


OBJECT-ORIENTED PROGRAMMING
1.1 INTRODUCTION TO OBJECT TECHNOLOGY

Nguyen Thi Thu Trang
trangntt@soict.hust.edu.vn



1056 1 1

1

2 2

Outline

1. Programming Paradigm
2. Object Technology
3. Java
4. Exercises

1056 1 1

2

3 3

Programming paradigm

- Machine code
 - Assembly language
- Procedural programming languages
 - COBOL, FORTRAN, BASIC, Pascal, C
- Object programming languages
 - C++, Java, C#.NET, Python

1056 1 1

3

4 4

1.1. Machine Code

```
;CLEAR SCREEN USING BIOS
CLR: MOV AX,0600H    ;SCROLL SCREEN
      MOV BH,30      ;COLOUR
      MOV CX,0000     ;FROM
      MOV DX,104FH    ;TO 20,79
      INT 10H         ;CALL BIOS;
;INPUTTING OF A STRING
KEY: MOV AH,0AH      ;INPUT REQUEST
      LEA DX,BUFFER   ;POINT TO BUFFER WHERE STRING STORED
      INT 21H         ;CALL DOS
      RET             ;RETURN FROM SUBROUTINE TO MAIN PROGRAM;
; DISPLAY STRING TO SCREEN
SCR: MOV AH,09        ;DISPLAY REQUEST
      LEA DX,STRING   ;POINT TO STRING
      INT 21H         ;CALL DOS
      RET             ;RETURN FROM THIS SUBROUTINE;
```

Assembly code

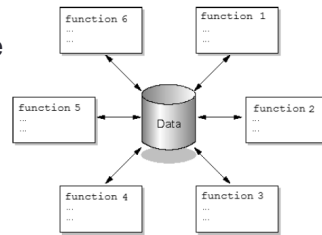
- Assembly language
 - Is a sequence programming language, is very close to machine codes of CPU.
 - Hard to remember, to write, especially for complex systems.
 - Hard to fix, to maintain.

1056 1 1

4

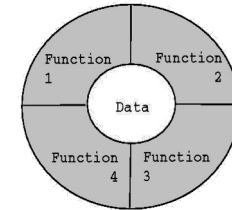
1.2. Procedural Languages

- Procedural or structural programming focuses mainly on writing software based on functions
 - Data and functions that manipulate data are kept separate
- Procedural programs cannot easily enforce a common way to access data



1.3. Object-Oriented Programming

- Programming implementation technique based on objects
- Software engineering best practice
 - The use of objects allow systems to become extensible, scalable, maintainable and adaptable
- Closely represent objects of the real world, keeping the programmer in touch with the problem

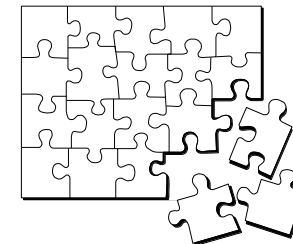


Outline

1. Programming Paradigm
- ➔ 2. Object Technology
3. Java
4. Exercises

Object Technology

- Object technology is a set of rules (e.g. abstraction, encapsulation, polymorphism), instructions to build a software, together with languages, databases and other tools to support these rules.



(Object Technology - A Manager's Guide, Taylor, 1997)

The Strengths of Object Technology

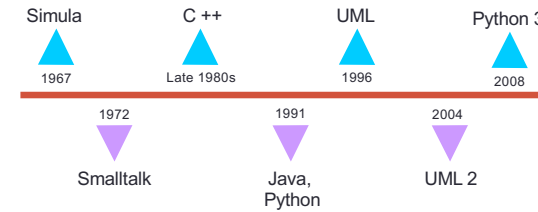
- Reflects a single paradigm
- Facilitates architectural and code reuse
- Reflects real world models more closely
- Encourages stability
- Is adaptive to change

105618

9

The History of Object Technology

- Major object technology milestones

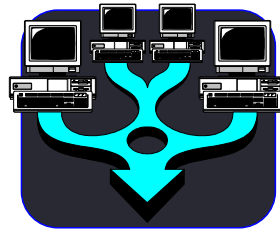


105618

10

Where Is Object Technology Used?

