INTRODUCTION TO JAVA PROGRAMMING LANGUAGE

David Corral Plaza

Universidad de Cádiz

Puerto Real, July 2018

CONTENTS

CONTENTS

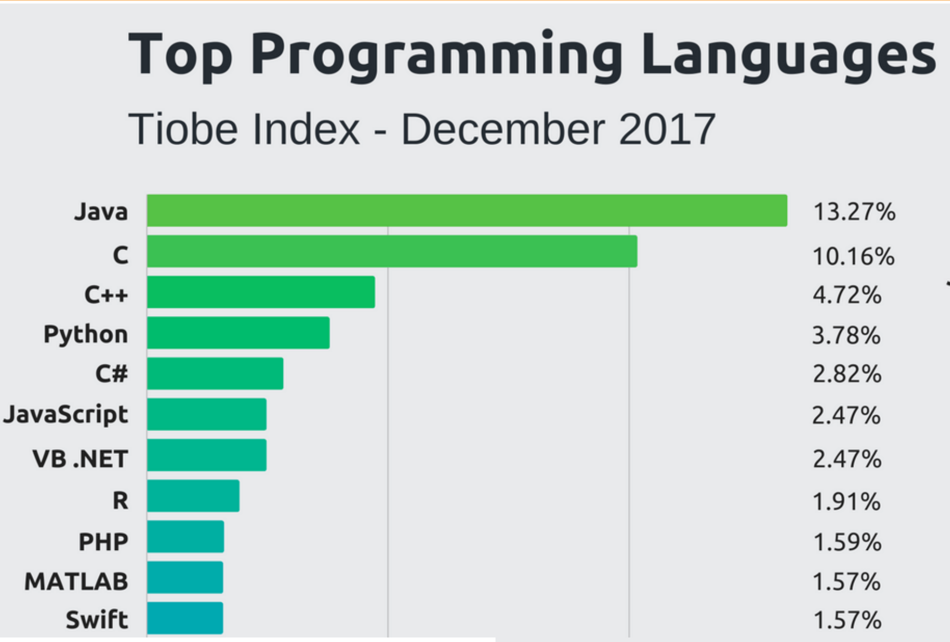
WHAT IS JAVA?

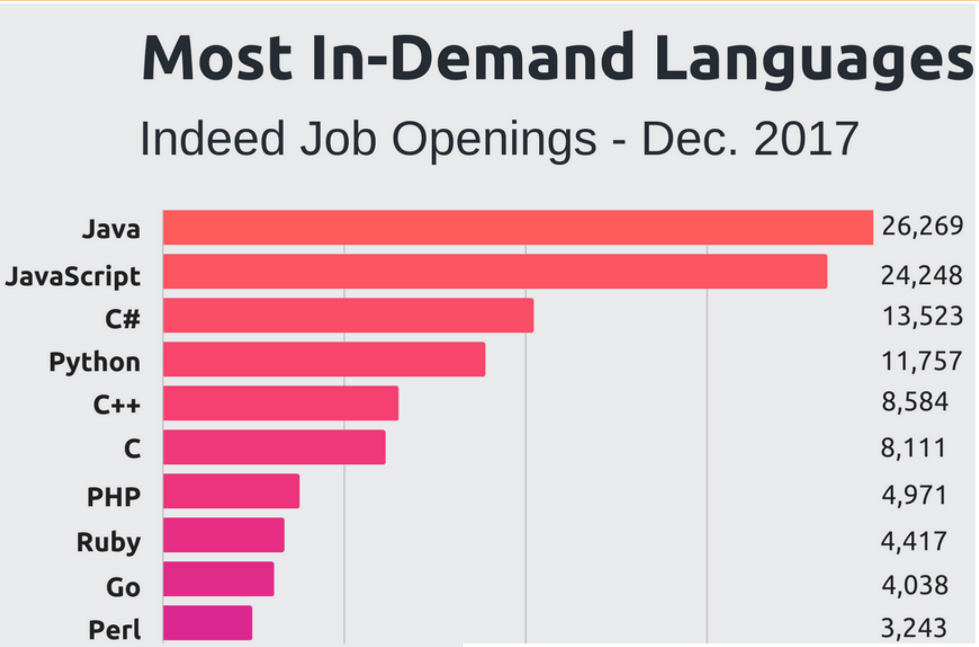
- Java is an **OOP** (Object-Oriented Programming) language: these objects have a unique identity, attributes and operations, and are instances of classes.

