Міністерство освіти і науки, молоді та спорту
Національний технічний університет України
“Київський політехнічний інститут”

Факультет прикладної математики
Кафедра прикладної математики

РЕФЕРАТ

з дисципліни “Англійська мова професійного спрямування
на тему: **“Programming languages”**

Виконала: Лук’яненко А. М.
група КМ-51

Викладач: Чижова Н.В.

Київ 2017

CONTENTS

INTRODUCTION

A programming language is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for a computer. Programming languages can be used to create programs that implement specific algorithms.

The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning). Some languages are defined by a specification document (for example, the C programming language is specified by an ISO Standard) while other languages (such as Perl) have a dominant implementation that is treated as a reference. Some languages have both, with the basic language defined by a standard and extensions taken from the dominant implementation being common.

1 MAIN PART

A programming language is a formal computer language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs to control the behavior of a machine or to express algorithms.

This is a list of notable programming languages, grouped by type.

1.1 Assembly languages

An assembly (or assembler) language, often abbreviated «asm», is a low-level programming language for a computer, or other programmable device, in which there is a very strong (but often not one-to-one) correspondence between the language and the architecture's machine code instructions. Each assembly language is specific to a particular computer architecture. In contrast, most high-level programming languages are generally portable across multiple architectures but require interpreting or compiling. Assembly language may also be called symbolic machine code.

Assembly language is converted into executable machine code by a utility program referred to as an assembler. The conversion process is referred to as assembly, or assembling the source code. Assembly time is the computational step where an assembler is run.

Assembly languages directly correspond to a machine language so machine code instructions appear in a form understandable by humans. Assembly languages lets programmers use symbolic addresses, which the assembler converts to absolute addresses. Most assemblers also support macros and symbolic constants.

1.2 Object-oriented programming (OOP)

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated. In OOP, computer programs are designed by making them out of objects that interact with one another.

There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which typically also determine their type.

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce,

web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms. Many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM.

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.

It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is a general-purpose, object-oriented programming language. Its development team is led by Anders Hejlsberg. The most recent version is C# 7.1, which was released in 2017 along with Visual Studio 2017 Update 3.

1.3 Data-oriented languages

Data-oriented languages provide powerful ways of searching and manipulating the relations that have been described as entity relationship tables which map one set of things into other sets.

SQL Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).

SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

The Wolfram Language, a general multi-paradigm programming language developed by Wolfram Research, is the programming language of Mathematica and the Wolfram Programming Cloud. It emphasizes symbolic computation, functional programming, and rule-based programming and can employ arbitrary structures and data.

It includes built-in functions for generating and running Turing machines, creating graphics and audio, analyzing 3D models, matrix manipulations, and solving differential equations. It is extensively documented.

The Wolfram language was released for the Raspberry Pi in 2013 with the goal of making it free for all Raspberry Pi users. It was controversially included in the recommended software bundle that the Raspberry Pi Foundation provides for beginners. Plans to port the Wolfram language to the Intel Edison were announced after the board's introduction at CES 2014. There was also a short lived proposal to make Wolfram libraries compatible with the Unity game engine, giving game developers access to the language's high level functions.