- Client/Server Systems and Web Development
 - Object technology allows companies to encapsulate business information in objects and helps to distribute processing across the Internet or a network.



105618

11

Where Is Object Technology Used? (2)

- Real-time systems
 - Object technology enables real-time systems to be developed with higher quality and flexibility.




105618

12

13

Object-Oriented Programming

- **Object** is the key to understand the object technology
- In a OO system, all are objects

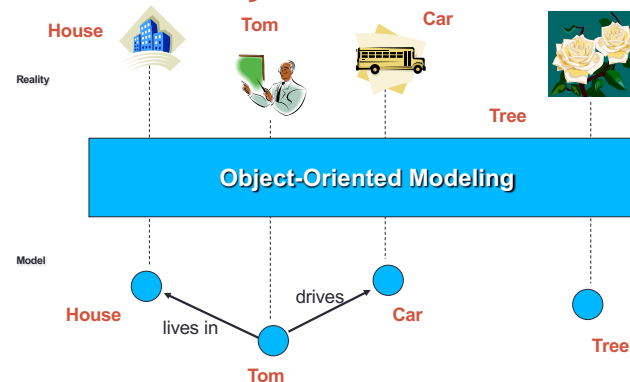


Writing a OO program means to build a model of some parts in the real world

13

14 14

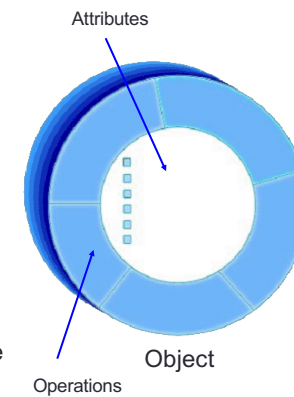
What is an Object?



14

What is an Object?

- An object is an entity with a well-defined boundary and identity that encapsulates state and behavior.
 - State is represented by attributes and relationships.
 - Behavior is represented by operations, methods, and state machines.




15

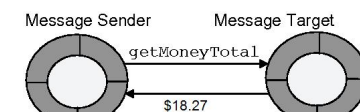
16 16

Interactions between objects

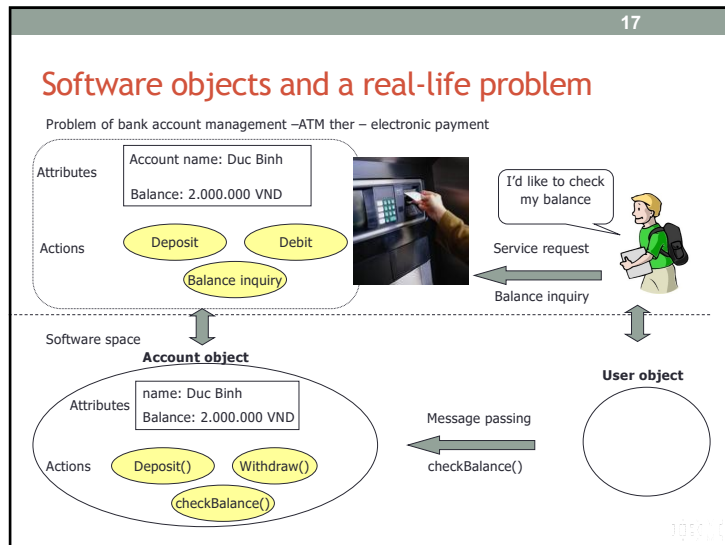
- Communication between objects in the real world



- Objects and their interactions in programming
 - Objects communicate to each other by message passing



