Advanced Programming Methods

Iuliana Bocicor
maria.bocicor@.ubbcluj.ro

Babes-Bolyai University

2024

Overview

Software Development

Software Development Methodologies

Software Design Principles

Introduction in Java

Syntax
- Syntax basics
- Data types

Summary

Software Development I

- Software development is a very complex, challenging task, that involves many diverse activities.
- The software development life cycle includes several phases:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Testing and integration
  - Deployment
  - Maintenance

Software Development II

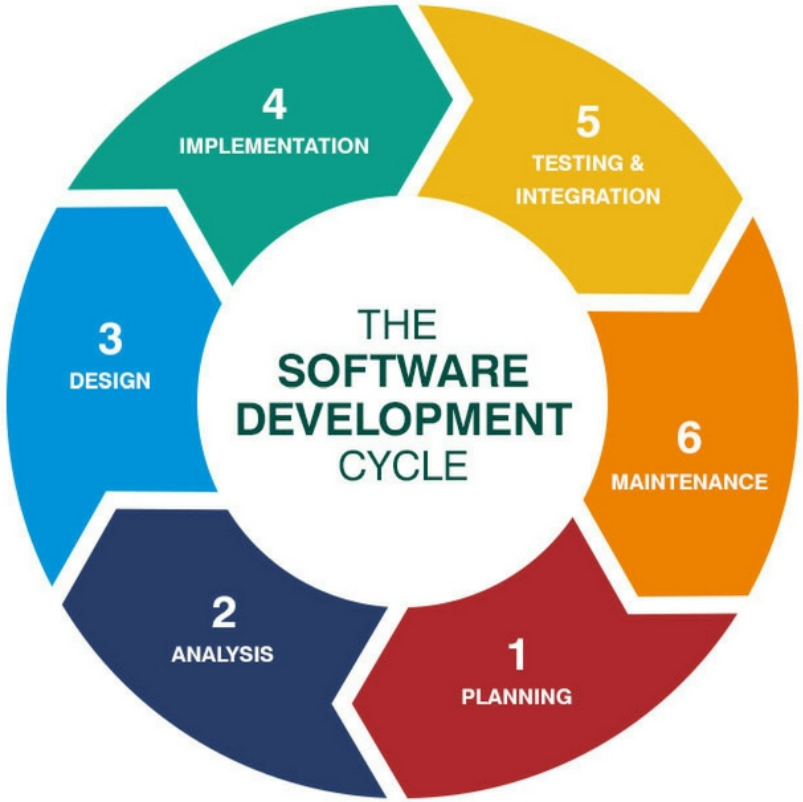


Figure: Figure source: medium.com

Software Development III

- Difficulties in the software development process arise from:
  - Changing specifications.
  - Dynamics and evolution of technologies and standards.
  - Heterogeneous teams (that might also be distributed across the world).
  - The necessity to make predictions and estimates.
  - Poor user communication (feedback, inputs).
  - Integration issues.

Software development methodologies

- A software development methodology is a set of guidelines, rules, values and principles used in the software development process.
- Such models provide a structure and a management direction for the software development process.
- Each model has advantages and limitations.
- The best methodology for a project will be chosen according to the task, risk management, costs, predictability, the necessity for progress demonstration and customer involvement and feedback.

Heavyweight Methodologies

- Traditional way of developing software.
- The steps to be performed are sequential (requirements definition, solution build, testing and deployment).
- The flow of development is unidirectional and each phase in development has specific deliverables.
- A steadfast set of requirements is defined at the beginning of the project.
- Waterfall, Spiral Model, Rational Unified Process.

Agile Methodologies

"An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organising teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders."

Source: Moniruzzaman, A. B. M., and Hossain, D. S. A. (2013). Comparative study on agile software development methodologies. arXiv preprint arXiv:1307.3356.

