**Практична робота №01. Порівняння мов програмування.**

**Мета**: навчитися писати аналізувати можливості мов програмування як інструменту розроблення власних програм**.**

**Завдання.**

Переглянути в Інтернет описи мов програмування С++, С#, Java. JavaScript, Pyton. Заповнити надану нижче таблицю для кожної мови, провести аналіз та надати свій висновок. В рядку "Основні особливості" вказати можливості мови, які відсутні в інших (конкретизувати, вказавши мову, яка не має таких властивостей). В рядку "Складність вивчення" визначте оцінку від 1 до 5, за якою ви визначаєте для себе складність вивчення цієї мови у порівнянні між цими мовами. У висновку визначити мову програмування, яку б ви хотіли використовувати, і причини, які спонукали вас зробити цей вибір.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Характеристика | С++ | С# | Java | JavaScript | Python |
| Область використання | Системне програмування, написання драйверів, розробка прикладного програмного забезпечення | Розробка ігр, веб-сервісів, розробка додатків для Windows, хмарні сервіси | Мобільні додатки, серверні розробки | Веб-розробка(фронт-енд, бек-енд), мобільні додатки | Веб-розробка, автоматизація, машинне навчання |
| Механізм використання | Компільована мова (g++, MSVC, clang) | Компільована в байткод (CLR, .NET) | Компільована в байткод(JVM) | Інтерпретована | Інтерпретована |
| Парадигми, які використовуються | Процедурна, об'єктно-орієнтовна і узагальнена | Об'єктно-орієнтована, імперативна, функціональна | Об'єктно-орієнтована, імперативна, структурна | Імперативна, Об'єктно-орієнтована | Імперативна, функціональна, об’єктно-орієнтована |
| Основні особливості | Прямий доступ до пам'яті, стандартизація, продуктивність | Статична типизація, автоматичне управління пам'яттю | Портативність, стабільність, велика екосистема | Динамічна типізація, DOM-доступ, асинхронність, універсальність | Проста у синтаксис, AI/Data Science, велика бібліотека |
| Наявність стандарту мови | Так(ISO/IEC JTC1/SC22/WG21) | Так (ECMA, .NET Standard) | Так (Java SE Oracle) | Так (ECMAScript — стандарт мови JavaScript) |  |
| Складність вивчення | 4 | 3 | 3 | 2 | 1 |

**Джерела:**

1. Мова програмування С ++: <https://promoter.net.ua/articles/mova-programuvannya-s-plus-plus.html>
2. С#: що це за мова та де її використовують: <https://robotdreams.cc/uk/blog/284-s-chto-eto-za-yazyk-i-gde-ego-ispolzuyut>
3. Сторінка мови Java: <https://uk.wikipedia.org/wiki/Java>
4. Ключові характеристики та функціональність JavaScript: <https://www.freshcode.training/blog-post/what-is-javascript-part-3>
5. Путівник мовою програмування Python: <https://pythonguide.rozh2sch.org.ua/>

Контрольні запитання

1. Мова програмування це формальна знакова система, призначена для опису алгоритмів у формі, яка зручна для комп’ютера.
2. Специфікація задач, кваліфікації самого програміста, апаратної частини ПК та програмної платформи.
3. Мнемокодом називають мови асемблера. Він використовує текст, який легко запам’ятати, замість двійкових кодів інструкцій процесора.
4. Fortran, яка призначався для швидких наукових обчислень.
5. Fortran, яка була створена 1957 року Джоном Бекусом.
6. Перше покоління(початок 1950-х років); Друге покоління(кінець 1950-х – початок 1960-х р.р.); Третє покоління(1960-ті роки); Четверте покоління(початок 1970-х років – по сьогодення).
7. На мою думку поділ йшов відносно характеристик й функціоналу самих мов, починаючи з третього покоління починають з’являтися високорівневі мови, а з четвертого наближення до мови людини.
8. Мови обробки даних, скриптові мови, об’єктно-орієнтовні мови, мови паралельного програмування, неімперативні мови, функціональні мови, мови логічного програмування
9. Особливість полягає в тому, що програми на них, є покроковим описом рішення тієї або іншої задачі. До них відносяться: Python, JavaScript, Perl та інші.
10. Компілятори спочатку повністю перекладають весь текст програми з високорівневої мови на мову машинних команд, щоб можна було запускати отриману в результаті цього машинну програму. Коли інтерпретатори натомість читають текст програми мовою високого рівня та виконують його по мірі прочитання. Переклад програми на машинну мову не запам’ятовується, отже для повторного виконання, її знову треба пропустити через інтерпретатор
11. До відомих компіляторів можна віднести: GCC, Clang, javac і тд. Інтерпретатори: Python Interpreter (CPython), Perl Interpreter, Ruby Interpreter (MRI), Bash (Shell)