OOP

Object Oriented Programming, CS211

Dr. Sirojiddin Jo’raev

About me

- Background
  - B.Sc: Tashkent University of Information Technology (TUIT)
  - Master: Yeungnam University, South Korea
  - PhD: Yeungnam University, South Korea
- Working Experience
  - New Uzbekistan University, Assistant Professor 2021~ Present
  - Computer Network Lab Research Assistant and Teaching Assistant 2011-2021
  - Korean Center, Researcher 2011

Contact information

- Contact details: s.juraev@newuu.uz
- Office hours: Monday 10AM ~11AM, Tuesday 10AM ~ 11AM
- Office : UCA Building 2nd floor, 218

Basic course information

Subject	Object Oriented Programming, CS211
ECTS Credits	6
Content	Introduction to Programming and Java. Introduction to Java Applications. Objects and Classes. Inheritance and interfaces in Java. Polymorphism. Lifecycle of Objects. The SOLID principles of OOP and Design Patterns. GUI components in Java (Java FX) JAVA Relationship Database Development, JDBC The Collections Framework
Prerequisites	Introduction to Programming, C201

Textbooks and reference books

- Liang Y.D. - Introduction to Java Programming and Data Structures, Comprehensive Version, 12th edition – 2020
- Java How to Program, Early Objects, Harvey Deitel
- Head First Java 2nd edition, Bert Bates and Kathy Sierra
- Object-Oriented Software Engineering Using UML, Patterns, and Java™, 3rd edition, Bernd Bruegge & Allen H. Dutoit

Grading Criteria

Grading criteria	Total points
Midterm	30 points
Final	40 points
Homework Assignments	**5 points**
Lab Assignments	**10 points**
Team Project Assignment	**15 points**
Total	100 points

Team Project and Assignments

- All individual assignments should be submitted through GitHub account. There are total 3 homework till Midterm.
- There will be team project that each team will pick up their own projects.
- Students will be required to form themselves into 5 ~ 6 pairs of members each.

Class policies

- Students are **responsible for all missed work**, regardless of the reason for absence.
- Homework deadline usually one week from assigned time. I will not accept any homework after deadline. There is no single point after deadline.
- You have only two free extension (Late days)

Honor Code

- Don’t share your code
- If you share your code, **you** and **other person** will get 0 point
- Don’t copy single line of code from anyone
- You can discuss about assignment as much as you want.
- You can ask suggestion as much as you want.
- You can come to my office during office hours discuss the issue.

Attendance

75%

Tools

- IntelliJ IDEA Community Edition-open source and free
- StarUML is open source a software engineering tool for system modeling using the Unified Modeling Language
- You can use university computers or personal computer
- We use GitHub to submit homework and assignments

Today’s topic

- Introduction to OOP.
- About Java: What is it? Why Java? and When to use Java? SDK ? JVM,JRE,JDK
- Introduction to Object Technology
- The UML
- Software Technologies
- Developing Java Programs Using IntelliJ IDEA
- A Simple Java Program
- Variables
- Named Constants
- Numeric Data Types and Operations
- Reading Input from the Console
- Software Development Process

Who is this course for?

- Have you done some programming?
- Do you want to learn OOP using Java?
- Do you prefer typing code and understand how it works why it works and why it doesn't work ?
- Do you want to learn how to manage and work with team?

Who started it ?

- Ivan Sutherland’s seminal Sketchpad application was an early inspiration for OOP, created between 1961 and 1962 and published in his Sketchpad Thesis.
- Any object could become a ”master” and additional instances of the objects were called ”occurrance”.

Who invented Objects, Classes, and Inheritance

- **Simula** was developed in the 1965 at the Norwegian Computing Center in Oslo, by Ole-Johan Dahl and Kristen Nygaard.

- Like Sketchpad, Simula featured objects, and eventually introduced classes, class inheritance, subclasses, and virtual methods.

(c) Wikipedia

Who coined “OOP”?

- Smalltalk was created in the 1970s at Xerox PARC by Learning Research Group (LRG) scientists, including
- Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Diana Merry, and Scott Wallace.