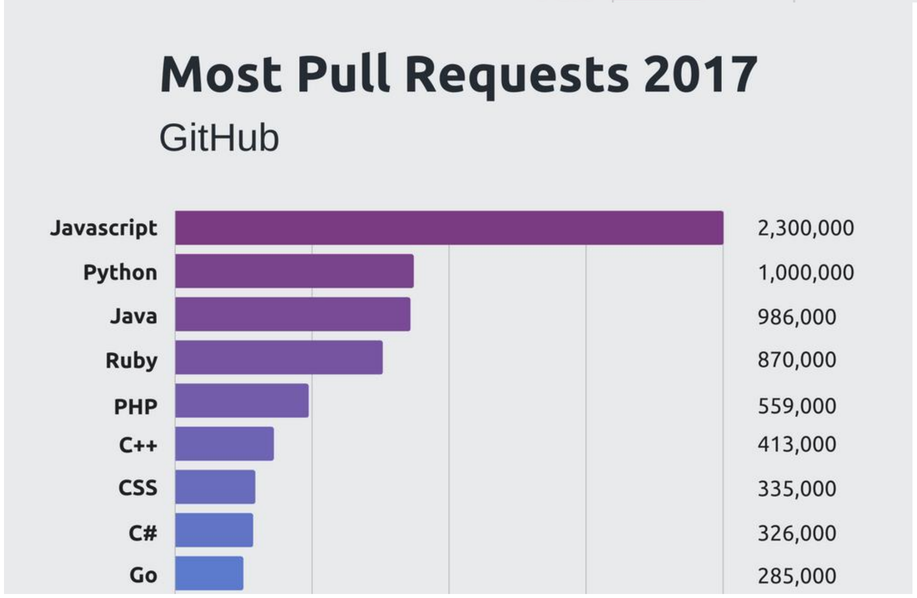
- Java is **interpreted**: is compiled to bytecodes, which are interpreted by a Java run-time environment.

- Java is **multithreaded**: we can perform several tasks at the same time.

JAVA: THE MOST WANTED



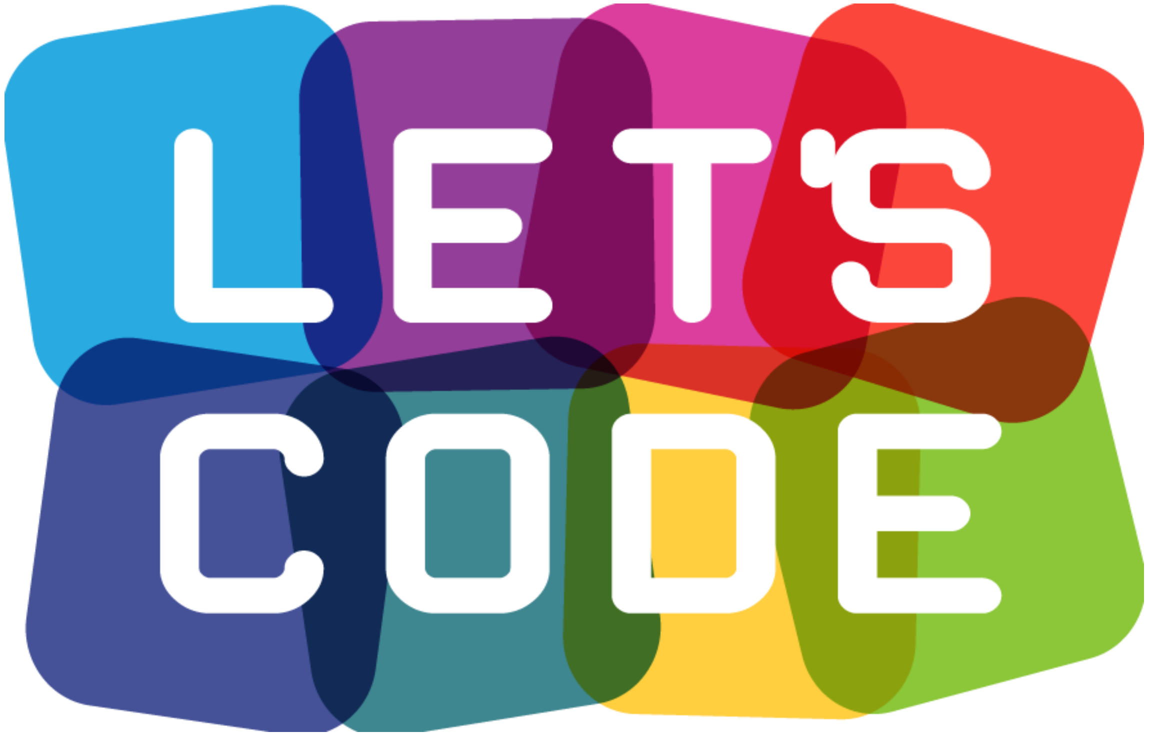




HOW TO

- Install Java Development Kit >= 8
- An IDE (Interface Development Environment)
  - **IntelliJ Ultimate Edition -** https://goo.gl/cvShpa
  - Eclipse
  - NetBeans

