Féidearthachtaí as Cuimse
Infinite Possibilities

Java : Parts and Set up

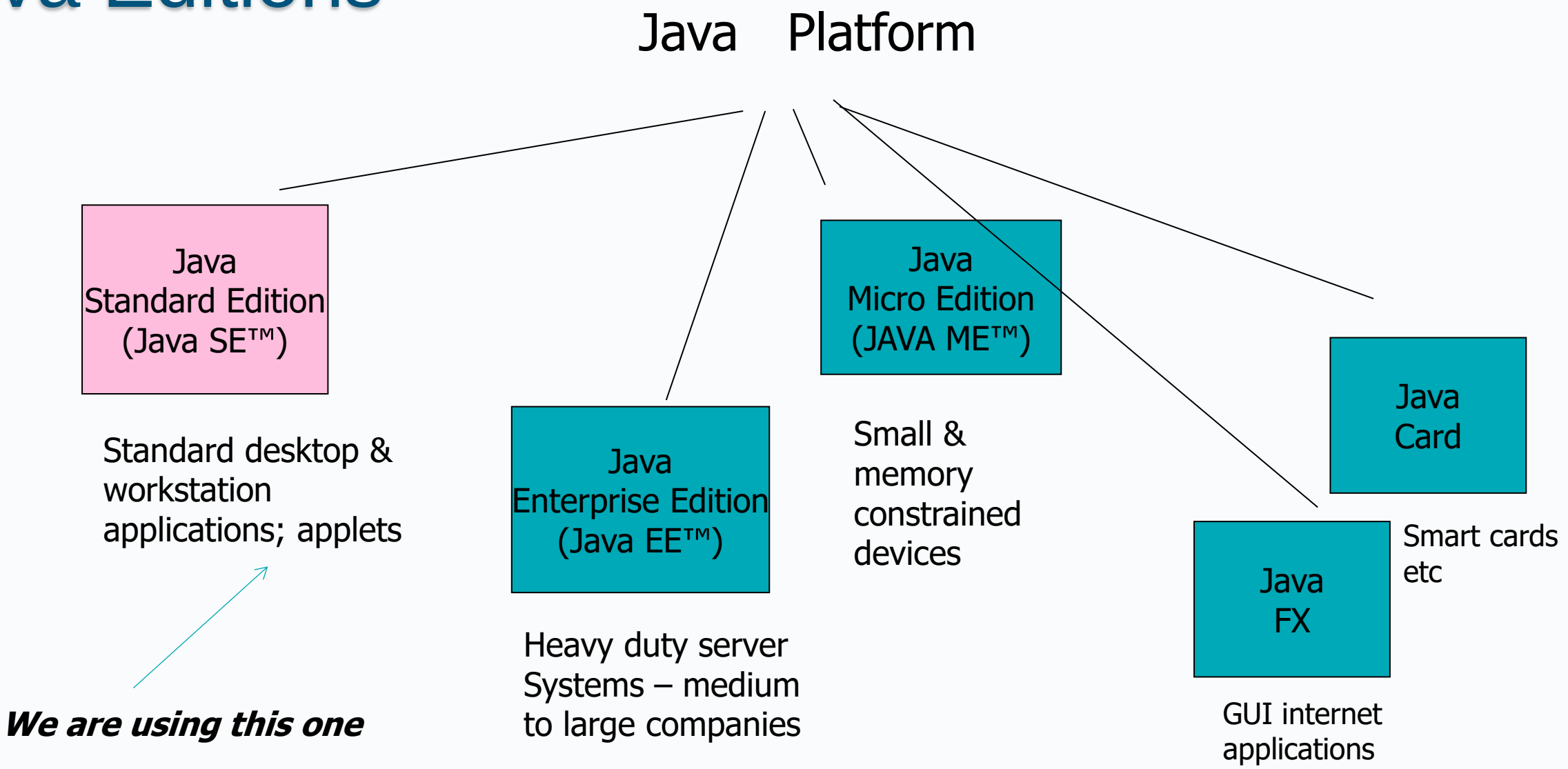
Object Oriented programming

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Java Overview

- JavaAPI
- IDEs
- Packages

Java Editions



Why use Java/Benefits

- Java is “write once, run anywhere”
  - architecture neutral
  - portable across different platforms
  - Due to Java Virtual Machine (JVM)