16



17

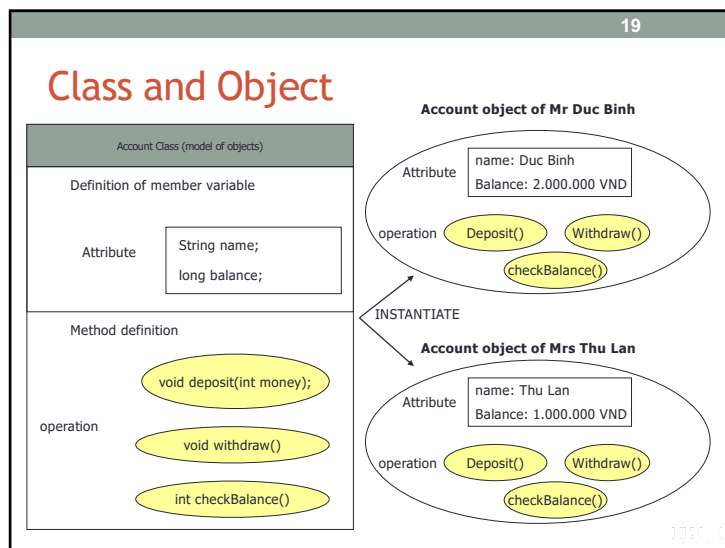
What Is a Class?

- A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics.
 - An object is an instance of a class.
- A class is an abstraction in that it
 - Emphasizes relevant characteristics.
 - Suppresses other characteristics.

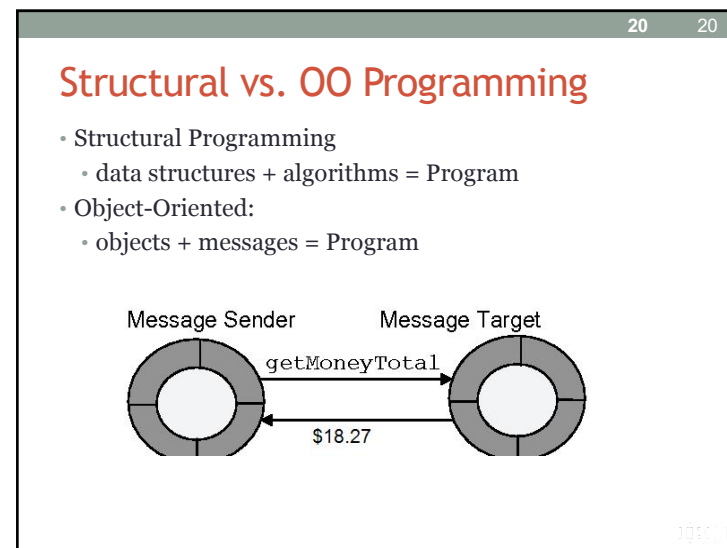
Class

1056/16

18



19



20

21

Procedural vs. Object-Oriented Programming

- Procedural Programming:
 - Main components are procedures, functions
 - Data is independent with procedures
- Object-Oriented Programming
 - Main components are objects
 - Data is associated to function (method) in an object
 - Each data structure has methods executing on it

1056 1 0

21

22 22

Outline

1. Programming Paradigm
2. Object Technology
- ➔ 3. Java
4. Exercises

1056 1 0

22

23 23

3.1 What is Java?

- Java is a object-oriented programming language developped by Sun Microsystems, and now bought by Oracle
- Java is a popular programming language
 - Initially used for building control processor applications inside the electronics consumer devices such as cell phones, microwaves ...
 - Initially used in 1995



Green Team and James Gosling (the leader)



1056 1 0

23

24 24

Java SE (Java Platform Standard Edition)

- <https://www.oracle.com/java/technologies/java-se-glance.html>
- Former name: J2SE
- Develop and deploy Java applications on desktops and servers

1056 1 0

24

25

JRE and JDK

- Java SE Runtime Environment (JRE)
 - Executable Environment or JRE provides Java APIs, Java Virtual Machine (JVM) and other necessary components to run applets and applications written in Java.
- Java SE Development Kit (JDK)
 - Super set of JRE, and contains everything in the JRE, additional tools such as compilers and the debugger need to develop applets and applications.