Traditional vs Agile I

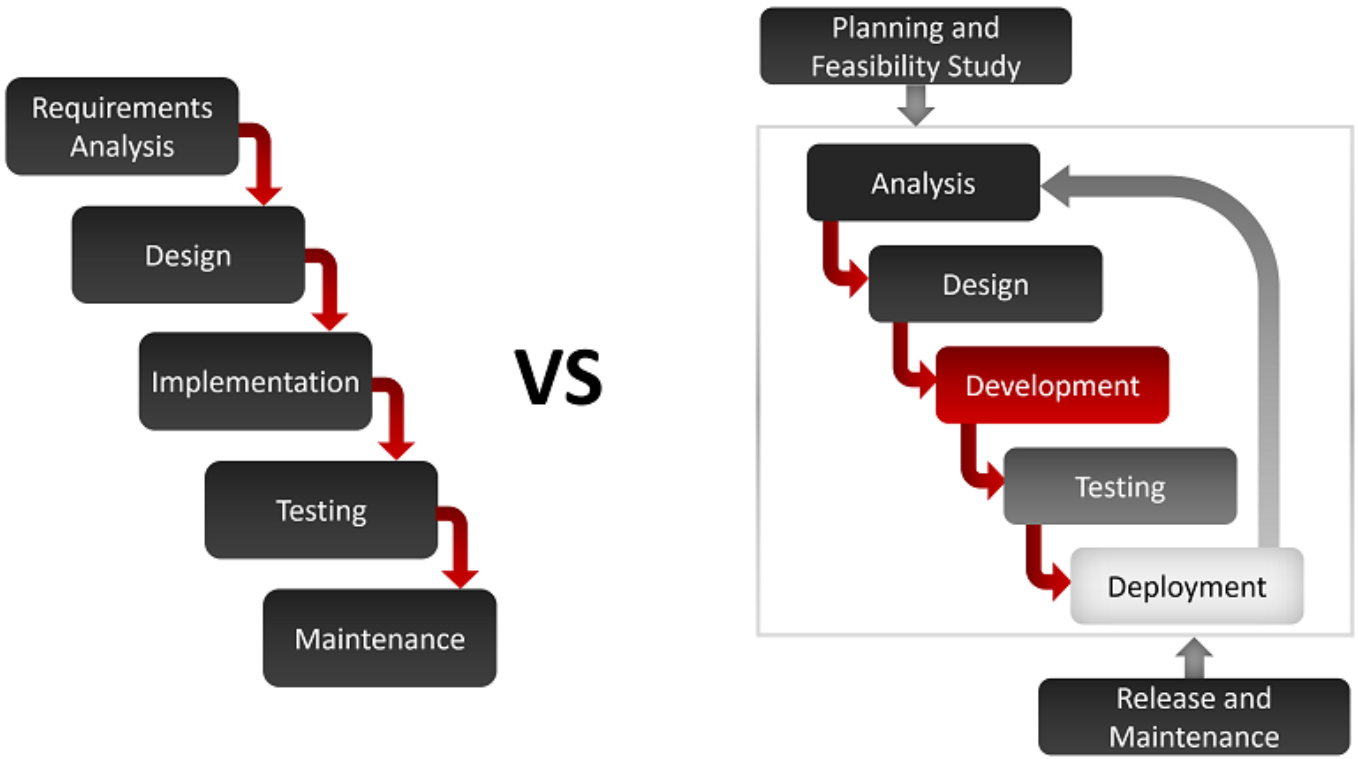


Figure: Figure source: KPI Partners

Traditional vs Agile II

Property	**Traditional**	**Agile**
User requirements	Clearly defined before implementation.	Dynamic, interactive, can be updated periodically.
Involvement of clients	Low, more at project start.	High, communication during project development.
Organisational structure	Linear.	Iterative.
Development model	Traditional life cycle.	Evolutionary delivery.
Model preference	Favours anticipation.	Favours adaptation.
Model suitability	Suitable for situations where requirements are well understood from the beginning.	Suitable for dynamic projects.

Software Design Principles I

- These are a set of guidelines (best practices) for developers, helping them to design software that will be clear and maintainable.
- GRASP (**G**eneral **R**esponsibility **A**ssignment **S**oftware **P**atterns): controller, creator, indirection, information expert, low coupling, high cohesion, polymorphism, protected variations, and pure fabrication.
- SOLID principles:
  - *Single Responsibility Principle*: Each class should be responsible for a single part or functionality of the system.
  - *Open-Closed Principle*: Software components should be open for extension, but closed for modification.

Software Design Principles II

- *Liskov Substitution Principle*: Objects of a superclass should be replaceable with objects of its subclasses without altering any of the desirable properties of the program (without breaking the system).
- *Interface Segregation Principle*: Clients should not be forced to depend upon interfaces that they do not use.
- *Dependency Inversion Principle*: High-level modules should not depend on low-level modules, both should depend on abstractions.