- Security features
  - highly configurable security levels prevent any piece of Java code doing harm to the host system

Why use Java/Benefits (2)

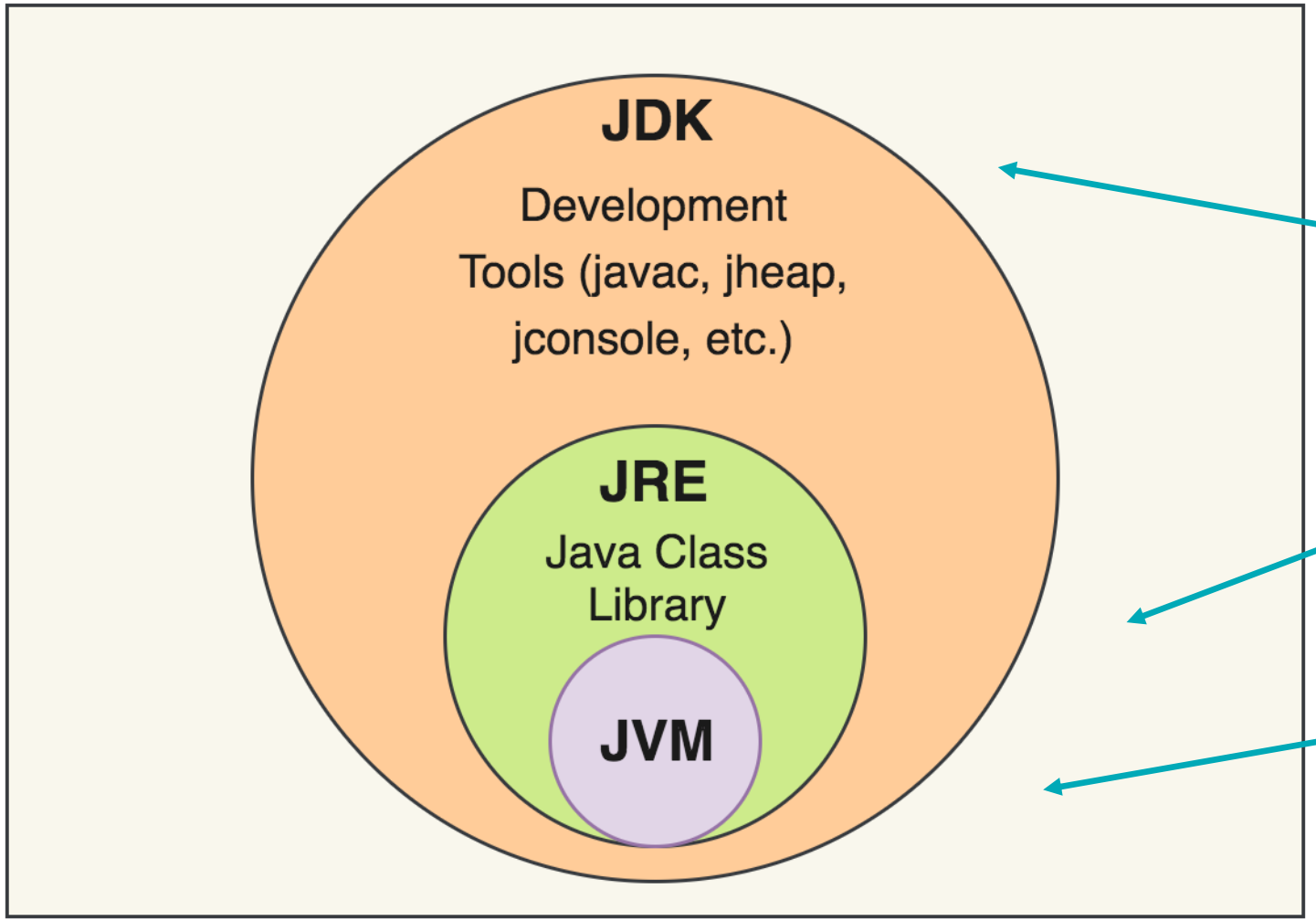
- Network-centric platform
  - easy to work with resources across a network and to create network based applications
- Object Oriented
  - an interacting collection of independent software components
  - dynamic extensible programs