2 GLOSSARY

Implementation	Реалізація	Stream processing	Обробка потоку
To treat	Розглядати		
Extension	Розширення	Relational	Реляційний
Particularly	Особливо	Game engine	Ігровий двіжок
Notable	Видатний	Emphasize	Підкреслює
Correspondence	Відповідність	Adjustments	Коригування
Assembly time	Час збірки	Despite	Незважаючи на
Computational	Обчислювальний	Entirely	Повністю
Associated	Взаємодіють	Adhering	Дотримуватися
Significant	Значний	Revised	Переглянутий
Diversity	Розмаїття	Employ	Використовувати
Instance	Випадок	Arbitrary	Довільний
Concurrent	Одночасно	Extensively	Широко
Intended	Призначений	Controversially	Суперечливо
Regardless	Незалежно	Software bundle	Пакет ПЗ
Imperative	Вимагальний	Board’s introduction	Введення плати
Facility	Можливість		
Bias	Упередження	Compatible	Сумісний
Embedded	Вбудований	General-purpose	Загально-призначений
Constrained	Обмежений		
Performance	Продуктивність	Available	Доступний
Highlights	Основні моменти	Flexibility	Гнучкість
Key strength	Ключові переваги	Low-level	Низько-рівневий
Performance-critical	Критично-важливі	Access	Доступ
		Modify	Модифікувати
Switch	Комутатор	Directly	Строго
Space probe	Космічний зонд	Source	Джерело
Vendor	Постачальник	Require	Вимагає
Foundation	Фонд	Application	Додаток
Encompassing	Охоплює	Development	Розробка
Strong typing	Контроль типів	Precision	Точність
Approved	Схвалений	To port	Впровадити
Relations	Відношення	Ambiguous	Неоднозначність
To map	Відображати	Intent	Намір
Entity	Сутність	Figuratively	Образно
Query	Запит	External	Зовнішній
To hold	Зберігати	Interleaves	Пов’язує

SUMMARY

Programming languages differ from most other forms of human expression in that they require a greater degree of precision and completeness. When using a natural language to communicate with other people, human authors and speakers can be ambiguous and make small errors, and still expect their intent to be understood.

However, figuratively speaking, computers "do exactly what they are told to do", and cannot "understand" what code the programmer intended to write. The combination of the language definition, a program, and the program's inputs must fully specify the external behavior that occurs when the program is executed, within the domain of control of that program.

On the other hand, ideas about an algorithm can be communicated to humans without the precision required for execution by using pseudocode, which interleaves natural language with code written in a programming language.

Мови програмування відрізняються від більшості інших форм людського вираження, оскільки вони потребують більшої точності та повноти. Використовуючи природну мову для спілкування з іншими людьми, автори та виступаючі можуть бути неоднозначними та робити невеликі помилки, і все ще очікують, що їхній намір буде зрозумілий.

Однак, образно кажучи, комп'ютери "роблять саме те, що їм кажуть робити", і не можуть "зрозуміти", який код програміст мав намір написати. Комбінація визначення мови, програми та вхідних даних повинне повністю вказувати зовнішню поведінку, яка виникає при виконанні програми, в межах контролю над цією програмою.

З іншого боку, ідеї про алгоритм можуть бути передані людям без точності, необхідної для виконання, за допомогою псевдокоду, який пов’язує природну мову з кодом, написаним на мові програмування.

REFERENCES

1. Introduction to Programming Languages [Електронний ресурс] – Режим доступу http://www.emu.edu.tr/aelci/Courses/D-318/D-318-Files/plbook/intro.htm
2. Information Technology - Database Language SQL [Електронний ресурс] – Режим доступу http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt
3. Assembly language [Електронний ресурс] – Режим доступу https://dmoztools.net/Computers/Programming/Languages/Assembly/
4. Dr. Alan Kay on the Meaning of "Object-Oriented Programming" [Електронний ресурс] – Режим доступу http://userpage.fu-berlin.de/~ram/pub/pub_jf47ht81Ht/doc_kay_oop_en
5. Java's 20 Years of Innovation [Електронний ресурс] – Режим доступу https://www.forbes.com/sites/oracle/2015/05/20/javas-20-years-of-innovation/
6. C++ Has Become More Pythonic [Електронний ресурс] – Режим доступу http://preshing.com/20141202/cpp-has-become-more-pythonic/
7. What's new in C# [Електронний ресурс] – Режим доступу https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/
8. SQL [Електронний ресурс] – Режим доступу https://www.britannica.com/technology/SQL
9. A guide to community resources about Wolfram Language [Електронний ресурс] – Режим доступу http://wolframlanguage.org/