- Design patterns: Factory Method, Adapter, Composite, Decorator, Observer, Strategy.

Introduction in Java I

Advantages:

- Easy to learn and use.
- Object Oriented and functional (beginning with Java 8).
- Platform independent.
- Rich APIs.
- Robust (compiler, exceptions, garbage collection).
- Secure.
- Web programming, mobile applications.
- Distributed.
- Multithreaded.

Disadvantages: slower and more memory consuming than programming languages such as C or C++.

Introduction in Java II

Figure: Figure sources: Sutter's Mill, catb.org

Java vs. C++

Java	**C++**
Supports interfaces.	Does not have the notion *interface*.
Does not allow multiple inheritance.	Allows multiple inheritance.
Automatic polymorphism.	Explicit polymorphism.
Responsibility: system	Reponsibility: programmer.
Does not have pointers, uses reference values (references).	Use of pointers.
Everything must be inside a class.	Functions and data can reside outside classes.
Platform-independent.	C++ executable files are platform dependent.

Program compilation and execution

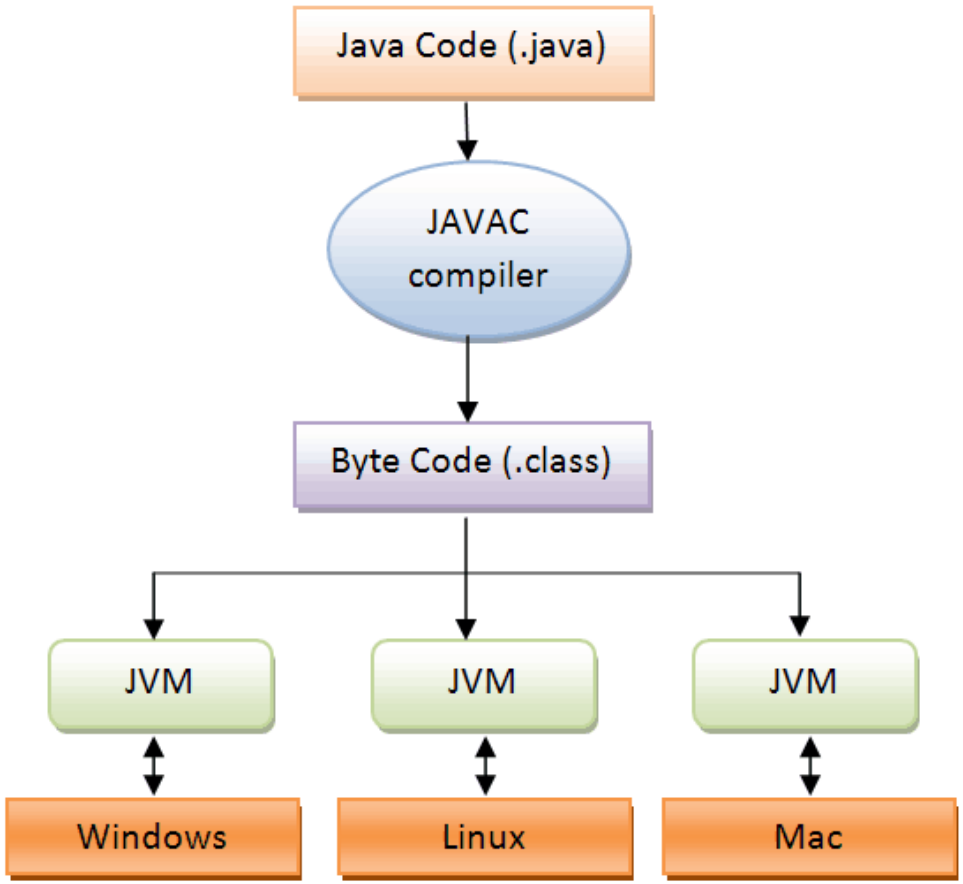


Figure: Figure source: javalearningonline

Java Virtual Machine (JVM)

- Is the runtime environment of the Java platform.
- Converts Java byte code into machine language and then executes it.
- It includes a JIT (Just In Time compiler) that converts byte code into machine language.
- Enables any program written in Java to run on any computer that has a native JVM.
- Manages and optimises program memory.