Why use Java/Benefits (3)

- Internationalisation
  - uses 16 bit Unicode characters that represents the phonetic and ideographic character sets of the entire world
- Performance
  - although an interpreted language Java programs run almost as fast as native C, C++ programs

- **Simple and easy to develop**
  - **powerful & well designed set of APIs**
  - https://docs.oracle.com/en/java/javase/23/docs/api/index.html

Different Parts to Java



Java Development Kit
(may be part of your
IDE e.g. Eclipse)

Java Run
time
Environment

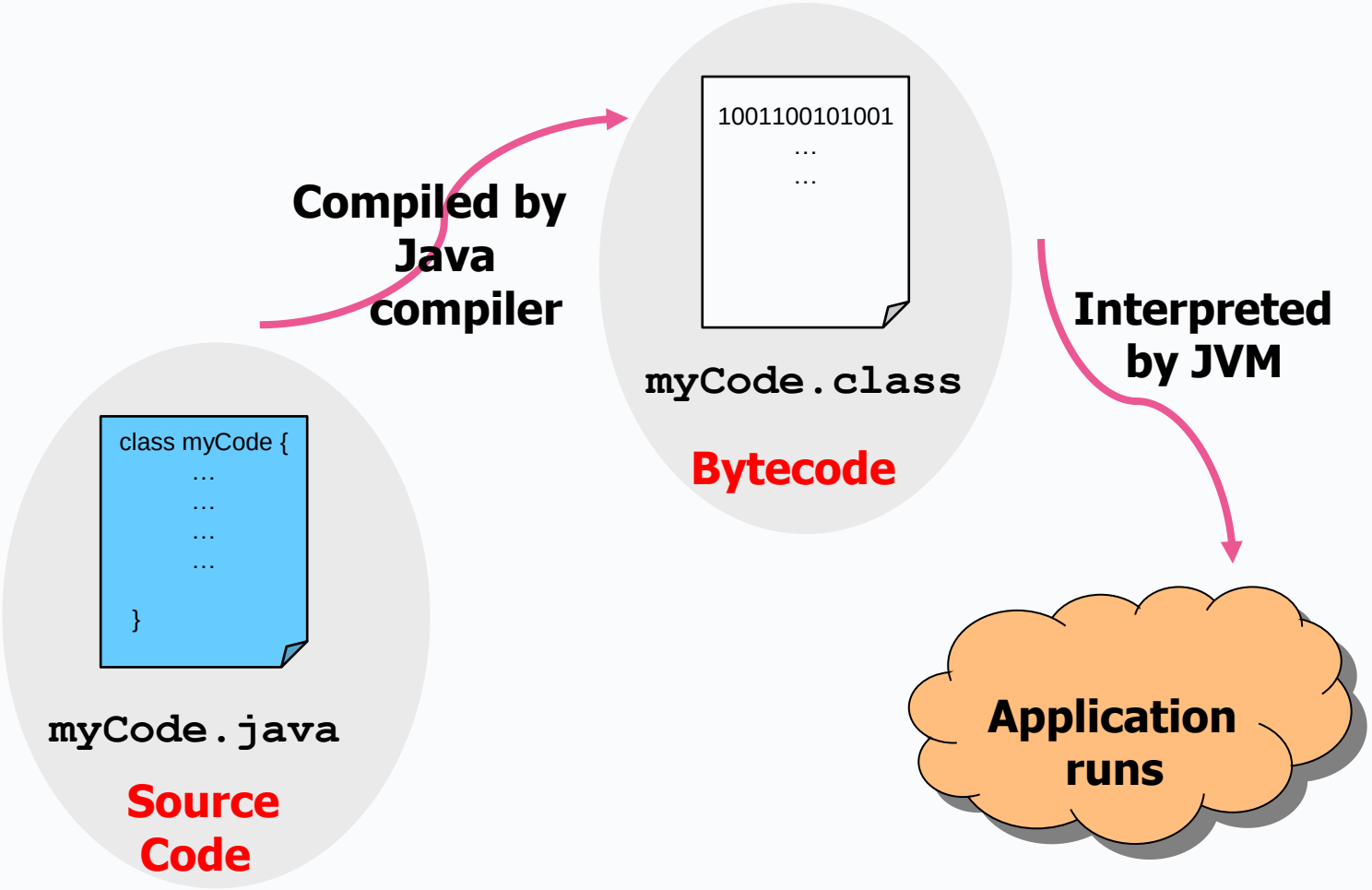
Java Virtual
Machine
*(Write once
Run everywhere)*