100%

25

26 26

Java EE (Java Platform Enterprise Edition)

- <https://www.oracle.com/java/technologies/java-ee-glance.html>
- Former name: J2EE. New name (from 2020)
- Extending Java SE with specifications for enterprise features such as distributed computing and web services
 - Web Applications: Servlet/JSP, JSF...
 - Enterprise Applications: EJB, JavaMail...

100%

26

27

Java ME (Java Platform Micro Edition)

- <https://www.oracle.com/java/technologies/javame-overview.html>
- Applications running on embedded and mobile devices in the Internet of Things
 - Micro-controllers, sensors, gateways,
 - Mobile phones, personal digital assistants (PDAs),
 - TV set-top boxes, printers

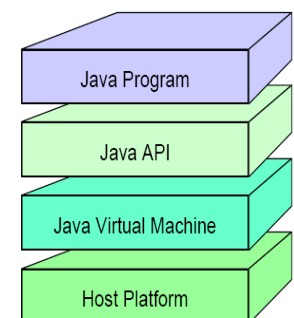
100%

27

28 28

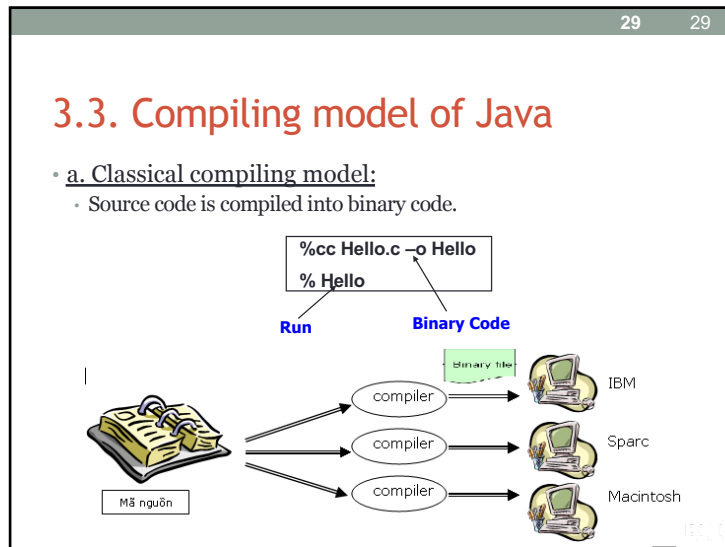
3.2 Java platform

- Platform is environment for development of deployment.
- Java platform can be run on all OSs
 - Other platforms depend on hardware
 - Java platform provides:
 - Java Virtual Machine (JVM).
 - Application Programming Interface (API).