LET’S CODE: HelloWorld.java

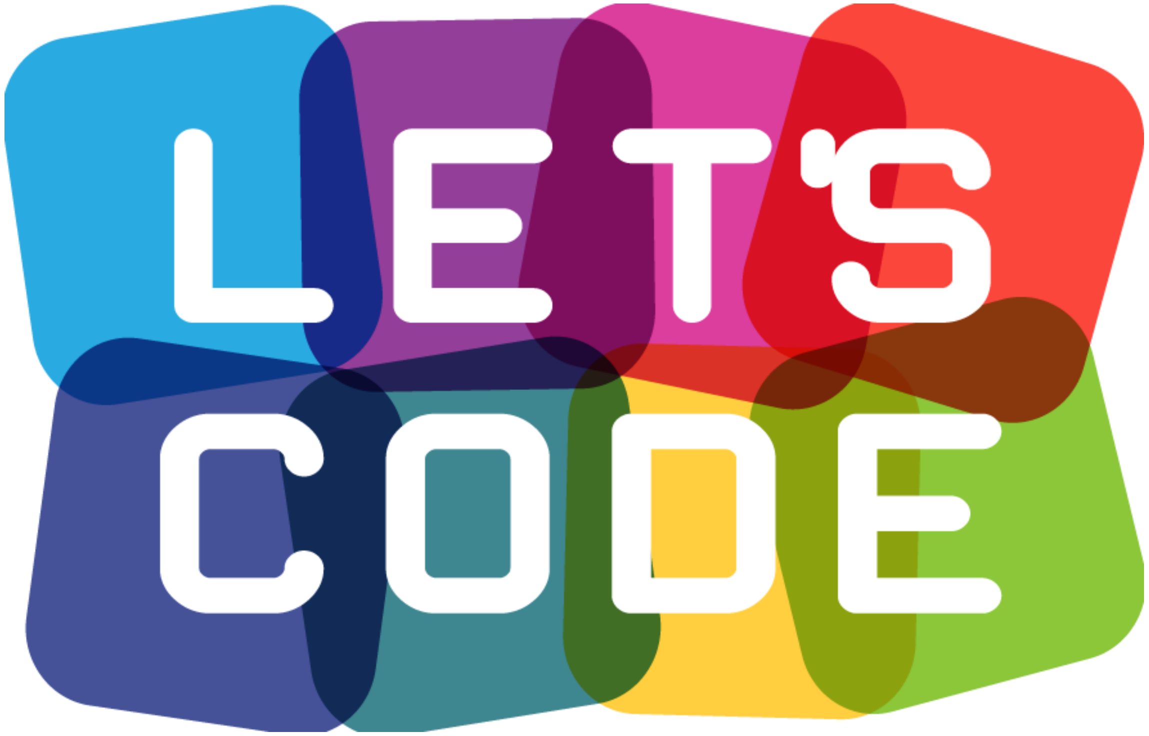


CONTENTS

VARIABLES

- They have types:
  - Numeric: int, float, double, long
  - Alphanumeric: char, string
  - True/False: boolean
- They can be local or global
- They can be collections of types, i.e.: arrays
- They can be constants
- They can be private, public and protected

LET’S CODE: Variables.java



CONTENTS

CONTROL FLOW

- They decide the way of our data in our code:
  - If-Else
  - Switch
  - While / do while
  - For

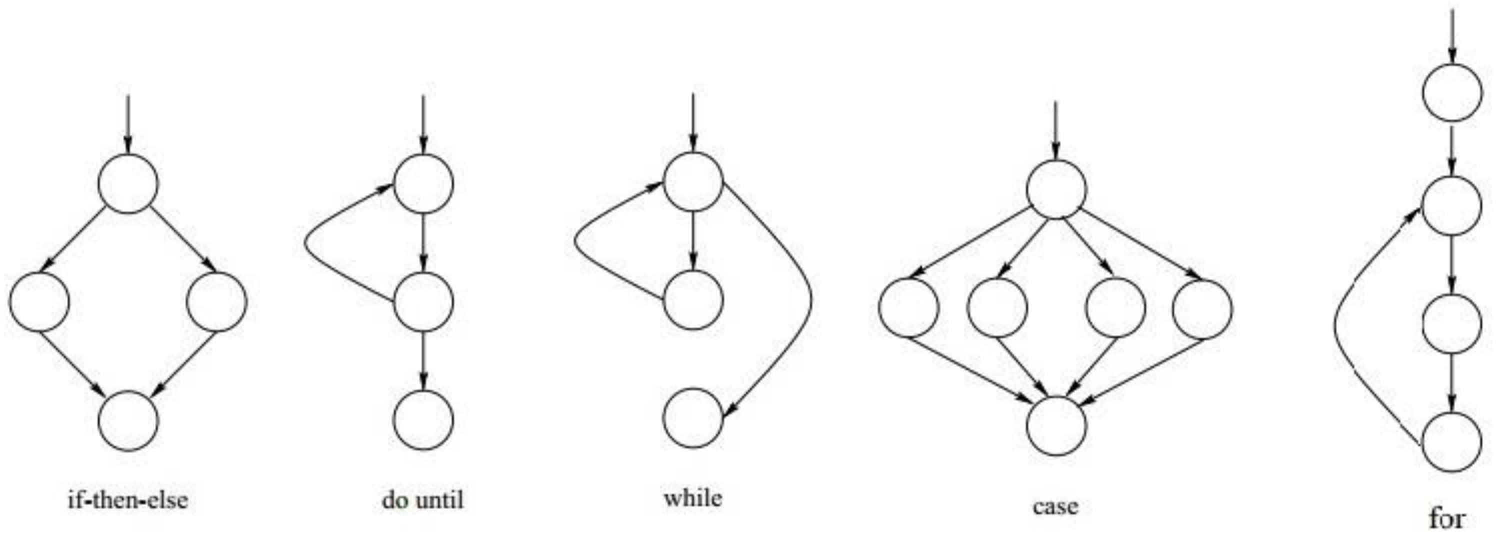


Figure 1: Flow graph representation.

