

# Системне Програмування

З використанням мови програмування **Rust**

# Основні види програмування

- прикладне
- мобільне (UI)
- фронтенд (UI)
- ігрове
- системне



# Деякі класифікації мов програмування

## - **Компілятори**

- Assembler
- C, C++, C#, F# ...
- Java, Scala, Haskell
- Rust

## - **Інтерпретатори**

- PHP
- Python
- JavaScript
- Ruby

# Деякі класифікації мов програмування

- **Статична типізація**

- C, C++, C#, F# ...
- Java, Scala, Haskell
- Rust

- **Динамічна типізація**

- PHP
- Closure
- Python
- JavaScript
- Ruby



# Деякі класифікації мов програмування

- **Процедурні**

- Assembly
- Pascal
- C, C++

- **Функціональні**

- Scala
- Haskell
- Closure
- OCaml

- **Об'єктно-орієнтовані**

- PHP
- Java
- Python
- JavaScript
- Ruby

# Деякі класифікації мов програмування

## - Без збирача сміття

- Assembly
- C, C++
- Rust

## - Зі збирачем сміття

- Java
- Haskell
- Rust
- JavaScript
- Python
- Ruby



# Деякі класифікації мов програмування

- **Виконуються на віртуальній машині (JVM, CLR, )**
  - Java, Scala, Closure
  - C#, F#, ...
  - Haskell
  - JavaScript
  - Python
  - Ruby
- **Виконуються “на залізі”**
  - C, C++
  - Pascal
  - Rust

# Чому з'являються нові мови програмування

- **Змінюються підходи**
  - Процедурні
  - ООП
  - ФП
- **Змінюються потреби**
  - Швидкість розробки
  - Швидкість виконання
  - Коректність
  - Необхідність розширення мови програмування
  - З'являються специфічні задачі
- **З'являються нові targets**
  - mobile
  - browser
  - VM
- ...



# Чому потрібно вчити нові мови програмування

- Більшість мов програмування мають унікальні підходи, заради яких вони розроблялися
- Комбінуючи ці підходи можна знайти нові підходи, які можуть зробити розробку більш продуктивно та більш коректно
- Не всі мови програмування є достатньо виразними для вирішення специфічних завдань
- Знаючи декілька підходів можливо набагато швидше та коректніше визначити структуру майбутнього проєкту

# Чому Rust?

- Відносно нова мова яка взяла [майже] все найкраще з інших мов
- Гарна екосистема та наявність якісної IDE
- Компілятор (дуууже розумний, з купою підказок стосовно більшості помилок)
- Статична типізація
- Гнучка система типів: підтримка **struct**, **enum**
- Відділення даних (**struct**) від коду (**impl**)
- Відділення декларацій (**trait**) від реалізації (**impl**)
- **No null**
- **No exceptions**
- Extension methods
- Pattern matching / Exhaustiveness checking
- macros / templating
- All targets (native, js, wasm)
- ...



**Код з лекцій,  
презентації Keypnote,  
PDF-файли  
знаходяться на GitHub:**

<https://github.com/djnzx/rust-course>