Who make popular OOP?

- C++ was created by Danish computer scientist Bjarne Stroustrup in 1985, by enhancing C language.
- C was chosen because it was general-purpose, fast, portable and widely used.

What happened next?

- C++ was released in 1985. And then...
- Erlang 1986
- Eiffel 1986
- Self 1987
- Perl 1988
- Haskell 1990
- Python 1991
- Lua 1993
- JavaScript 1995
- Ruby 1995
- **Java 1995**
- Go 1995
- PHP3 1998
- C# 2000
- Rust 2010
- Swift 2014

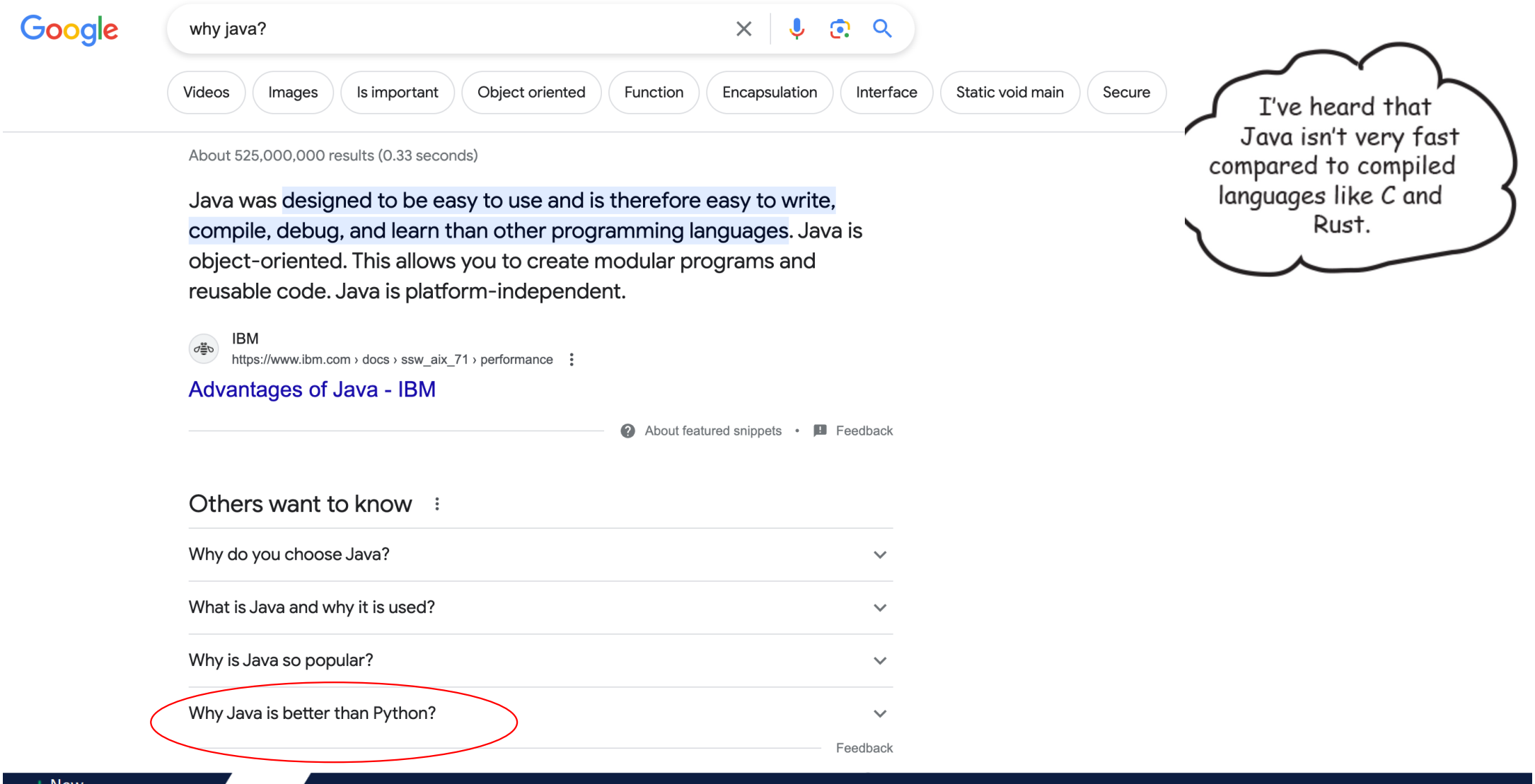
Incomplete list of OOP features, so far:

