Large-Scale and Multi-Structured Databases

JAVA Recap

Prof Pietro Ducange

Eng. José Corcuera

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

UNIVERSITÀ DI PISA

CROSSLAB Innovation for industry 4.0

Objective of this Module

- Let the student to think and remember his/her skills about programming in JAVA.
- Indeed, it is assumed that the student is able to work with JAVA.
- We will recall some basics elements of JAVA.
- Students will be asked to solve some programming exercises using an IDE (Eclipse or NetBeans).

Some Salient Characteristics of Java

- Java is ***platform independent:*** the same program can run on any correctly implemented Java system
- Java is ***object-oriented:***
  - Structured in terms of ***classes***, which group data with operations on that data
  - Can construct new classes by ***extending*** existing ones
- Java designed as
  - A ***core language*** plus
  - A rich collection of ***commonly available packages***
- Java can be embedded in Web pages

Program Structure

- Typical Java program consists of
  - ***User*** written ***classes***
  - Java Application Programming Interface (***API***) classes
- Java application
  - Has one class with a ***main*** method
- Java program basic elements:
  - Packages
  - Classes
  - Data fields
  - Methods

Java Processing and Execution

- Begin with Java ***source code*** in text files: **`Model.java`**
- A Java source code compiler (**javac**) produces Java ***byte code***
  - Outputs one file per class: **`Model.class`**
  - May be standalone or part of an IDE
- A ***Java Virtual Machine*** loads and executes class files (**java**)
  - May compile them to native code (e.g., x86) internally

Classes and Objects

- The ***class*** is the unit of programming
- A Java program is a ***collection of classes***
  - Each class definition (usually) in its own **`.java`** file
  - *The file name must match the class name*
- A class describes ***objects (instances)***
  - Describes their common characteristics
  - Thus all the instances have these same characteristics
- These characteristics are:
  - ***Data fields*** for each object
  - ***Methods*** (operations) that do work on the objects