A small comparison I

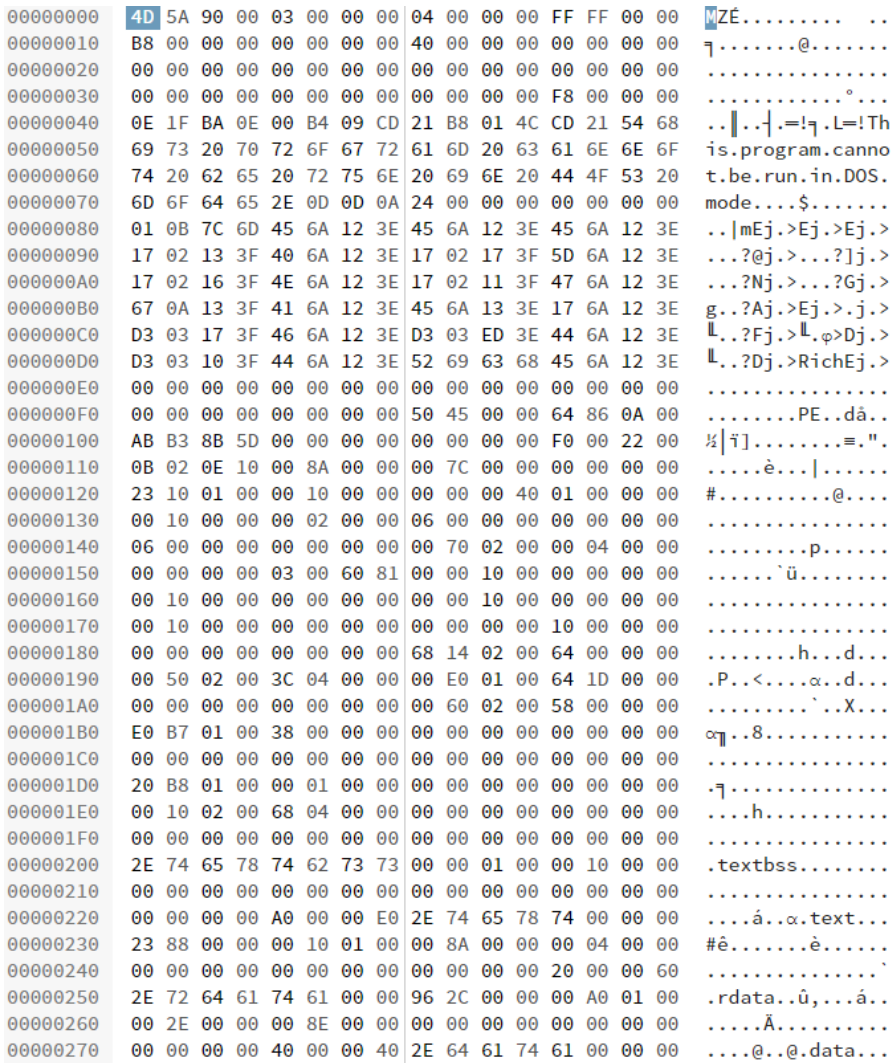
```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!";
    return 0;
}
```

Preprocessing → Compiling → Linking → Executable (compiled for compiled for x86_64 architecture, executed by the operating system)