- Polymorphism
- Nested Objects
- Traits
- Templates
- Generics
- Invariants
- Classes
- NULL
- Exceptions
- Operators
- Methods
- Static Blocks
- Virtual Tables
- Coroutines
- Annotations
- Interfaces
- Constructors
- Destructors
- Synchronization
- Macros
- Inheritance
- Overloading
- Encapsulation
- Access Modifiers
- Pattern Matching
- Enumerated Types
- Namespaces
- Modules
- Type Aliases
- Decorators
- Lambda Functions
- Multiple Inheritance
- Events
- Callbacks
- NULL Safety
- Immutability

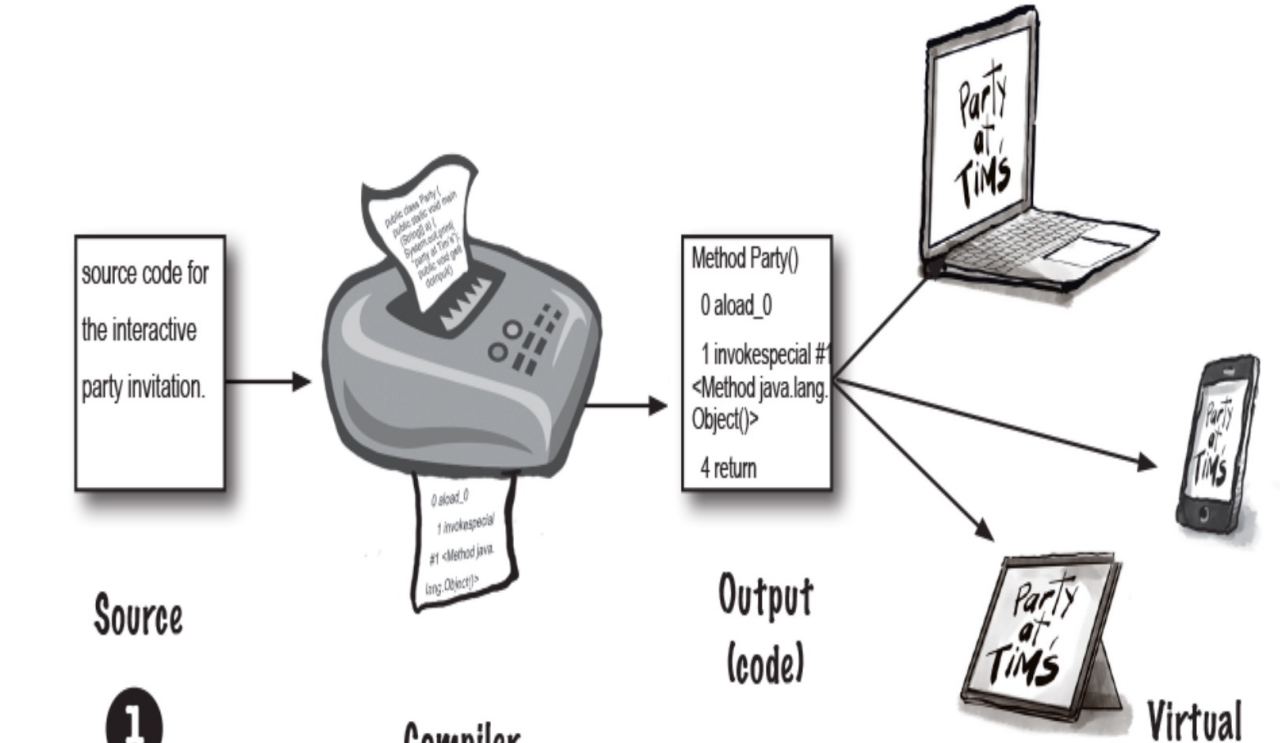
Why use OOP?

