div. teori

Class. In Java, you implement a data type in a class. As usual, we put the code for a data type in a file with the same name as the class, followed by the .java extension.

In summary, to define a data type in a Java class, you need instance variables, constructors, instance methods,

Access modifiers. We designate every instance variable and method within a class as either public (this entity is accessible by clients) or private (this entity is not accessible by clients). The final modifier indicates that the value of the variable will not change once it is initialized—its access is read-only.

Constructors. A constructor is a special method that creates an object and provides a reference to that object. Java automatically invokes a constructor when a client program uses the keyword new. Each time that a client invokes a constructor, Java automatically

Encapsulation. The process of separating clients from implementations by hiding information is known as encapsulation. We use encapsulation to enable modular programming, facilitate debugging, and clarify program code. Private. When you declare an instance variable (or method) to be private, you are making it impossible for any client (code in another class) to directly access that instance variable (or method). This helps enforce encapsulation.

Immutable- An object from a data type is immutable if its data-type value cannot change once created. An immutable data type is one in which all objects of that type are immutable. The main drawback of immutability is that a new object must be created for every value. Final. When you declare an instance variable as final, you are promising to assign it a value only once. This helps enforce immutability. - Final garanterer dog ikke immutability. defensive copy.

Java provides the *interface* construct for declaring a relationship between otherwise unrelated classes, by specifying a common set of methods that each implementing class must include. Interfaces enable us to write client programs that can manipulate objects of varying types, by invoking common methods from the interface.

The body of the interface contains a list of abstract methods. An abstract method is a method that is declared but does not include any implementation code; it contains only the method signature. You must save a Java interface in a file whose name matches the name of the interface, with a .java extension.

Implementation inheritance (subclassing). Java also supports another inheritance mechanism known as subclassing. The idea is to define a new class (subclass, or derived class) that inherits instance variables (state) and instance methods (behavior) from another class (superclass, or base class), enabling code reuse. Typically, the subclass redefines or overrides some of the methods in the superclass.

Exceptions: An exception is a disruptive event that occurs while a program is running, often to signal an error. The action taken is known as throwing an exception. Java includes an elaborate inheritance hierarchy of predefined exceptions, several of which we have encountered previously.

A compiler is an application that translates programs from the Java language to a language more suitable for executing on the computer. It takes a text file with the .java extension as input (your program) and produces a file with a .class extension (the computer-language version). To compile HelloWorld.java type the boldfaced text below at the terminal. (We use the % symbol to denote the command prompt, but it may appear different depending on your system.) %javac HelloWorld.java.

Executing (or running) a Java program. Once you compile your program, you can execute it. This is the exciting part, where the computer follows your instructions. To run the HelloWorld program, type the following in the terminal window: % java HelloWorld

Nesting. The if, while, and for statements have the same status as assignment statements or any other statements in Java; that is, we can use them wherever a statement is called for. In particular, we can use one or more of them in the body of another statement to make compound statements. To emphasize the nesting, we use indentation in the program code.

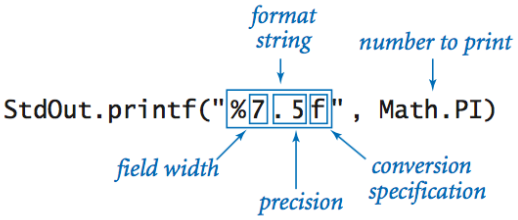
```
public class MarginalTaxRate {

   public static void main(String[] args) {
      int income = Integer.parseInt(args[0]);
      double rate = 0.0;

      if     (income <      0) rate = 0.00;
      else if (income <   8925) rate = 0.10;
      else if (income <  36250) rate = 0.15;
      else if (income <  87850) rate = 0.23;
      else if (income < 183250) rate = 0.28;
      else if (income < 398350) rate = 0.33;
      else if (income < 400000) rate = 0.35;
      else                      rate = 0.396;

      System.out.println("marginal tax rate = " + rate);
   }
}
```

Formatted printing basics. In its simplest form, printf() takes two arguments. The first argument is called the format string. It contains a conversion specification that describes how the second argument is to be converted to a string for output.



A data type is a set of values and a set of operations defined on those values. The primitive data types that you have been using are supplemented in Java by extensive libraries of reference types that are tailored for a large variety of applications. In this section, we consider reference types for string processing and image processing.