14 Programming

Topic

Programming process, programming languages

Learning objectives

Students will be able:

- To understand the basics of programing
- To describe steps in a programming process
- To describe and use flowcharts
- To list the main programming languages
- To distinguish between a high-level programming language and a low-level programming language
- To explain the term machine code
- To describe the function of assemblers, compilers and interpreters
- To acquire vocabulary connected with programming

Key words

Programming, flowchart, coding, machine code, bug, debugging, source code, object code, programing languages, low-level programming language, high-level programming language, assembler, compiler, interpreter, object-oriented programming, markup language, markup tag, algorithm.

Programming

Programming is the process of writing a program using a computer language. A program is a set of instructions which a computer uses to do a specific task.

Algorithm is a defined set of steps that can be used to complete a task.

A **flowchart** is a type of diagram that represents an algorithm or process. It can give a step-by-step solution to a given problem. Process operations are represented in the boxes of various kinds connected with arrows. Flowcharts are used in analyzing and designing programs.

Programming languages

Low-level languages

are languages that are very close to machine language. They are easy for the computer to understand, but more difficult for a programmer. The only language the PC can directly execute is **machine code**. This code is binary, consisting of 1s and 0s. This language is difficult to write, so we use symbolic languages. Machine code is often machine specific – the machine code of one computer sometimes is not understood by a different type of a computer.

Assembly languages are symbolic languages using abbreviations such as ADD, SUB, MPY to represent simple instructions. The program then has to be translated into machine code by software called an **assembler**.

High-level languages

Are closer to human languages, commands are written in a form that resembles English. They are not machine specific. Once the program has been written, it can be used on various types of computers.

- **FORTRAN** was developed by IBM in 1954 and is still used in scientific and mathematical applications.
- **COBOL** (**C**ommon **B**usiness **O**riented **L**anguage) was developed in 1959 mainly for business data processing because it offers excellent file handling
- **BASIC –** developed in 1960s was easy to learn and used as a teaching language. **Visual BASIC** is its modern version and is used to build graphical elements such as buttons and windows in Windows programs.
- **PASCAL** was created in 1971 and used to teach the fundamentals of programming.
- **C** was developed in the 1980s and is used to write system software, graphics and commercial applications.
- **C++** is a version of C incorporating object-oriented programming, the programmer concentrates on one piece of text, graphics, etc. and gives it functions which can be altered without changing the entire program. The programs are thus easier to modify. It was used to produce the Microsoft Windows operating system.
- **JAVA** is based on C++ but optimized to run on the Web. It was originally designed for programming small electronic devices such as mobile phones. Java is somewhat simpler and easier to learn than C++ and has characteristics that give it other advantages over C++. **Java applets** are small programs that run automatically on web pages and let you watch animated characters play music and games.

Programs written in high-level languages have to be translated into a machine code by a **compiler** or an **interpreter**.

An **interpreter** takes each instruction in turn, converts it into machine code and carries it out. If the document needs to be read again, the same process has to be repeated.

A **compiler** converts the whole program into machine code in one go. The conversion – **object code** - can be written to a disk for repeated use.

Markup languages

Are languages used to create web documents. They use instructions, known as **markup tags**, to format and link text files.

- **HTML** or **H**yper**T**ext **M**arkup **L**anguage defines the way how images, multimedia, and text are displayed in web browsers. It includes elements to connect the documents (hypertext) and make the web documents interactive. It uses tags to define the structure of your text. Elements and tags are defined by the < and > characters (angle brackets).
- **XML—**e**X**tensible **M**arkup **L**anguage is not limited by a fixed set of pre-defined tags as HTML, it enables the programmer to define his own tags.
- **XHTML -** a short for E**x**tensible **H**yper**T**ext **M**arkup **L**anguage, a hybrid between HTML and XML.
- **Voice XML** makes Internet content accessible via speech recognition and phone. Instead of using a web browser on a PC, you can use a telephone to access voice-equipped websites. You just dial the phone number of the website, give spoken instructions and get the required information.