- Object Oriented Programming is one of the most widely used programming paradigm.
- Well suited for building trivial and complex application
- Allows re-use of code thereby increasing productivity
- New features can be easyly built into the existing code
- Reduced production cost and maintenance cost

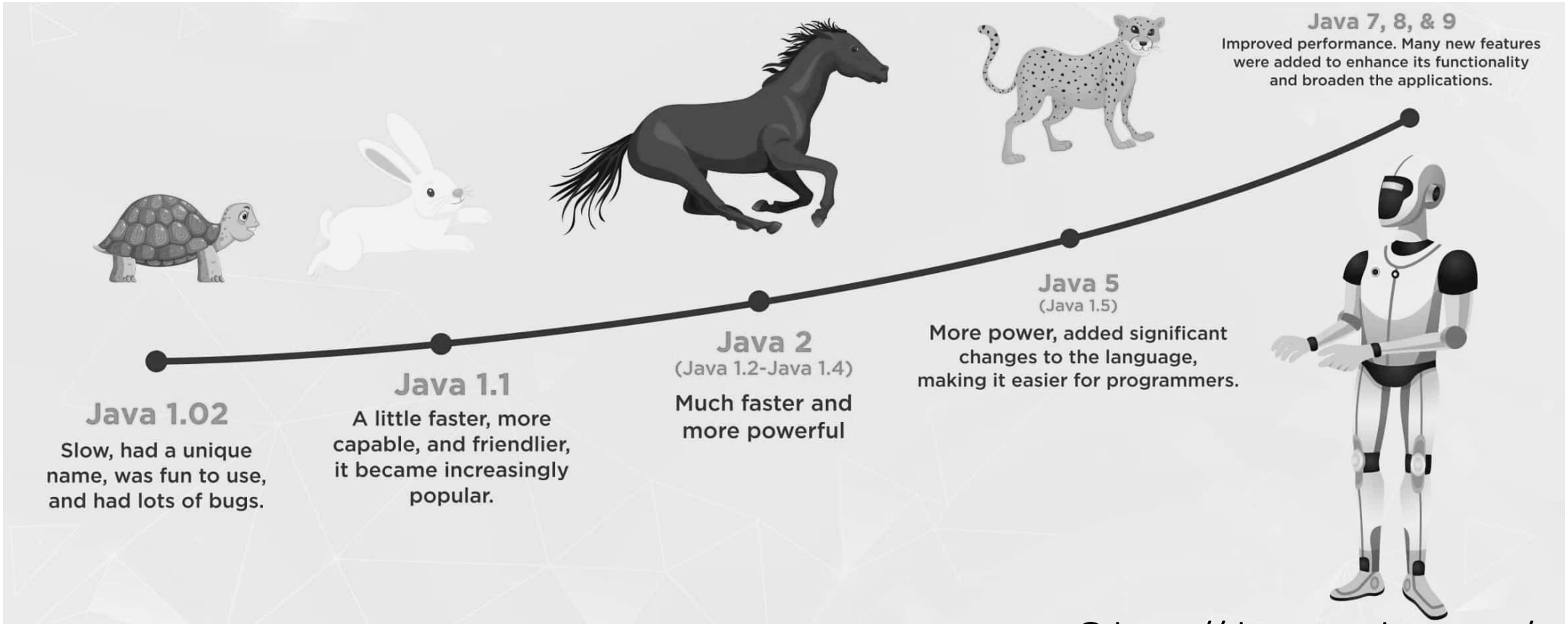
Anything you can do in one programming language, you can do in another programming language


Google
why java?
Videos
Images
Is important
Object oriented
Function
Encapsulation
Interface
Static void main
Secure
About 525,000,000 results (0.33 seconds)
Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages. Java is object-oriented. This allows you to create modular programs and reusable code. Java is platform-independent.
IBM
https://www.ibm.com › docs › ssw_aix_71 › performance
Advantages of Java - IBM
About featured snippets
Feedback
Others want to know
Why do you choose Java?
What is Java and why it is used?
Why is Java so popular?
Why Java is better than Python?
Feedback
I've heard that Java isn't very fast compared to compiled languages like C and Rust.