A small comparison II



```
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZÉ.........  ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ╕.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 F8 00 00 00  ............°...
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..▌..┤.=!╕.L=!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is.program.canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t.be.run.in.DOS.
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  01 0B 7C 6D 45 6A 12 3E 45 6A 12 3E 45 6A 12 3E  ..|mEj.>Ej.>Ej.>
00000090  17 02 13 3F 40 6A 12 3E 17 02 17 3F 5D 6A 12 3E  ...?@j.>...?]j.>
000000A0  17 02 16 3F 4E 6A 12 3E 17 02 11 3F 47 6A 12 3E  ...?Nj.>...?Gj.>
000000B0  67 0A 13 3F 41 6A 12 3E 45 6A 13 3E 17 6A 12 3E  g..?Aj.>Ej.>.j.>
000000C0  D3 03 17 3F 46 6A 12 3E D3 03 ED 3E 44 6A 12 3E  ╙..?Fj.>╙.φ>Dj.>
000000D0  D3 03 10 3F 44 6A 12 3E 52 69 63 68 45 6A 12 3E  ╙..?Dj.>RichEj.>
000000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0  00 00 00 00 00 00 00 00 50 45 00 00 64 86 0A 00  ........PE..då..
00000100  AB B3 8B 5D 00 00 00 00 00 00 00 00 F0 00 22 00  ½│ï]........≡.".
00000110  0B 02 0E 10 00 8A 00 00 00 7C 00 00 00 00 00 00  .....è...|......
00000120  23 10 01 00 00 10 00 00 00 00 00 40 01 00 00 00  #..........@....
00000130  00 10 00 00 00 02 00 00 06 00 00 00 00 00 00 00  ................
00000140  06 00 00 00 00 00 00 00 00 70 02 00 00 04 00 00  .........p......
00000150  00 00 00 00 03 00 60 81 00 00 10 00 00 00 00 00  ......`ü........
00000160  00 10 00 00 00 00 00 00 00 00 10 00 00 00 00 00  ................
00000170  00 10 00 00 00 00 00 00 00 00 00 00 10 00 00 00  ................
00000180  00 00 00 00 00 00 00 00 68 14 02 00 64 00 00 00  ........h...d...
00000190  00 50 02 00 3C 04 00 00 00 E0 01 00 64 1D 00 00  .P..<....α..d...
000001A0  00 00 00 00 00 00 00 00 00 60 02 00 58 00 00 00  .........`..X...
000001B0  E0 B7 01 00 38 00 00 00 00 00 00 00 00 00 00 00  α╖..8...........
000001C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001D0  20 B8 01 00 00 01 00 00 00 00 00 00 00 00 00 00  ·╕..............
000001E0  00 10 02 00 68 04 00 00 00 00 00 00 00 00 00 00  ....h...........
000001F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000200  2E 74 65 78 74 62 73 73 00 00 01 00 00 10 00 00  .textbss........
00000210  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000220  00 00 00 00 A0 00 00 E0 2E 74 65 78 74 00 00 00  ....á..α.text...
00000230  23 88 00 00 00 10 01 00 00 8A 00 00 00 04 00 00  #ê.......è......
00000240  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60  ............ ..`
00000250  2E 72 64 61 74 61 00 00 96 2C 00 00 00 A0 01 00  .rdata..û,...á..
00000260  00 2E 00 00 00 8E 00 00 00 00 00 00 00 00 00 00  .....Ä..........
00000270  00 00 00 00 40 00 00 40 2E 64 61 74 61 00 00 00  ....@..@.data...
```