Debugging

Is the technique of detecting and correcting errors (or **bugs**) which may occur in a program. To **debug** a program or hardware device is to start with a problem, isolate the source of the problem, and then fix it.

There are three main types of errors:

- **System errors** affect the computer as a whole or its peripherals. It can make the computer stop working altogether and you will have to restart it.
- **Syntax errors** are mistakes in the programming language, for example typing errors, errors in sequence of characters. The program does not run.
- **Logic error** is a **bug** in a program that causes it to operate incorrectly, but not to terminate abnormally (or crash). A logic error produces unintended or undesired output or other behavior, although it may not immediately be recognized. Unlike a program with a syntax error, a program with a logic error is a valid program in the language, though it does not behave as intended. The only clue to the existence of logic errors is the production of wrong solutions.

Vocabulary

	Definition	**Translation**
algorithm	a precise step-by-step plan for a procedure that begins with an input and leads to an output in a finite number of steps	algoritmus
assembler	a program converting a program written in a low-level language into a machine code	asembler
assembly language	a low-level language that uses abbreviations to represent instructions	kompilační jazyk
bug	an error in a computer program	chyba v programu

coding	the process of writing instructions for a computer	kódování
compiler	a program converting a source program written in a high-level language into a machine code in one go	kompilátor
debugger	the program used to test programs and correct errors	ladicí program
debugging	detecting and correcting errors in programs	ladění programu
flowchart	a diagram that shows the logical steps of a computer program	vývojový diagram
high-level programming language	a language in which each statement represents several machine code instructions	vyšší programovací jazyk
interpreter	a program translating the source code line by line into the machine code as the program is running	interpret, překladač
low-level programming language	a language that is very closed to machine code	nižší programovací jazyk
machine code	binary code, consisting of 1s and 0s, the only language that a computer can understand directly	strojový kód
markup language	a language using instructions called markup tags to format and link web documents	značkový jazyk
markup tag	a command used in HTML, inserted in a document that specifies how the document, or a portion of the document, should be formatted	značka, tag
object code	an intermediary form of transforming the program from a high-level language to machine code, produced by a compiler	objektový kód
program	a set of instructions that tell the computer what to do	program
programming language	a computer language used for coding computer programs	programovací jazyk
programming	the process of writing a program using a programming language	programování
source code	computer instructions written in a high-level programming language	zdrojový kód

Summary

Programming is the process of writing a program using a computer language. A program is a set of instructions which a computer uses to do a specific task.

Algorithm is a defined set of steps that can be used to complete a task.

A flowchart is a type of diagram that represents an algorithm or process. It can give a step-by-step solution to a given problem. Process operations are represented in the boxes of various kinds connected with arrows.

Programming languages are languages used for coding computer programs, there are low-level programming languages which are very closed to the machine code and high-level programming languages the commands of which are written in a form that resembles English.

A program written in a programming language has to be converted into the machine code by special programs called assemblers, interpreters and compilers.

Finally the program has to be tested with sample data to see if there are any bugs or errors. The process of correcting them is called debugging.

Tasks

1) Steps in a programming process

Listening: Infotech p. 120. Listen and complete the text.