How Java Works?

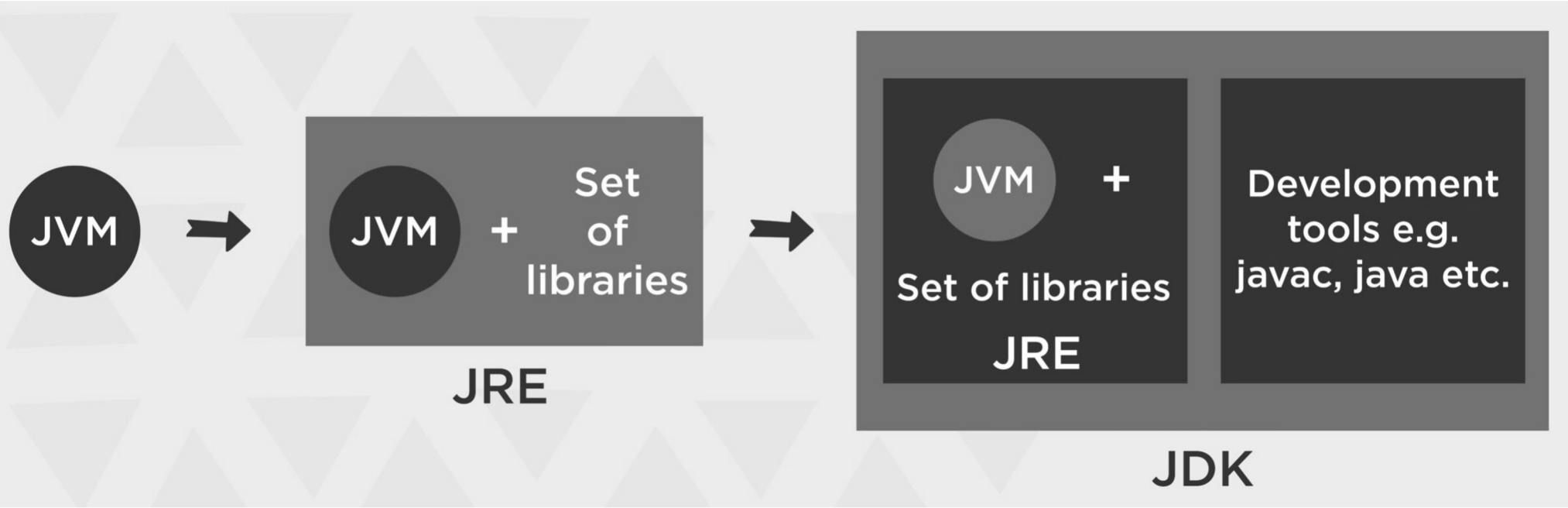


Breif history of Java



- © https://thetapacademy.com/

Difference between JVM, JRE, and JDK



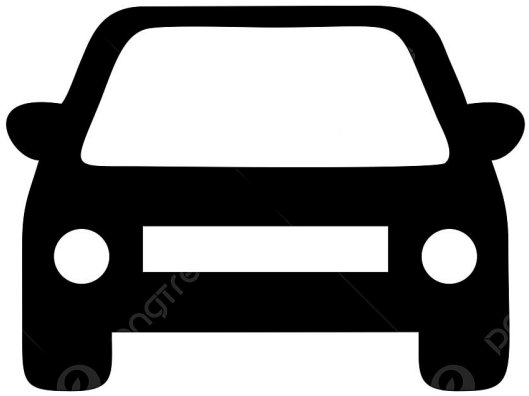
- © https://thetapacademy.com/

Introduction to Object Technology

- The Automobile as an Object

Main.java | Car.java

```java
public class Car {
    private String color;
    private int capacity;
    private double weight;
    private double engineSize;
    public void start(){
        //start engine
    }
    public void stop(){
        //stop engine
    }
    public void accelerate(){
        //accelerate car
    }
}
```



