**Група ІП-51мн**

**Завдання**

Розробити просту імперативну мову програмування загального призначення, синтаксично подібну мові‑прототипу. Мову-прототип кожен студент обирає самостійно із наведеного далі списку, див. Список мов-прототипів.

Синтаксис операторів (statements) циклу (повторення) і умовного (розгалуження), операторів присвоювання, введення, виведення, однорядкових коментарів та інших елементів мови обрати із мови-прототипу.

Семантику усіх елементів мови визначити самостійно (тобто не має значення, чи семантика розробленої мови відповідає семантиці мови-прототипу).

Розроблена мова (**мінімальний** обсяг):

* містить один оператор (statement) циклу;
* містить один оператор (statement) розгалуження;
* підтримує роботу з цілими і дійсними числами та містить оператори (operators):
  + основних чотирьох арифметичних операцій — додавання, віднімання, ділення та множення;
  + піднесення до степеня (правоасоціативна операція);
* визначає логічний тип для представлення значень true/false;
* містить оператори (operators) порівняння: суворі і несуворі більше/менше, дорівнює/не дорівнює;
* дужки.

Розроблена мова (**повний** обсяг), крім того передбачає :

* наявність рядкового типу (подібного типу string);
* наявність іменованих констант;
* визначення та використання функцій користувача.

Результат виконання — специфікація мови програмування.

1. Про структуру і зміст специфікації див. розд. 2.4. у посібнику [1].
2. Приклад специфікації простої імперативної мови програмування див. у файлі «Приклад специфікації. 2025.pdf». Мова визначена виключно для ілюстрації того, як може виглядати специфікація, зокрема у ній не визначено інструкцій розгалуження чи умовного переходу.
3. Наявність «Базового прикладу» програми — обов’язкова. Базовий приклад програми — це (лексично, синтаксично і семантично правильна) програма, у якій зустрічаються приклади усіх лексем із таблиці символів (таблиці токенів, таблиці лексем) мови.
4. У випадку спільного проєкту:
   1. кожен виконавець обирає власний варіант синтаксису (і семантики) інструкцій повторення та розгалуження (або модифікує наявний у мові-прототипі);
   2. кожен виконавець надає власну специфікацію (звіт);
   3. кожен виконавець особисто представляє/захищає роботу.

Вимоги до оформлення:

* титульний аркуш має відповідати зразку, див. Зразок титульного аркуша   
  (з очевидним необхідним редагуванням);
* текст специфікації має бути граматично правильним, відформатованим, зручним для читання;
* синтаксичні діаграми мають бути згенерованими за граматикою (у жодному разі не створеними вручну!). Рекомендований інструмент — EBNF-Vizualiser, або інший, наприклад, якийсь зі згаданих у розд. 2.2.4 у навч. посібнику [1];

Література

1. Стативка Ю.І. Формальні мови: Основні концепти i представлення [Текст]: навч. посіб. / Ю. І. Стативка. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 87 с.

Список мов-прототипів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Мова - прототип | Посилання | Розробники  (не більше трьох) | Спільний проєкт  так/ні |
| 1 | Ada | http://www.ada-auth.org/standards/rm12\_w\_tc1/RM.pdf |  |  |
|  |  |
|  |  |
| 2 | Dart | https://dart.dev/language |  |  |
|  |  |
|  |  |
| 3 | Delphi | https://docwiki.embarcadero.com/RADStudio/Alexandria/en/Delphi\_Language\_Reference |  |  |
|  |  |
|  |  |
| 4 | Go | https://go.dev/doc/effective\_go |  |  |
|  |  |
|  |  |
| 5 | Grovy | https://groovy-lang.org/documentation.html |  |  |
|  |  |
|  |  |
| 6 | Kotlin | https://kotlinlang.org/spec/introduction.html#introduction  https://play.kotlinlang.org/byExample/02\_control\_flow |  |  |
|  |  |
|  |  |
| 7 | R | https://cran.r-project.org/doc/manuals/r-release/R-lang.pdf  https://www.r-project.org/ |  |  |
|  |  |
|  |  |
| 8 | Ruby | https://www.ruby-lang.org/en/documentation/ |  |  |
|  |  |
|  |  |
| 9 | Scala | https://docs.scala-lang.org/ |  |  |
|  |  |
|  |  |
| 10 | Swift | https://www.swift.org/documentation/ |  |  |
|  |  |
|  |  |

Зразок титульного аркуша

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ**

**СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Лабораторна робота № 1**

ПРОЄКТУВАННЯ ТА СПЕЦИФІКАЦІЯ   
ПРОСТОЇ ІМПЕРАТИВНОЇ МОВИ ПРОГРАМУВАННЯ   
ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

(з дисципліни «Побудова компіляторів»)

Учасники проєкту:

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Звіт студентки І курсу, групи ІП-51мн

спеціальності F2 Інженерія програмного забезпечення

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

(Прізвище, ім’я, по батькові) (Підпис)

Перевірив: доцент, к.т.н. Стативка Ю.І. \_ \_ \_ \_ \_ \_ \_ \_ \_

(Посада, науковий ступінь, прізвище та ініціали) (Підпис)

Київ – 2025