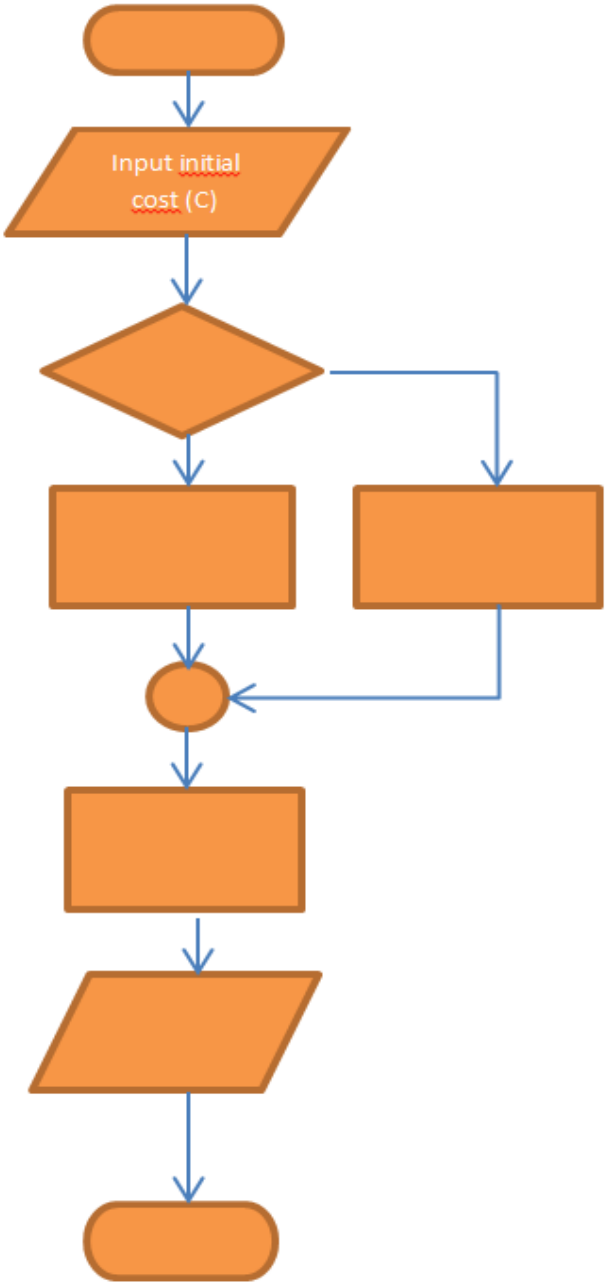
1. The programmer has to understand exactly what the problem is and decide a _____ how to solve the problem.
2. Then he or she designs a _____ – a diagram that shows the successive logical steps of the program, where data is input, output, where to make decision, etc.
3. Next he or she writes the _____ in a _____ _____ (Pascal, C, etc.) This is called coding. These instructions are called _____. Then the source code has to be converted into _____ using a _____.
4. When the program is written it has to be tested with sample data to see if there are any _____ or _____. The process of correcting them is called _____. The programmer has to find the origin of each error, write the correct instruction, compile the program again and test until it runs smoothly.
5. Finally the _____ has to be written – a detailed description of how to use the program

2) Identify the following symbols used in flowcharts and match them with their definitions.

a. **b.** **c.** **d.** **e.**

1. A connector symbol, it is used when two separate paths through a process come together. It is always empty.
2. The parallelogram is the INPUT or OUTPUT symbol. It contains words like Input, print, etc.
3. The ellipse is the START or STOP symbol used at the beginning and end of a flowchart.
4. The diamond shape is the decision symbol. It is used whenever a decision has to be made. It often contains comparison function such as “greater than” or “less than”. It has a “YES” or “TRUE” branch at one corner and “NO” or “FALSE” branch at another.
5. The rectangle is the OPERATION or PROCESS symbol. It indicates the kind of operation. It can contain words like “ADD”, “SUBTRACT”, “MULTIPLY” or “DIVIDE”.

3) Fill in the following flowchart according to the teacher´s instructions.



4) Draw a flowchart for the following procedures:

- Making a cup of tea
- Sending a text message
- Making a telephone call

5) Listening Infotech p. 128

B Complete this extract from a lecture handout about Java with the correct form of the verbs in the box.