LET’S CODE: ControlFlow.java

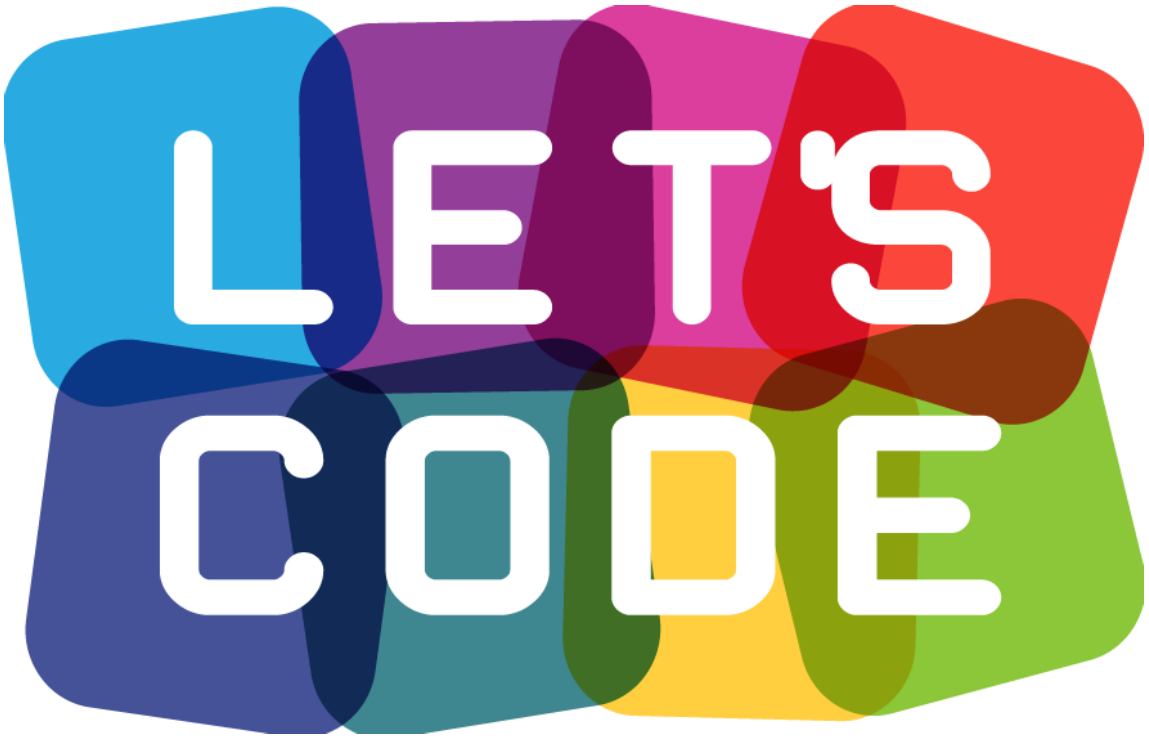
LET'S CODE

CONTENTS

INPUT SOURCES

- We will use as the source of information in our applications.
- They can be:
  - Arguments from command line
  - Read from command line
  - Files
  - Web Service
  - Queues, Topics
  - …

LET’S CODE: InputSources.java



CONTENTS

JAVA OOP

- Java is OOP, that means that we may work with objects.
- An object is a representation of a Class with attributes, functions, etc.
- You can provide more information about the class, such as the name of its superclass, whether it implements any interfaces, and so on…

CLASSES

- Class declarations can include these components, in order:
  - Modifiers such as public, private, …
  - The class name, with the initial letter capitalized by convention.
  - The name of the class's parent (superclass), if any, preceded by the keyword **extends**. A class can only extend (subclass) one parent.
  - A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword **implements**. A class can implement more than one interface.
  - The class body, surrounded by braces, {}.

MEMBER VARIABLES

- There are several kinds of variables:
  - Member variables in a class — these are called *fields*.
  - Variables in a method or block of code — these are called *local variables*.
  - Variables in method declarations — these are called *parameters*.

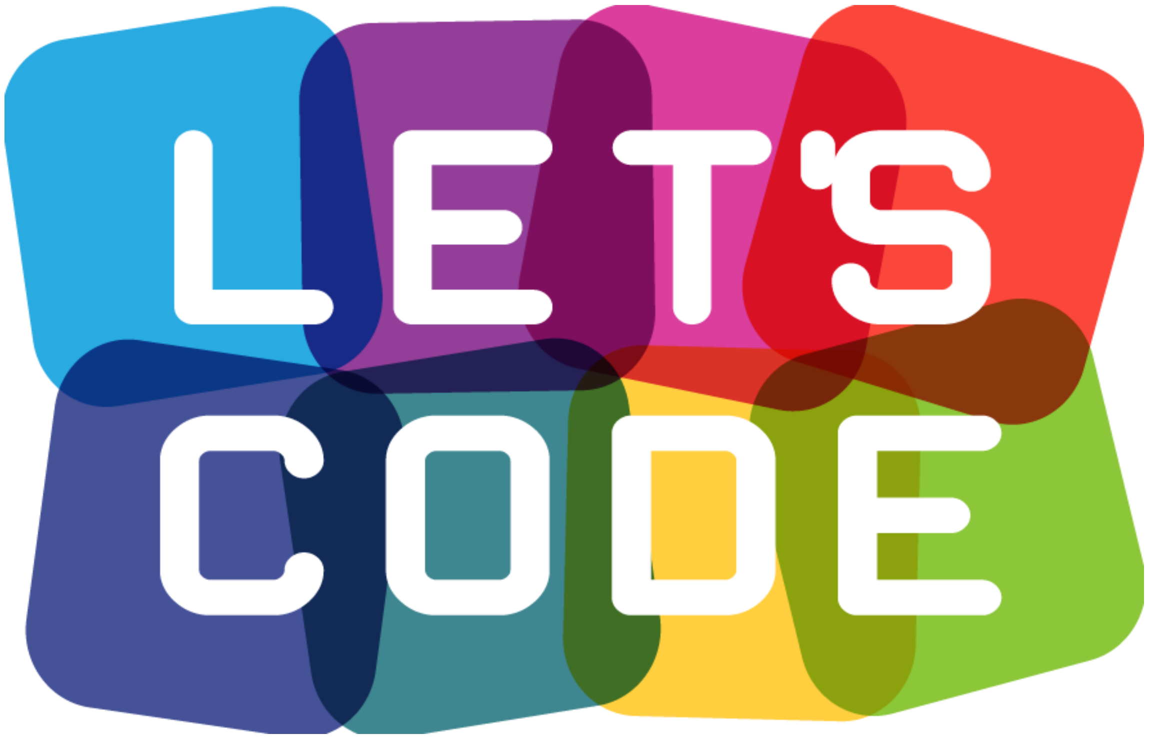
- Access:
  - **public** modifier — the field is accessible from all classes.
  - **private** modifier — the field is accessible only within its own class.