Classes and Objects

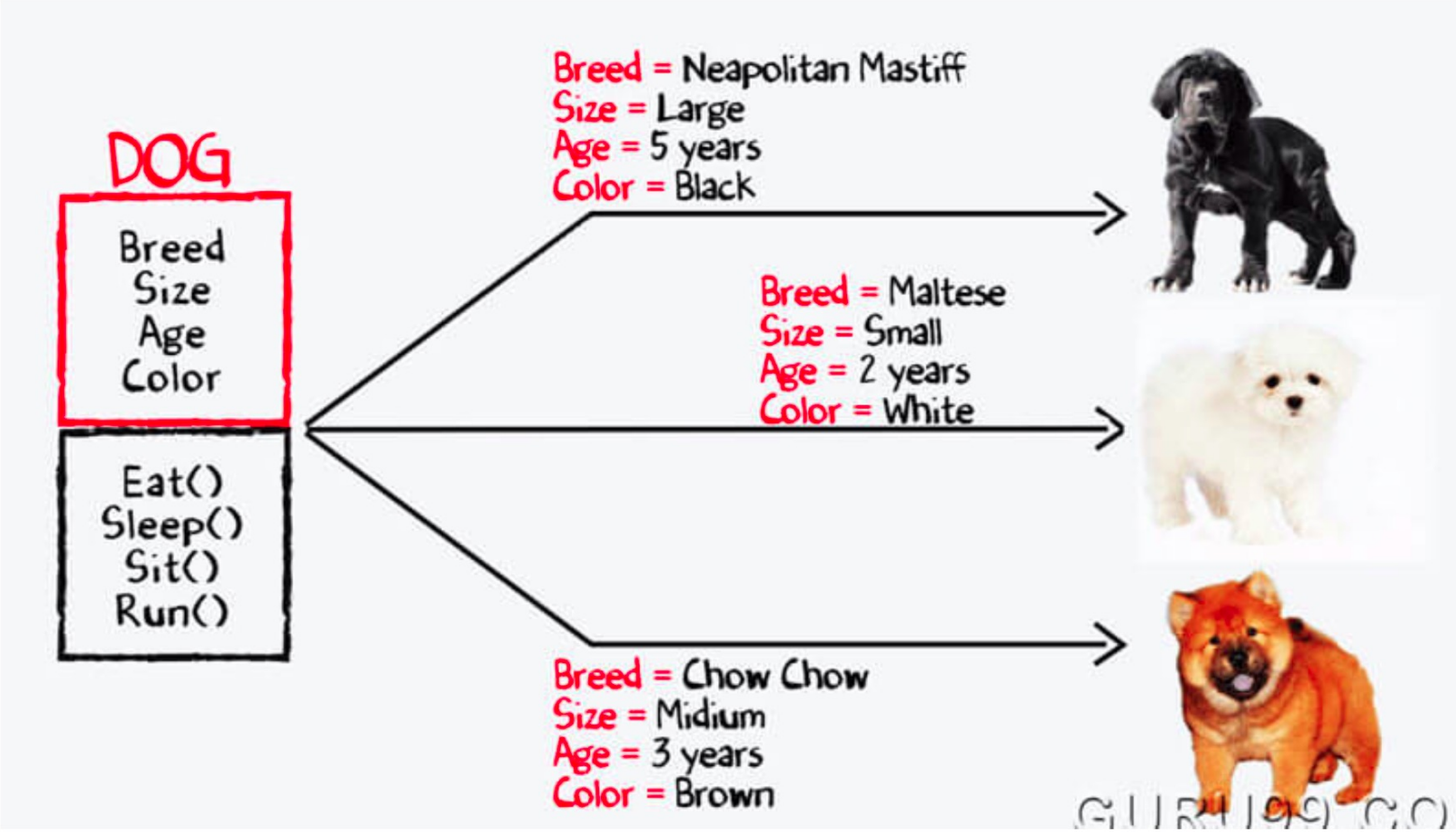


Image extracted from: https://www.guru99.com/java-oops-class-objects.html

Classes and Objects

```
// Class Declaration
public class Dog {
    // Instance Variables
    String breed;
    String size;
    int age;
    String color;


    // method 1
    public String getInfo() {
        return ("Breed is: "+breed+" Size is:"+size+" Age is:"+age+" color is: "+color);
    }


    public static void main(String[] args) {
        Dog maltese = new Dog();
        maltese.breed="Maltese";
        maltese.size="Small";
        maltese.age=2;
        maltese.color="white";
        System.out.println(maltese.getInfo());
    }
}
```

Image extracted from: https://www.guru99.com/java-oops-class-objects.html

Polymorphism

Polymorphism is the ability to create a variable, a function, or an object that has more than one form.

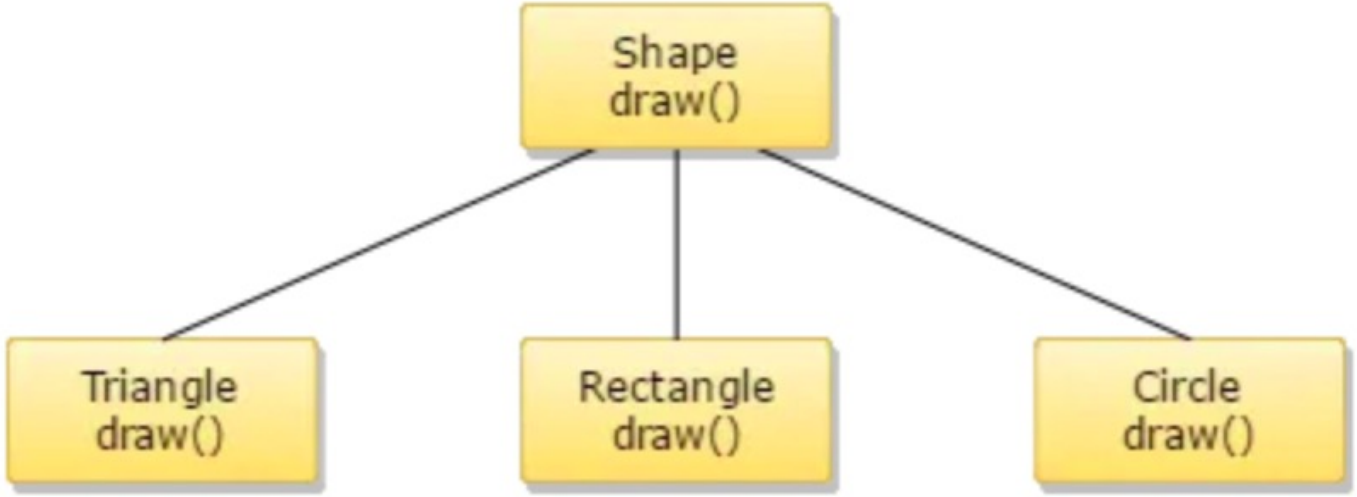


Image extracted from: https://www.w3schools.in/java/polymorphism/

Polymorphism

```java
public class Animal {
    public void makeNoise()
    {
        System.out.println("Some sound");
    }
}

class Dog extends Animal{
    public void makeNoise()
    {
        System.out.println("Bark");
    }
}

class Cat extends Animal{
    public void makeNoise()
    {
        System.out.println("Meawoo");
    }
}
```

```java
public class Demo
{
    public static void main(String[] args) {
        Animal a1 = new Cat();
        a1.makeNoise(); //Prints Meowoo

        Animal a2 = new Dog();
        a2.makeNoise(); //Prints Bark
    }
}
```

