття синтаксису, семантики та типізації

Для того щоб говорити та читати будь-якою іноземною мовою, необхідно вивчити алфавіт та граматику цієї мови. Подібна умова спостерігається і при вивченні мов програмування.

Синтаксис в граматиці вивчає будову слів, словосполучень та речень. В програмуванні синтаксис визначає набір правил, що визначає комбінації символів в тексті програми.

Семантика мови – це смислове значення слів. У програмуванні — значення операторів, основних конструкцій мови, тощо.

Тема курсу

Поняття синтаксису, семантики та типізації

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Синтаксис Java

```
1  .model small  
2  .stack 100h  
3  .data  
4  msg      db "Hello World! $"  
5  .code  
6  main:  
7      mov ax, @data  
8      mov ds, ax  
9  
10     mov ah, 09  
11     mov dx, offset msg  
12     int 21h  
13  
14     mov ah, 01  
15     int 21h  
16  
17     mov ax, 4c00h  
18     int 21h  
19  
20 end main
```

Синтаксис Assembler

Тема курсу

Змінні, типи даних

Мова програмування Java є чітко типізованою (strongly typed language). Це означає, що будь-яка змінна та будь-який вираз повинні мати відомий тип ще на момент компіляції.

Деякі типи даних в Java:

- **int** – містить будь-яке ціле значення від -2147483648 до 2147483647;
- **float** – зберігає дробові числа;
- **char** – це символьна константа у форматі Unicode. Будь-який символ цього типу повинен записуватися в одинарних лапках, наприклад: 'G';
- **boolean** – логічний тип, має два значення: **false** – брехня і **true** – істина. Цей тип часто використовується в циклах, про які трохи пізніше.
- **String** в Java — або рядок, це об'єкт, який насправді є масивом символів char. Але на даному етапі досить запам'ятати що це усього лише рядок тексту.

Тема курсу

Масиви

Масив – це змінна, яка може містити не одне, а відразу кілька значень, на відміну від звичайної змінної, що містить єдине значення.

Масив – це такий собі "магічний" контейнер, у який ми можемо покласти відразу кілька значень.

Гарним прикладом може бути поштовий ящик в багатоквартирному будинку. Це одна шафка з декількома пронумерованими скриньками, в кожна з яких поштар кладе, до прикладу, квитанцію на оплату комунальних послуг. Квитанція – це тип даних, а скринька це елемент масиву, що зберігає дані. Набір таких скриньок утворюють масив даних.



```
int [ ] array1 = { 5, 17, 350 }
```

↑ ↑ ↑ ↑

тип даних квадратні дужки ім'я масиву в фігурних дужках
масиву вказують, що масиву вказані числа, які
 це масив ми присвоїли в array1

Тема курсу

Операції над даними

- Арифметичні операції (+, -, *, / та інші);
- Логічні операції (&, |, &&, || та інші);
- Операції відношення (==, !=, >, < та інші);
- Операції над рядками, до прикладу **"Hello "** + **"World"**
(ця операція над рядками називається конкатенація).

Тема курсу

УМОВНІ КОНСТРУКЦІЇ

Умовні конструкції – це конструкції прийняття рішень. Найбільш розповсюджена умовна конструкція є конструкція **if/else**. Після **if** може слідувати вираз **else if ... else**. Дану конструкцію використовується для вибору одного з декількох блоків коду для виконання.

```
if(number > 0) {  
    System.out.println("Number is positive");  
} else if (number < 0) {  
    System.out.println("Number is negative");  
} else {  
    System.out.println("Number is equals zero");  
}
```

Тема курсу

УМОВНІ КОНСТРУКЦІЇ

Оператор **switch** також дозволяє будувати умовні конструкції. Після **switch** іде набір секцій **case**, а в кінці - **default**.

```
switch(number) {  
    case 0: System.out.println("Number is zero"); break;  
    case 1: System.out.println("Number is one"); break;  
    case 3: System.out.println("Number is tree"); break;  
    default: System.out.println("Unknown number");  
}
```

Тема курсу

Циклічні конструкції

Цикл – це конструкція, призначена для організації багаторазового виконання набору інструкцій (команд). Існує декілька циклічних конструкцій в мові програмування Java.