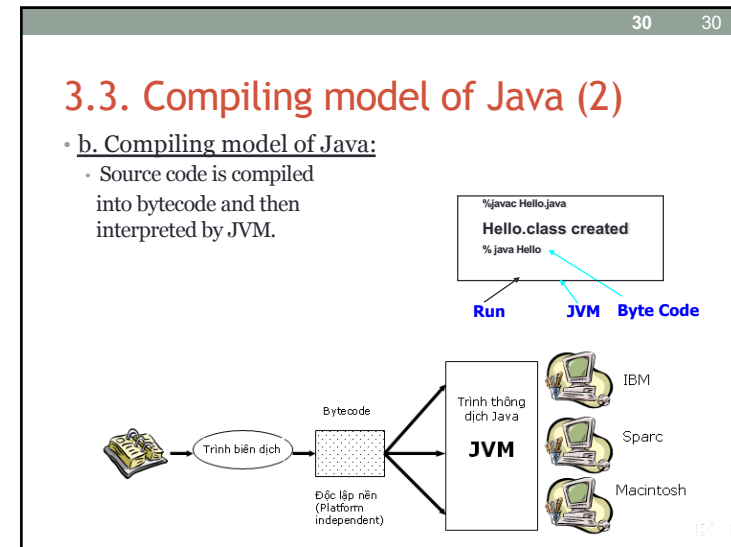


100%

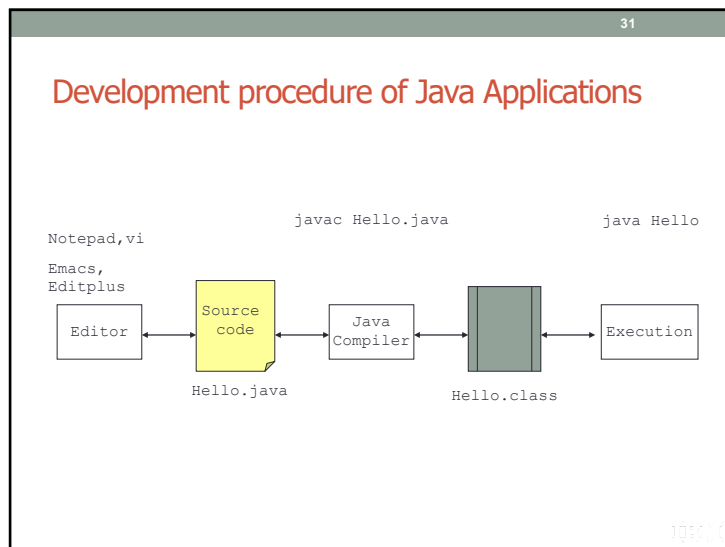
28



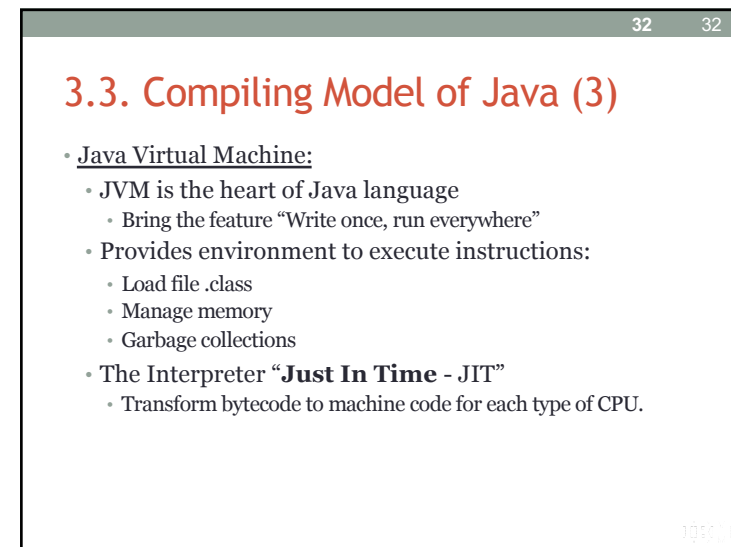
29



30



31



32

33 33

3.4. Features of Java

- Java is designed to be:
 - A powerful programming language, full of OO features and completely OO.
 - Easy to learn, syntax is similar to C++
 - Platform independence
 - Support the development of applications in network environment
 - Ideal for Web application

Navigation icons

33

34 34

3.4. Features of Java (2)

- Powerful
 - Class library: Hundreds of classes already written with utility functions.
 - Java uses pointer model without accessing directly to the memory; memory can not be over-written.
- Object-Oriented
 - Java supports software development by using OO
 - Software built in Java includes classes and objects

Navigation icons

34

35 35

3.4. Features of Java (3)

- Simple
 - Keywords
 - Java has 50 keywords
 - Compared to Cobol VB that have hundreds of keywords
- Network capable
 - Java supports the development of distributed applications
 - Some applications of Java are designed in order to be accessed via Web browser.