Image extracted from: https://howtodoinjava.com/java/oops/what-is-polymorphism-in-java/

Grouping Classes: The Java API

- API = *Application Programming Interface*
- Java = small core + extensive collection of packages
- A ***package*** consists of some related Java classes
- The ***import*** statement tells the compiler to make available classes and methods of another package
- A ***main*** method indicates where to begin executing a class (if it is designed to be run as a program)

References and Primitive Data Types

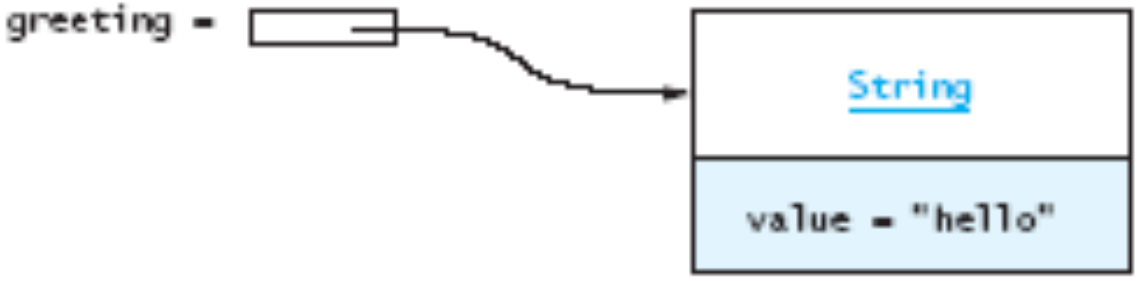
- Java distinguishes two kinds of entities
  - Primitive types
  - Objects
- Primitive-type data is stored in primitive-type variables
- Reference variables store the *address of* an object

Primitive Data Types

- Represent numbers, characters, boolean values
- Integers: byte, short, int, and long
- Real numbers: float and double
- Characters: char

Referencing and Creating Objects

- We can **declare reference variables**
  - They reference objects of **specified types**
- Two reference variables can reference **the same object**
- The **`new`** operator creates an instance of a class
- A **constructor** executes when a new object is created
- Example: **`String greeting = "hello";`**



Methods

- A Java method defines a group of statements as performing a particular operation
- **`static`** indicates a ***static*** or ***class*** method
- A method that is not **`static`** is an ***instance*** method
- All method arguments are ***call-by-value***
  - Primitive type: *value* is passed to the method
  - Method may modify local copy ***but*** will not affect caller's value
  - Object reference: *address of object* is passed
  - Change to reference variable does not affect caller
  - ***But*** operations can affect the object, visible to caller

Exercise 0

- Download your preferred IDE (include MAVEN) and install it
  - Eclipse: https://www.eclipse.org/
  - IntelliJ IDEA: https://www.jetbrains.com/idea/
  - Netbeans: https://netbeans.org/
- Create a Java Project
- Write and Run your First “Hello World!” program.

Exercise 0

How long it Takes?

Exercise 1

Write a Java program that takes three numbers from the user and prints the greatest number.

Input Data:

Input the 1st number: 25

Input the 2nd number: 78

Input the 3rd number: 87

Expected Output:

The greatest: 87

Exercise 2

Write a Java method to count all lowercase vowels in a string.

Input Data:

Input the string: unipi

Expected Output:

Number of Vowels in the string: 3

Exercise 3

Write a Java program to retrieve elements (at a specified index) from a given array list.

Test Data:

Consider to create the following array list[Red, Green, Orange, White, Black] and to retrieve the first and the third element

Expected Output:

First element: Red

Third element: Orange

Exercise 4 (homework)

Write a Java program that takes N numbers from the user and prints the greatest number. The first line specify the N numbers to read and the second a list of number separated by a white space.

Input:

6

25 78 87 154 -45 2023

Output :

2023

Exercise 5 (homework)

Given a list L of N – 1 numbers, where:

- 1 <= L[i] <= N
- 0 <= i < N - 1

Find the missing number. The first line specify the value of N.

Input:

6

1 5 6 3 2

Output :

4

Exercise 6 (homework)

Given a value N, compute its Fibonacci.

Input:

56

Output :

225851433717

Try to implement it with/without using recursion...

Suggested Readings

https://docs.oracle.com/javase/tutorial/

Mitsunori Ogihara, Fundamentals of Java Programming, Springer , 2018

Check books on http://onesearch.unipi.it/