JVM (Java Virtual Machine)

- JVM enables the Write-one-Run-Everywhere of java
- **Platform Independence**: Java source code is compiled into **bytecode** (platform independent) rather than machine (i.e. native) code. Bytecode can be executed on any platform that has a compatible JVM (JVM converts bytecode into native machine code for the platform)

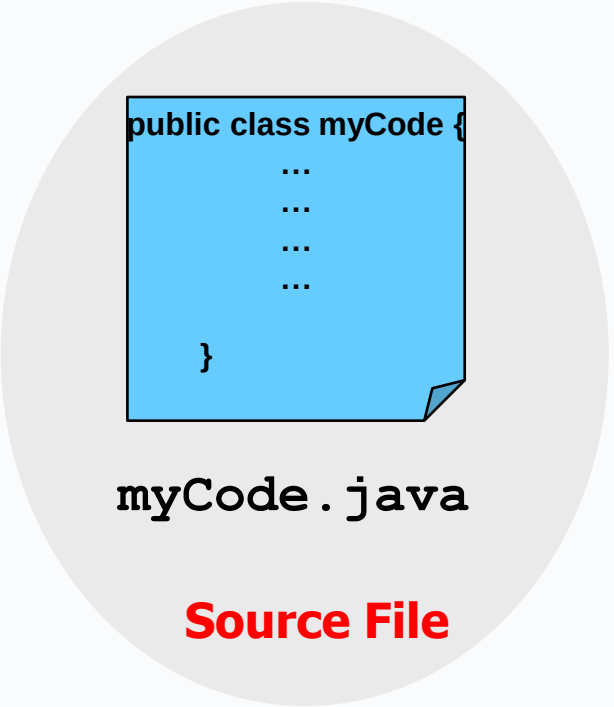
  e.g.
  •write a Java program on Windows, compile it to bytecode.
  •Then run the bytecode on Linux, Ma (or any other operating system that has a JVM installed)

Java Virtual Machine (JVM)



Using the JDK

- Each class is stored in a source file “xxx.java”
- The name of .java source file must be the exact same as the name of the class in our code



Integrated Development Environments

- Need one to support editing, compiling, API. package mgt
- Loads of choice e.g.
  - Eclipse
  - IntellJ
  - JDoodle https://www.jdoodle.com/online-java-compiler/

For ref: Setting up Java and Eclipse

- For your own use
  - Download Eclipse (or any IDE you prefer)
    - https://www.eclipse.org/downloads/
  - Bigger IDEs have JDK installed (e.g. Eclipse, IntelliJ) – easy !!!
  - IF you use a separate IDE/JDK may need to know where both the JDK AND the compiled classes are:
    - E.g. Set up a **java_home** environment variable if using Windows to point to jdk home directory;

Java API (application programming interface)

- Oracle provide a whole set of classes are there for you to use called “the java api”
  - https://docs.oracle.com/en/java/javase/23/docs/api/index.html
- **Constantly needed!**

- Java API is organised into
  - Modules – which consist of:
    - Packages - which consist of :
      - Classes/ interfaces

Packages

- Organises classes into groups (otherwise, big mess)
- Every class must be in a package *Bad practice to have classes not in a package*

- Packages
  - Groups
  - Access
  - Create your own
  - e.g. `Package com.test.lab1`
  - Naming convention: reverse domain name
    - (& lowercase)

Using packages

- To use someone else's (e.g. Java API) in your code:

  import

  - `import javax.swing.*;`
  - `import javax.swing.JButton;`
    - Only import what you need..
    -
  - Eclipse (or whatever IDE) helps

Modules in the java api-

- API for Java SE version 9 onwards
- Higher level organising than packages

  https://docs.oracle.com/en/java/javase/13/docs/api/index.html

- More later

- For now – choose NOT to create a module-info for your projects

Covered

- Java editions
- Various components of java
- Advantages of using Java
- Java install/ setup
- Java API
- Packages in java