Navigation icons

35

36 36

3.4. Features of Java (3)

- Java has 50 key words
 - assert (New in 1.5) enum (New in 1.5)

abstract	boolean	break	byte
case	catch	char	class
const	continue	default	do
double	else	extends	final
finally	float	for	goto
if	implements	import	instanceof
int	interface	long	native
new	package	private	protected
public	return	short	static
strictfp	super	switch	synchronized
this	throw	throws	transient
try	void	volatile	while

Navigation icons

36

37 37

3.4. Features of Java (5)

- Multi-threaded
 - Allows a program to run more than one task at the same time.
- Portable
 - Programs can be written once and run on different platforms
 - Based on compiler/interpreter model (WORE – Write Once, Run Everywhere)

1056 1056

37

38 38

3.4. Features of Java (6)

- Development Environment
 - Java Development Kit
 - Free on Sun Website: java.sun.com
 - Including: Compiler, JVM and existing classes
 - Integrated Development Environments (IDEs): Providing:
 - Complex Text Editors
 - Debugging Tools
 - Graphics Development Tools

1056 1056

38

39 39

3.5. Applications in Java

- Application
 - Do not need to run on browsers
 - Can call functions through commands or option menu (GUI)
 - main() method is the starting point of the program execution
- Applet
 - GUI application running on browser in the client side.
 - Can be viewed by appletviewer or embedded in Web browser with JVM installed.

1056 1056

39

40 40

3.5. Applications in Java (2)

- Web application
 - Create dynamic content on Server instead of on browsers.
 - Used in Server application
 - Servlet: manage requests from browsers and send the responses back
 - JavaServer Page (JSP): HTML pages with embedded Java code.

1056 1056

40

41 41

Outline

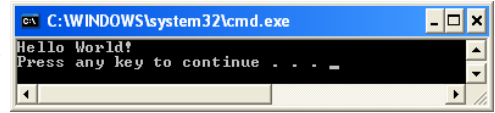
1. Object-Oriented Technology
2. Object and Class
3. Java programming languages
- 4. Examples and Exercises

41

42 42

Example 1 - HelloWorld

```
// HelloWorld.java
public class HelloWorld {
    /* The main() method will be called first in any
    Java applications */
    public static void main(String args[]){
        System.out.println( "Hello World!" );
    }
}
```



42

43 43

Example 1 (Cont.)

- Comment
 - In one line: Starts with //
 - In multiple lines: /* ... */
- Java distinguish between lowercase and uppercase
- Keywords in Java:
 - class: Class definition
 - public: Access permission
- Class name containing main function must have the same name with the file .java.

43

44 44

Installing and Running Java application

- Step 1: Install jdk, install environment variables (if using cmd)
- Step 2: Install Eclipse or Netbean IDE
- Step 3: Coding
- Step 4: Compile
 - cmd: javac HelloWorld.java
 - Eclipse/Netbean: Build automatically (Look at Console to see syntax errors if any)/F11 (Project) or F9 (File)
- Step 5: Run program
 - cmd: java HelloWorld
 - Eclipse/Netbean: Run as Java application (Alt+Shift+X+J)/F6 (Project) or Shift-F6 (File)

44

45 45

Environment Variables

- PATH = %PATH%;C:\Program Files\Java\jdkx.x\bin
- JAVA_HOME=C:\Program Files\Java\jdkx.x
- CLASSPATH = C:\Program Files\Java\jdkx.x\lib;.;C:\Program Files\Java\jdkx.x\include

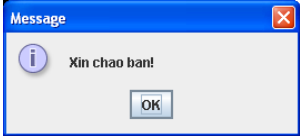
1056 / 8

45

46 46

Example 2 - GUI

```
import javax.swing.JOptionPane;
public class FirstDialog{
    public static void main(String[] args){
        JOptionPane.showMessageDialog(null,
            "Xin chao ban!");
        System.exit(0);
    }
}
```



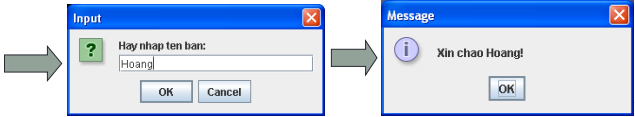
1056 / 8

46

47 47

Example 3 - Data Input/Output

```
import javax.swing.JOptionPane;
public class HelloNameDialog{
    public static void main(String[] args){
        String result;
        result = JOptionPane.showInputDialog("Hay nhap ten ban:");
        JOptionPane.showMessageDialog(null,
            "Xin chao " + result + "!");
        System.exit(0);
    }
}
```



1056 / 8

47