METHODS

- Method declarations have six components, in order:
  - Modifiers — such as public, private, …
  - The return type — such as void, int, String, …
  - The method name
  - The parameter list in parenthesis—a comma-delimited list of input parameters, preceded by their data types, enclosed by (). If there are no parameters, you must use empty parentheses.
  - An exception list
  - The method body, enclosed between {}

- There are a special method in each class that is the Constructor.

LET’S CODE: Classes.java



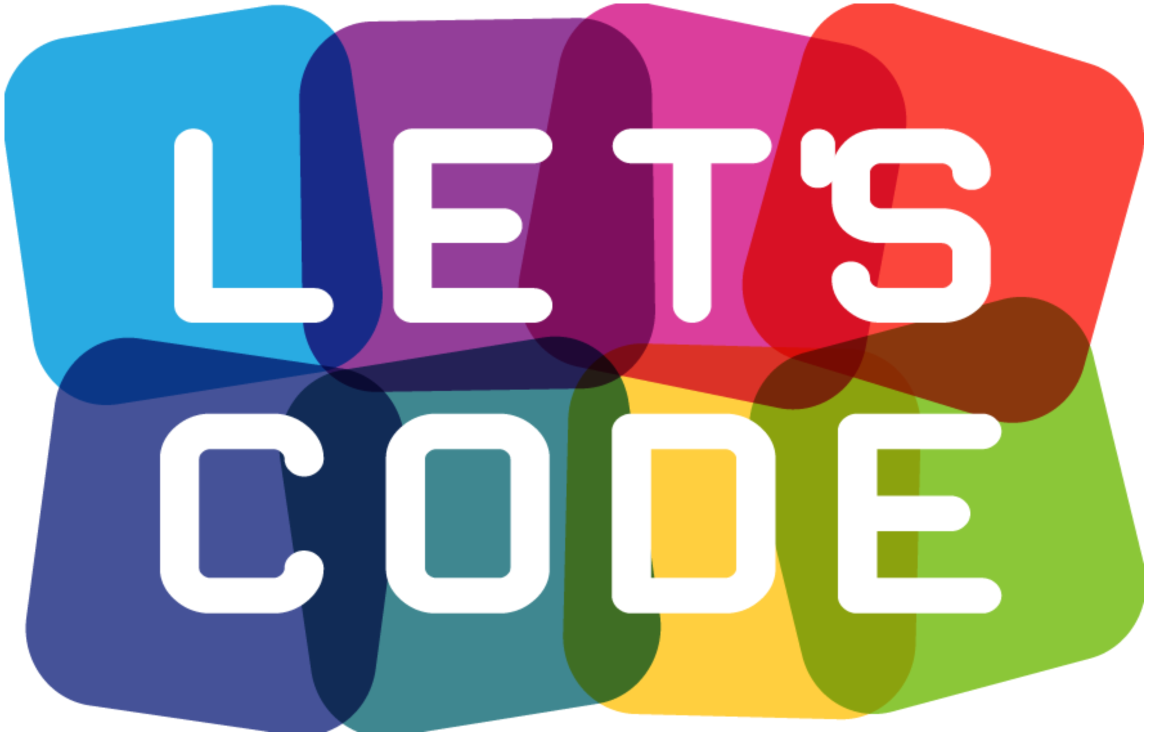
INTERFACES

- In the Java programming language, an *interface* is a reference type, similar to a class, that can contain *only* constants, method signatures, default methods, static methods, and nested types.

- Method bodies exist only for default methods and static methods. Interfaces cannot be instantiated.

- They can only be *implemented* by classes or *extended* by other interfaces.

LET’S CODE: Interfaces.java



INHERITANCE

- The idea of inheritance is simple but powerful:
  - When you want to create a new class and there is already a class that includes some of the code that you want, inheritance it!
  - In doing this, you can reuse the fields and methods of the existing class without having to write **(and debug!)** them yourself.