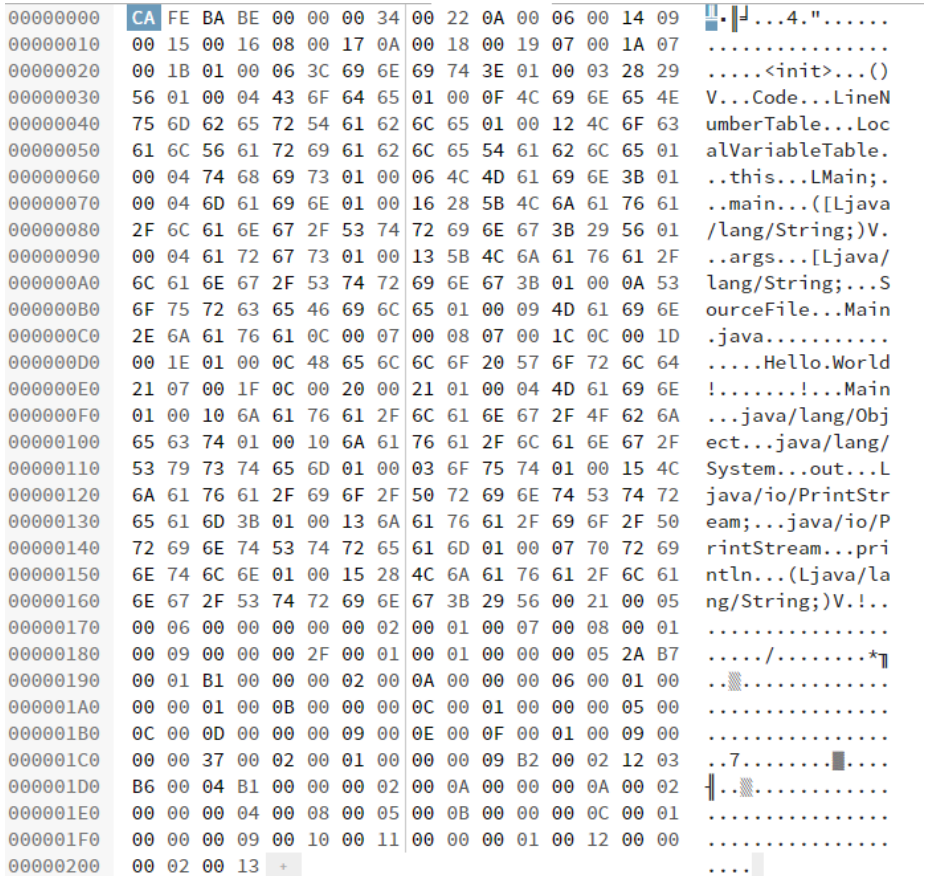
A small comparison III

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Compiling → Byte code (executed by the JVM)

A small comparison IV



```
00000000  CA FE BA BE 00 00 00 34 00 22 0A 00 06 00 14 09  ▒·╣...4."......
00000010  00 15 00 16 08 00 17 0A 00 18 00 19 07 00 1A 07  ................
00000020  00 1B 01 00 06 3C 69 6E 69 74 3E 01 00 03 28 29  .....<init>...()
00000030  56 01 00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E  V...Code...LineN
00000040  75 6D 62 65 72 54 61 62 6C 65 01 00 12 4C 6F 63  umberTable...Loc
00000050  61 6C 56 61 72 69 61 62 6C 65 54 61 62 6C 65 01  alVariableTable.
00000060  00 04 74 68 69 73 01 00 06 4C 4D 61 69 6E 3B 01  ..this...LMain;.
00000070  00 04 6D 61 69 6E 01 00 16 28 5B 4C 6A 61 76 61  ..main...([Ljava
00000080  2F 6C 61 6E 67 2F 53 74 72 69 6E 67 3B 29 56 01  /lang/String;)V.
00000090  00 04 61 72 67 73 01 00 13 5B 4C 6A 61 76 61 2F  ..args...[Ljava/
000000A0  6C 61 6E 67 2F 53 74 72 69 6E 67 3B 01 00 0A 53  lang/String;...S
000000B0  6F 75 72 63 65 46 69 6C 65 01 00 09 4D 61 69 6E  ourceFile...Main
000000C0  2E 6A 61 76 61 0C 00 07 00 08 07 00 1C 0C 00 1D  .java...........
000000D0  00 1E 01 00 0C 48 65 6C 6C 6F 20 57 6F 72 6C 64  .....Hello.World
000000E0  21 07 00 1F 0C 00 20 00 21 01 00 04 4D 61 69 6E  !.......!...Main
000000F0  01 00 10 6A 61 76 61 2F 6C 61 6E 67 2F 4F 62 6A  ...java/lang/Obj
00000100  65 63 74 01 00 10 6A 61 76 61 2F 6C 61 6E 67 2F  ect...java/lang/
00000110  53 79 73 74 65 6D 01 00 03 6F 75 74 01 00 15 4C  System...out...L
00000120  6A 61 76 61 2F 69 6F 2F 50 72 69 6E 74 53 74 72  java/io/PrintStr
00000130  65 61 6D 3B 01 00 13 6A 61 76 61 2F 69 6F 2F 50  eam;...java/io/P
00000140  72 69 6E 74 53 74 72 65 61 6D 01 00 07 70 72 69  rintStream...pri
00000150  6E 74 6C 6E 01 00 15 28 4C 6A 61 76 61 2F 6C 61  ntln...(Ljava/la
00000160  6E 67 2F 53 74 72 69 6E 67 3B 29 56 00 21 00 05  ng/String;)V.!..
00000170  00 06 00 00 00 00 00 02 00 01 00 07 00 08 00 01  ................
00000180  00 09 00 00 00 2F 00 01 00 01 00 00 00 05 2A B7  ...../........*╖
00000190  00 01 B1 00 00 00 02 00 0A 00 00 00 06 00 01 00  ..▒.............
000001A0  00 00 01 00 0B 00 00 00 0C 00 01 00 00 00 05 00  ................
000001B0  0C 00 0D 00 00 00 09 00 0E 00 0F 00 01 00 09 00  ................
000001C0  00 00 37 00 02 00 01 00 00 00 09 B2 00 02 12 03  ..7........▒....
000001D0  B6 00 04 B1 00 00 00 02 00 0A 00 00 00 0A 00 02  ╢..▒............
000001E0  00 00 00 04 00 08 00 05 00 0B 00 00 00 0C 00 01  ................
000001F0  00 00 00 09 00 10 00 11 00 00 00 01 00 12 00 00  ................
00000200  00 02 00 13                                      ....
```