call	be	begin	can	decide	rename	have	support	develop	base

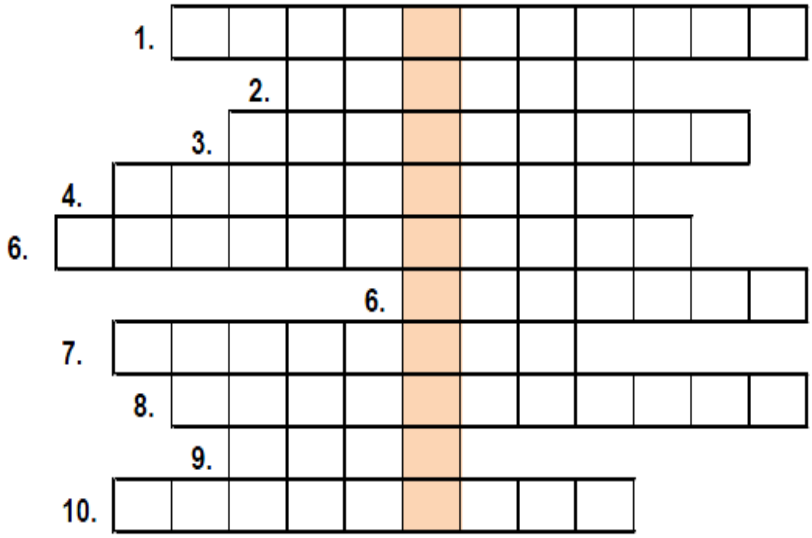
The idea for Java started in 1990, when a team of software engineers at Sun Microsystems (1) _____ to create a language for a handheld device that could control and interact with various kinds of electronic appliances, ranging from Nintendo Game Boys to VCRs and TV set-top boxes. They (2) _____ an object-oriented programming language that one of the engineers, James Gosling, (3) _____ *Oak*, after the tree outside his window. The device even (4) _____ an animated character named *Duke*, who would go on to become Java's mascot.

With the advent of the Web in 1993, the company made a web browser (5) _____ on the Oak language. Later on, this language was adapted to the Internet and (6) _____ *Java*. The 1.0 version of Java was officially introduced by Sun in May 1995.

At that time, web pages (7) _____ only display text, pictures and hyperlinks. With the arrival of Java, web designers (8) _____ able to include animation and interactive programs on web pages. The first major application created with Java was the HotJava browser. The Java language (9) _____ to attract serious attention from the internet community and was soon (10) _____ by Netscape Navigator and MS Internet Explorer. Today, Java is a hot technology that runs on multiple platforms, including smart cards, embedded devices, mobile phones and computers.

C  **Listen to an extract from the lecture and check your answers to C. Listen carefully to the pronunciation of the verbs that end in *-ed*.**

6) Complete the puzzle, solution – a competitive technology to Java



1. small programs that run automatically on web pages and let you watch animated characters play music and games
2. the process of writing instructions for a computer
3. a precise step-by-step plan for a procedure
4. a program converting a program written in a low-level language into a machine code
5. binary code, the only language that a computer can understand directly
6. a high-level programming language used in scientific and mathematical applications.
7. a program converting a source program written in a high-level language into a machine code in one go
8. a high-level programming language used to build graphical elements such as buttons and windows in Windows programs
9. instructions used in markup languages
10. a diagram that shows the logical steps of a computer program

Questions

1. What is programming?
2. Describe briefly steps in the programming process.
3. What is an algorithm?
4. What is a flowchart?
5. What is a programming language?
6. What is the machine code?
7. What are low-level programming languages? Give examples.
8. What is an assembler?
9. What are high-level programming languages? Give examples.
10. What is a compiler?
11. What is an interpreter?
12. What is the source code?
13. What is the object code?
14. What are markup languages? Give examples.
15. What are markup tags?

16. What are bugs?
17. What is debugging?
18. Name and explain the main types of errors.
19. What is Java?

Literatura

[1] ESTERAS, Santiago Remarcha. *Infotech : English for computer users*. 4th ed. Cambridge : Cambridge University Press, 2008. 168 s. ISBN 978-0-521-702997.

[2] ESTERAS, Santiago Remarcha; FABRÉ, Elena Marco. *Professional English in Use : ICT*. 1st ed. Cambridge : Cambridge University Press, 2007. 118 s. ISBN 978-0-521-68543-6.

[3] GLENDINNING, Eric H.; MCEVAN, John. *Basic English for Computing*. Revised and Updated. Oxford : Oxford University Press, 2003. 136 s. ISBN 0194574709.

[4] CUSHING, Steve. *ICT*: *written by GCSE examiners*. London: Letts, 2005. ISBN 1-84315-511-7.

[5] http://www.webopedia.com

[6] http://thecomputereducation.com

[7] http://www.teach-ict.com