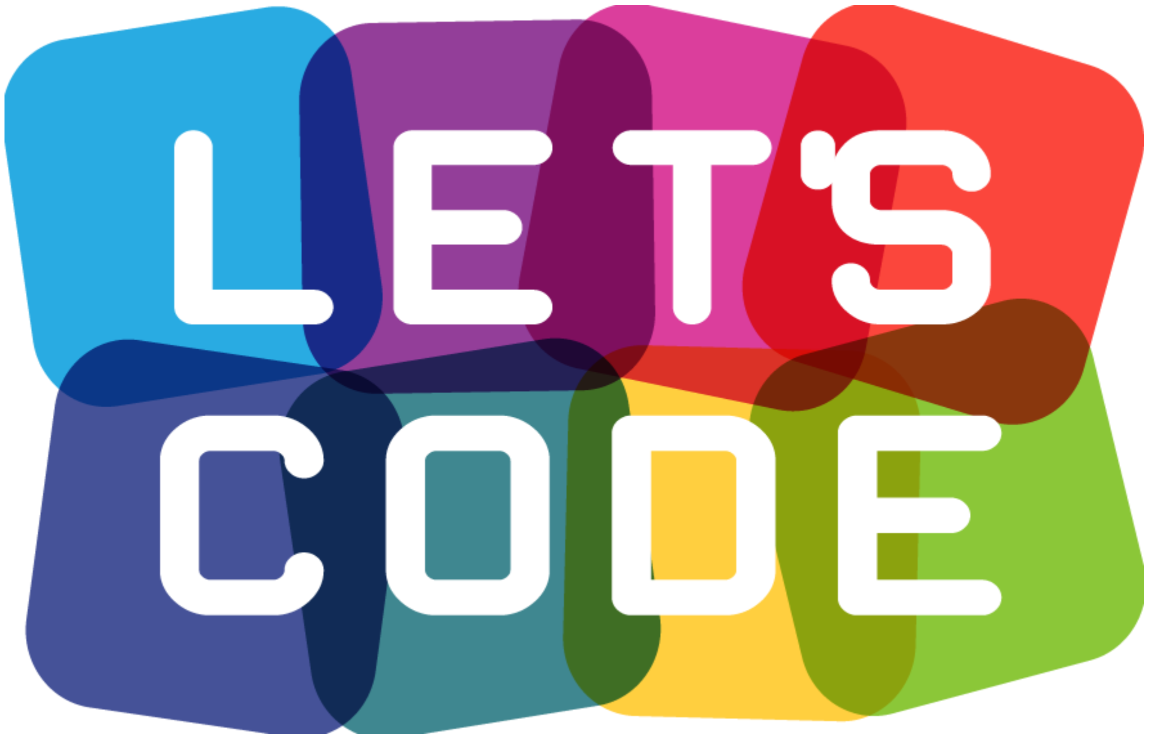
- A subclass inherits all of the *public* and *protected* members of its parent

- If the subclass is in the same package as its parent, it also inherits the *package-private* members of the parent

INHERITANCE

- The inherited fields can be used directly.
- You can declare a field in the subclass with the same name as the one in the superclass (not recommended).
- You can declare new fields in the subclass that are not in the superclass.
- The inherited methods can be used directly as they are.
- You can write a new instance method in the subclass that has the same signature as the one in the superclass, thus overriding it.
- You can write a new static method in the subclass that has the same signature as the one in the superclass, thus hiding it.
- You can declare new methods in the subclass that are not in the superclass.
- You can write a subclass constructor that invokes the constructor of the superclass, either implicitly or by using the keyword super.

LET’S CODE: Inheritance.java



CONTENTS

THREADS

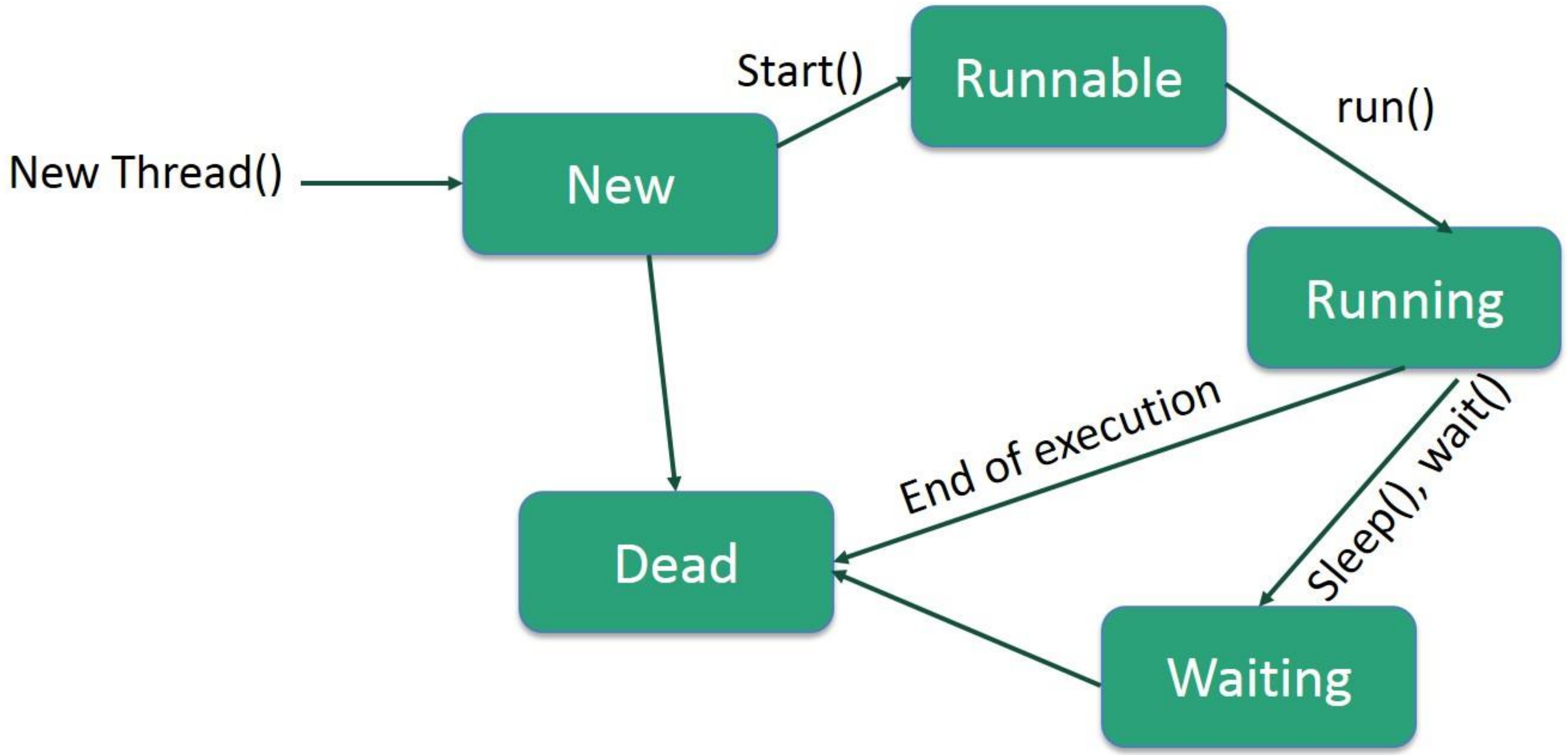
- Java is a *multi-threaded programming language* which means we can develop multi-threaded program using Java.