Introduction to Object Technology

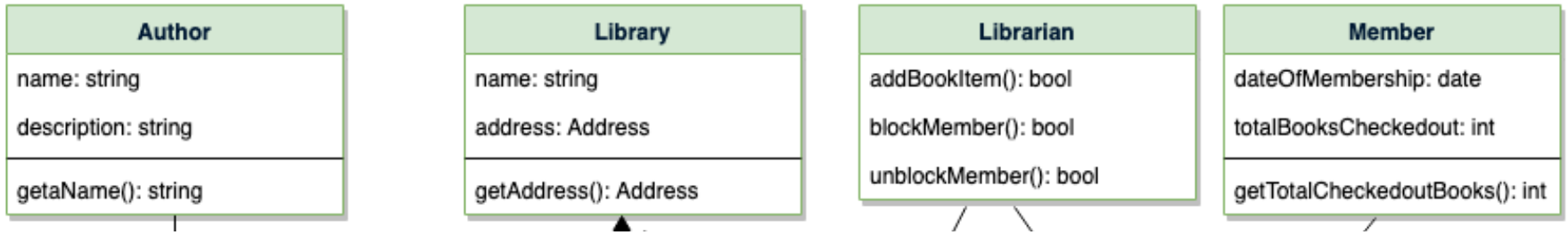
- **Methods and Classes**
- Performing a task in a program requires a **method.**
- We create a program unit called a **class** to house the set of methods the classes task
- Class is blueprint for objects.

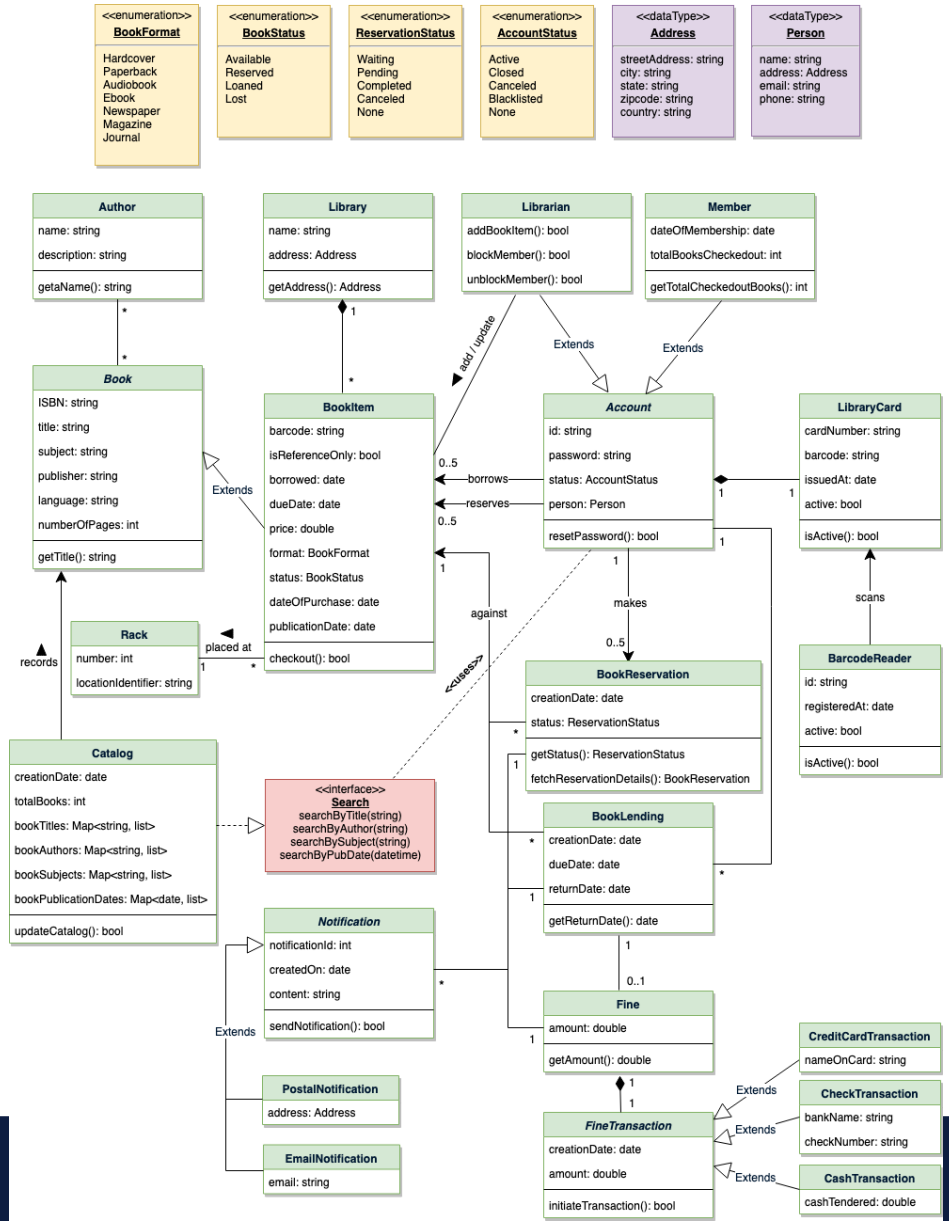
Main.java Car.java

```java
public class Car {
    private String color;
    private int capacity;
    private double weight;
    private double engineSize;
    public void start(){
        //start engine
    }
    public void stop(){
        //stop engine
    }
    public void accelerate(){
        //accelerate car
    }
}
```