Java's Magic Word

- Java magic word: CAFE BABE (see previous slide)
- James Gosling: "We used to go to lunch at a place called St Michaels Alley. According to local legend, in the deep dark past, the Grateful Dead used to perform there before they made it big. It was a pretty funky place that was definitely a Grateful Dead Kinda Place. [...] When we used to go there, we referred to the place as Cafe Dead. Somewhere along the line, it was noticed that this was a HEX number. I was re-vamping some file format code and needed a couple of magic numbers: one for the persistent object file, and one for classes. I used CAFEDEAD for the object file format, and in grepping for 4 character hex words that fit after CAFE (it seemed to be a good theme) I hit on BABE and decided to use it. [...] So CAFEBABE became the class file format. At that time, it didn't seem terribly important or destined to go anywhere but the trash can of history."

Installation and tools

- We are going to use Java Standard Edition 21 (LTS). Java SE 21 (LTS) offers Long Term Support.
- Download JDK 21 from Oracle.
- To verify, open **command.com** and type **java -version**.
- The examples in this lecture are going to use the IntelliJ IDEA.
- To have access the premium versions of JetBrains product, you may use your UBB student account here: JetBrains Products for Learning.
- We will use GitHub for our laboratories. Please make sure to create a GitHub account.

Java Syntax

- Derived from C and C++, very similar to the syntax of these languages.
- Key words (e.g. case, for, if, return, void, throws, public, class, boolean, break ... and may others)
- Separators: ( ) { } ; , .
- Literals: "Hello", 'a', 100, 12.3, true, false, null
- Operators: *, -, ++, ¡, !=, ——, += ... and others)
- Comments:
  - Single line: //
  - Multi line: /* ... */
  - Javadoc: /** ... */

Data types and references

- Primitive data types: byte, short, int, long, float, double, boolean, and char.
- void - not a data type, a cast to void is not allowed.
- **References**:
  - When an object is created, we get a *reference* to it. We cannot get hold of the actual object.
  - References and primitives are passed *by value*.
  - Objects are not passed as function parameters, references are.
- null - keyword that represents a null reference.

Java references vs C++ pointers and C++ references

	Java references	C++ pointers	C++ references
Points at objects	Yes	Yes	Yes
Initialized using new	Yes	Yes	No
Can be null	Yes	Yes	No
Can be updated to point to a different object	Yes	Yes	No
Passed by value	Yes	Yes	No
Is implicitly dereferenced	Yes	No	Yes

Variables, constants, instructions

- Variables:

```
int x = 10;
String animal = "raccoon";
```

- Constants: final.

```
final int MAX = 32000;
final int MIN;
MIN = -32000;
MIN = 200; // ERROR
```

- Instructions: if, while, do-while, for, switch.

Arrays I

- Declaration:

```
type name[]; OR type[] name;
```

- Initialisation:

```
name = new type[DIM];
```

- All arrays in Java are dynamically allocated.
- Indexing begins from 0.
- An array's capacity is given by length.

Arrays II

- N-dimensional array declaration:

```
type name[][]...[]; OR type[]...[] name;
```

- Initialisation:

```
name = new type[DIM_1][DIM_2]...[DIM_n];
```

Strings

- Arrays of characters, e.g.

```
char[] array = {'a', 'b', 'c'};
```

- Accessing: index-based.
- String class:

```
String s = "abc";
s += "def";
```

- String comparison: equals function:

```
String s = new String("abc");
String t = new String("abc");
System.out.println(s.equals(t));
```

Java quick start

- In Java we need to create a class for any application.
- The name of the file must be identical to the class name.
- A main() method is required.
- Arguments can be passed in the command line when executing a Java program.

Example

Introduction in Java.

Assignment

Write a Java program that generates all the prime numbers smaller than a number given as a command line parameter.

Summary

- The software development life cycle is complex, including several phases.
- There are many software development methodologies, belonging mostly to 2 categories: traditional and agile.
- When designing a software solution, it is important to consider the software design principles.
- The main programming language for this course is Java.
- *Next week*:
  - Object oriented programming in Java.