- Each of the threads can run in parallel.

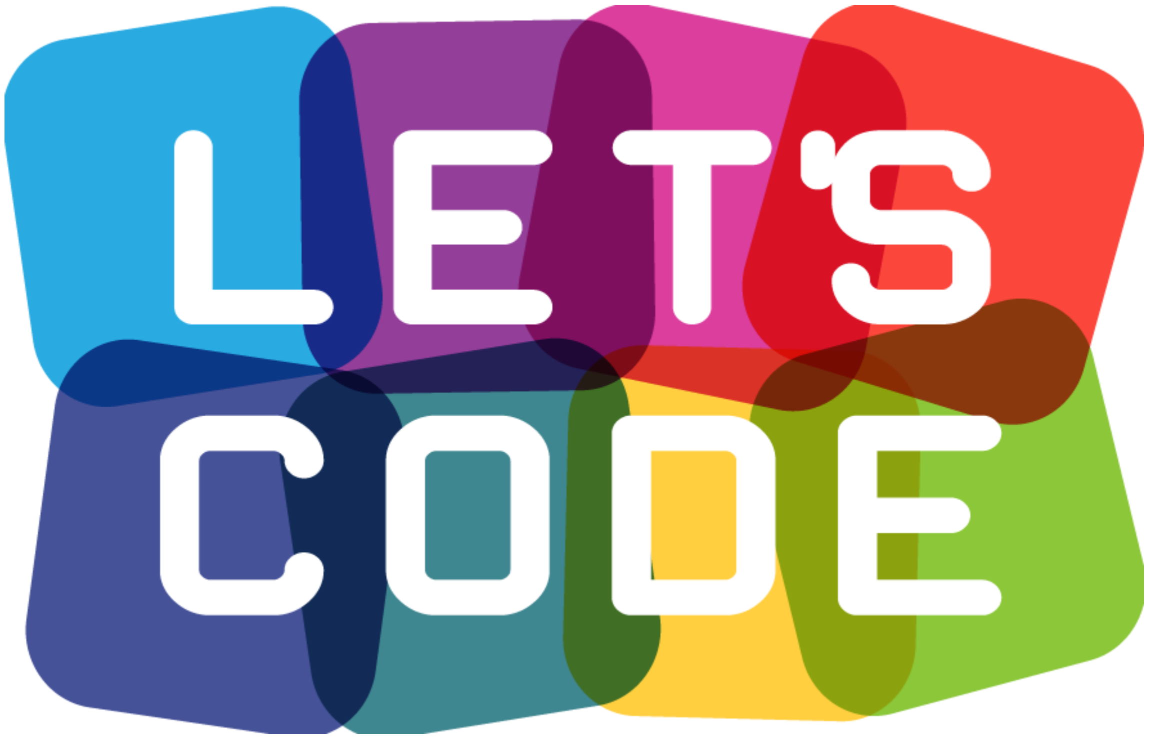
- Multi-threading enables you to write in a way where multiple activities can proceed concurrently in the same program.

- We can set priority to threads to determine the order in which them are scheduled.