The UML (Unified Modeling Language)

- The goal of UML is to provide a standard notation that can be used by all object-oriented programming languages.
- UML has been designed for a broad range of applications.




<<enumeration>>
BookFormat
Hardcover
Paperback
Audiobook
Ebook
Newspaper
Magazine
Journal
<<enumeration>>
BookStatus
Available
Reserved
Loaned
Lost
<<enumeration>>
ReservationStatus
Waiting
Pending
Completed
Canceled
None
<<enumeration>>
AccountStatus
Active
Closed
Canceled
Blacklisted
None
<<dataType>>
Address
streetAddress: string
city: string
state: string
zipcode: string
country: string
<<dataType>>
Person
name: string
address: Address
email: string
phone: string
Author
name: string
description: string
getaName(): string
Library
name: string
address: Address
getAddress(): Address
Librarian
addBookItem(): bool
blockMember(): bool
unblockMember(): bool
Member
dateOfMembership: date
totalBooksCheckedout: int
getTotalCheckedoutBooks(): int
add / update
Extends
Extends
Book
ISBN: string
title: string
subject: string
publisher: string
language: string
numberOfPages: int
getTitle(): string
BookItem
barcode: string
isReferenceOnly: bool
borrowed: date
dueDate: date
price: double
format: BookFormat
status: BookStatus
dateOfPurchase: date
publicationDate: date
checkout(): bool
Account
id: string
password: string
status: AccountStatus
person: Person
resetPassword(): bool
LibraryCard
cardNumber: string
barcode: string
issuedAt: date
active: bool
isActive(): bool
Extends
0..5
borrows
reserves
0..5
against
makes
0..5
scans
records
Rack
number: int
locationIdentifier: string
placed at
<<uses>>
BarcodeReader
id: string
registeredAt: date
active: bool
isActive(): bool
BookReservation
creationDate: date
status: ReservationStatus
getStatus(): ReservationStatus
fetchReservationDetails(): BookReservation
Catalog
creationDate: date
totalBooks: int
bookTitles: Map<string, list>
bookAuthors: Map<string, list>
bookSubjects: Map<string, list>
bookPublicationDates: Map<date, list>
updateCatalog(): bool
<<interface>>
Search
searchByTitle(string)
searchByAuthor(string)
searchBySubject(string)
searchByPubDate(datetime)
BookLending
creationDate: date
dueDate: date
returnDate: date
getReturnDate(): date
Notification
notificationId: int
createdOn: date
content: string
sendNotification(): bool
0..1
Fine
amount: double
getAmount(): double
Extends
PostalNotification
address: Address
EmailNotification
email: string
FineTransaction
creationDate: date
amount: double
initiateTransaction(): bool
CreditCardTransaction
nameOnCard: string
CheckTransaction
bankName: string
checkNumber: string
CashTransaction
cashTendered: double
Extends
Extends
Extends