Цикл **for** - управляюча конструкція повторення, застосовується при необхідності виконати інструкції певну кількість разів. Цикл корисний, коли заздалегідь відомо яка повинна бути кількість повторень.

```
for (int i = 0; i < 10; i++) {  
    //Тіло циклу  
}
```

Тема курсу

Циклічні конструкції

Цикл **while** – це цикл з передумовою, тіло циклу (тобто інструкція або блок інструкцій) виконується, якщо умова істинна. Заздалегідь невідомо скільки разів виконається код всередині циклу. Якщо умова хибна з самого початку, то цикл не виконається жодного разу.

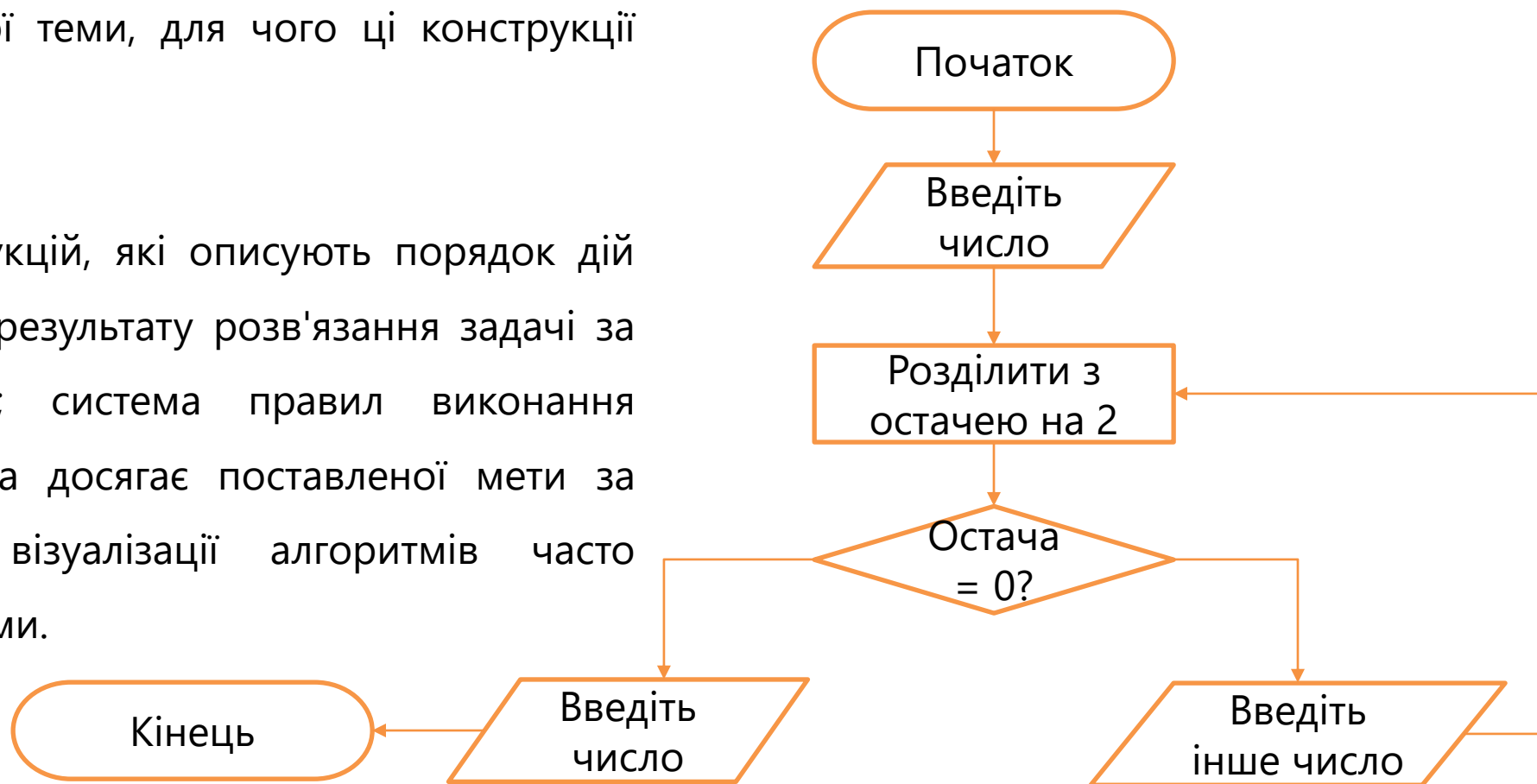
```
while (condition == true) {  
    //Тіло циклу  
}
```