LIFE CYCLE THREADS

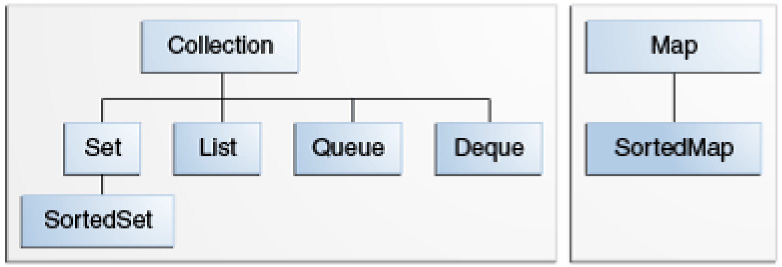


LET’S CODE: Multithreading.java



CONTENTS

COLLECTIONS IN JAVA



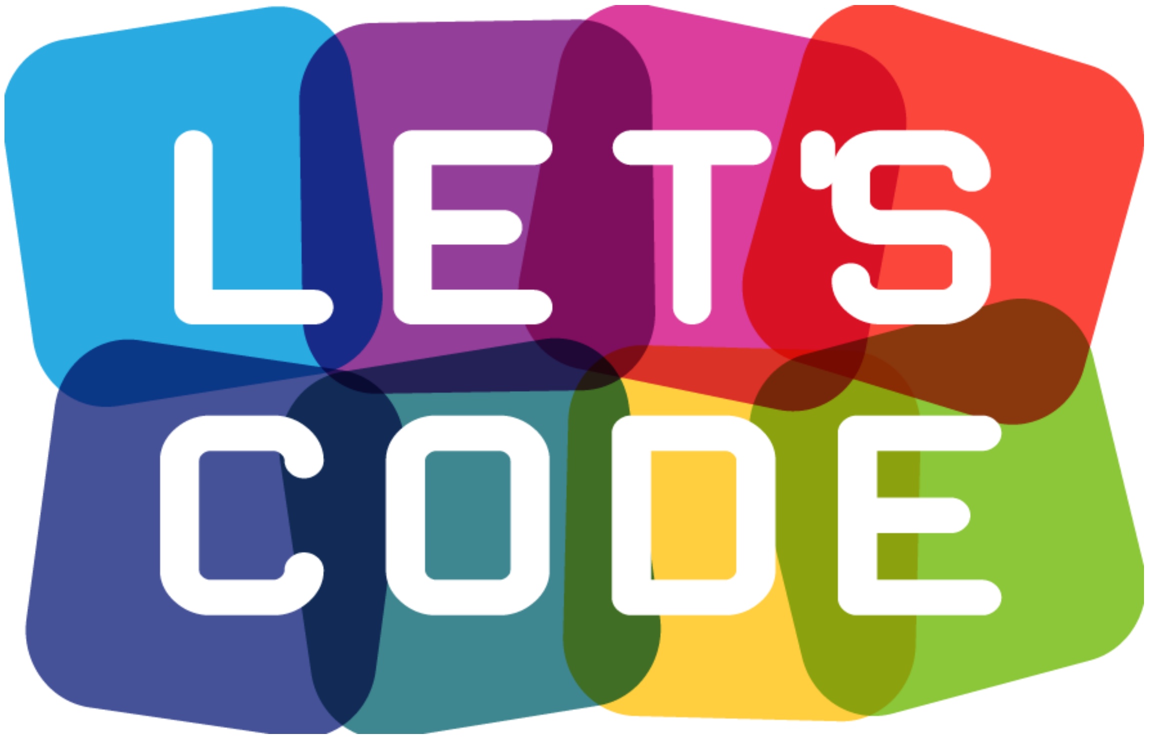
- Very useful to handle great quantities of elements

- **Set** — a collection that cannot contain duplicate elements.
- **List** — an ordered collection (sometimes called a sequence). Lists can contain duplicate elements.
- **Queue** / **Deque** — a collection used to hold multiple elements prior to processing.
- **Map** — an object that maps keys to values.
- **SortedSet** — a Set that maintains its elements in ascending order.
- **SortedMap** — a Map that maintains its mappings in ascending key order.

BULK OPERATORS

- **containsAll** — returns true if the target Collection contains all of the elements in the specified Collection.
- **addAll** — adds all of the elements in the specified Collection to the target Collection.
- **removeAll** — removes from the target Collection all of its elements that are also contained in the specified Collection.
- **retainAll** — removes from the target Collection all its elements that are not also contained in the specified Collection. That is, it retains only those elements in the target Collection that are also contained in the specified Collection.
- **clear** — removes all elements from the Collection.

LET’S CODE: Collections.java



CONTENTS

EXCEPTIONS

- **Definition**: An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

- The event, called an *exception object,* contains information about the error. Creating an exception object and handing it to the runtime system is called *throwing an exception*.

- Exceptions can be also handled in methods definitions.

MOS COMMON EXCEPTIONS

- NullPointerException: When trying illegal uses of the null object.
- NumberFormatException: When trying to convert a String into a Numeric variable.
- ArrayIndexOutOfBoundsException: When trying to access to an non-existing index of an Array.
- IllegalArgumentException: When a method has been passed an illegal or inappropriate argument.

LET’S CODE: Exceptions.java

LET'S
CODE

CONTENTS

PERSISTENCE IN JAVA

- Our applications need persistence in order to keep data always updated.
- Store in File or in memory is not a good alternative anymore.
- There are several drivers to connect our application with databases:
  - MySQL
  - MongoDB
  - ORACLE
  - …

LET’S CODE: Persistence.java

LET'S
CODE

CONTENTS

FOR YOUR INTEREST

- https://docs.oracle.com/javase/tutorial/java/data/index.html
- https://udemy.com/java-tutorial/
- http://mongodb.github.io/mongo-java-driver/3.7/
- https://blog.takipi.com/the-top-10-exceptions-types-in-production-java-applications-based-on-1b-events/

THANK YOU VERY MUCH!

David Corra Plaza