Тема курсу

Алгоритми

Ми підійшли до головної теми, для чого ці конструкції використовуються.

Алгоритм – набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для візуалізації алгоритмів часто використовують блок-схеми.



Алгоритм пошуку парних чисел

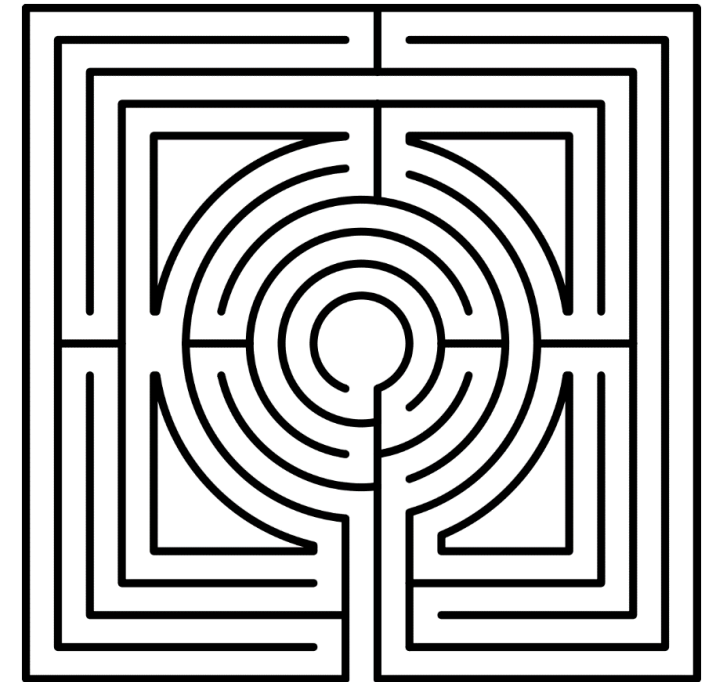
Тема курсу

Алгоритми

Для прикладу розглянемо процес пошуку виходу з лабіринту.

Існує ряд різних алгоритмів для розв'язування лабіринтів, тобто методів автоматичного пошуку виходу. Такі алгоритми, як **метод випадкової поведінки миші**, **«триматися за стіну»**, **застава (англ. Pledge)** та алгоритм **Тремо (англ. Tremaux)** розроблені для проходження лабіринту мандрівником без попереднього вивчення лабіринту, тоді як алгоритми: **заповнення тупиків** та **алгоритм найкоротшого шляху** створені для використання людиною або комп'ютерною програмою, яка має можливість бачити і обробляти весь лабіринт одночасно.

Вибір алгоритму залежить не тільки від бажання розробника, але і від будови лабіринту.



Інформаційний відеосервіс для розробників програмного забезпечення

Навчальний центр
інформаційних технологій
cbs.com.ua

CyberBionic
systematics

Відео курси
з програмування
itvdn.com



CyberBionic
systematics

Microsoft Partner



 Windows Azure



Проверка знаний

TestProvider.com



Перевірте як Ви засвоїли цей матеріал на

TestProvider.com

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT фахівця.

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.

Basis for Java

Дякую за увагу! До нових зустрічей!