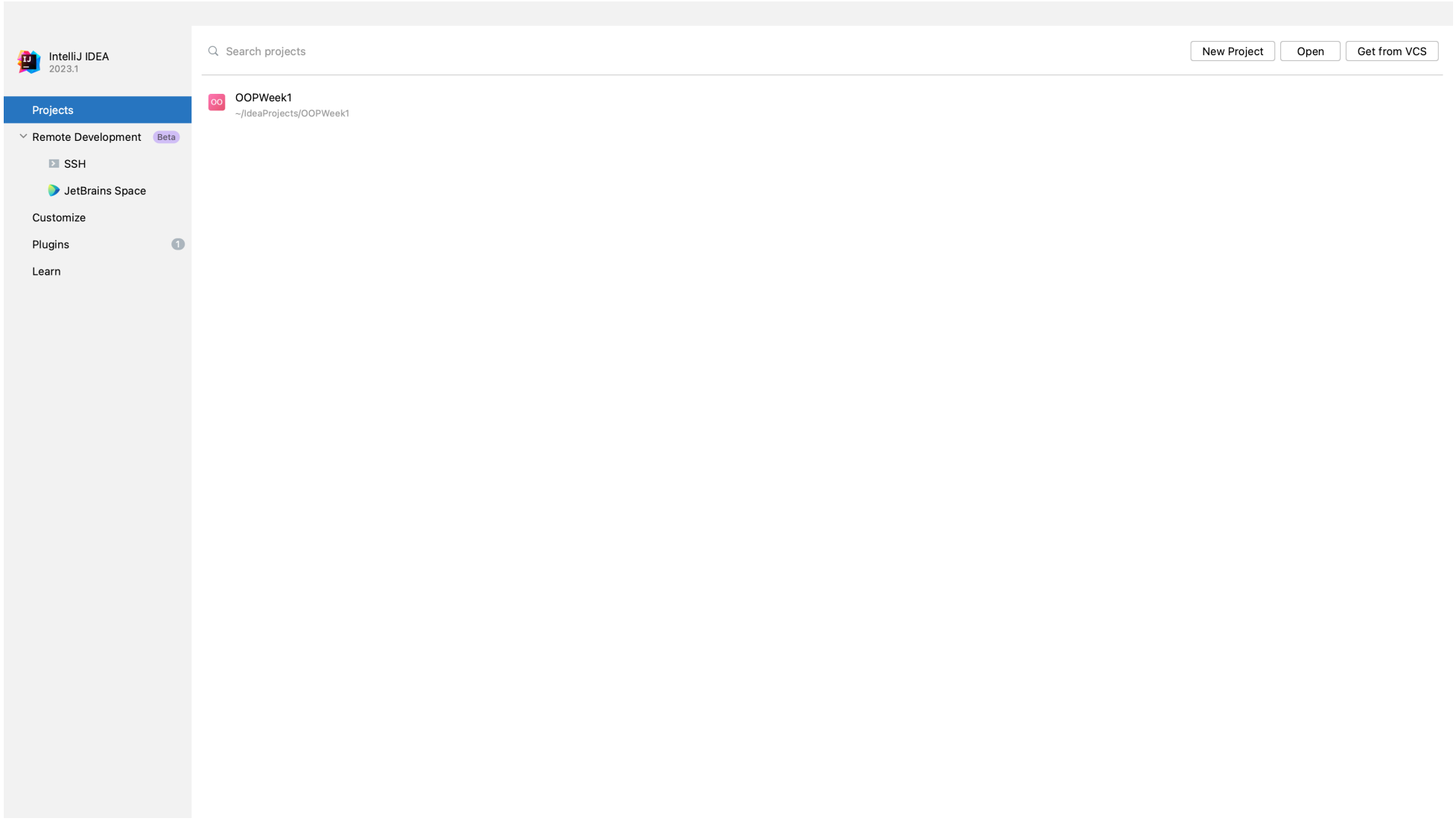
Software Technologies

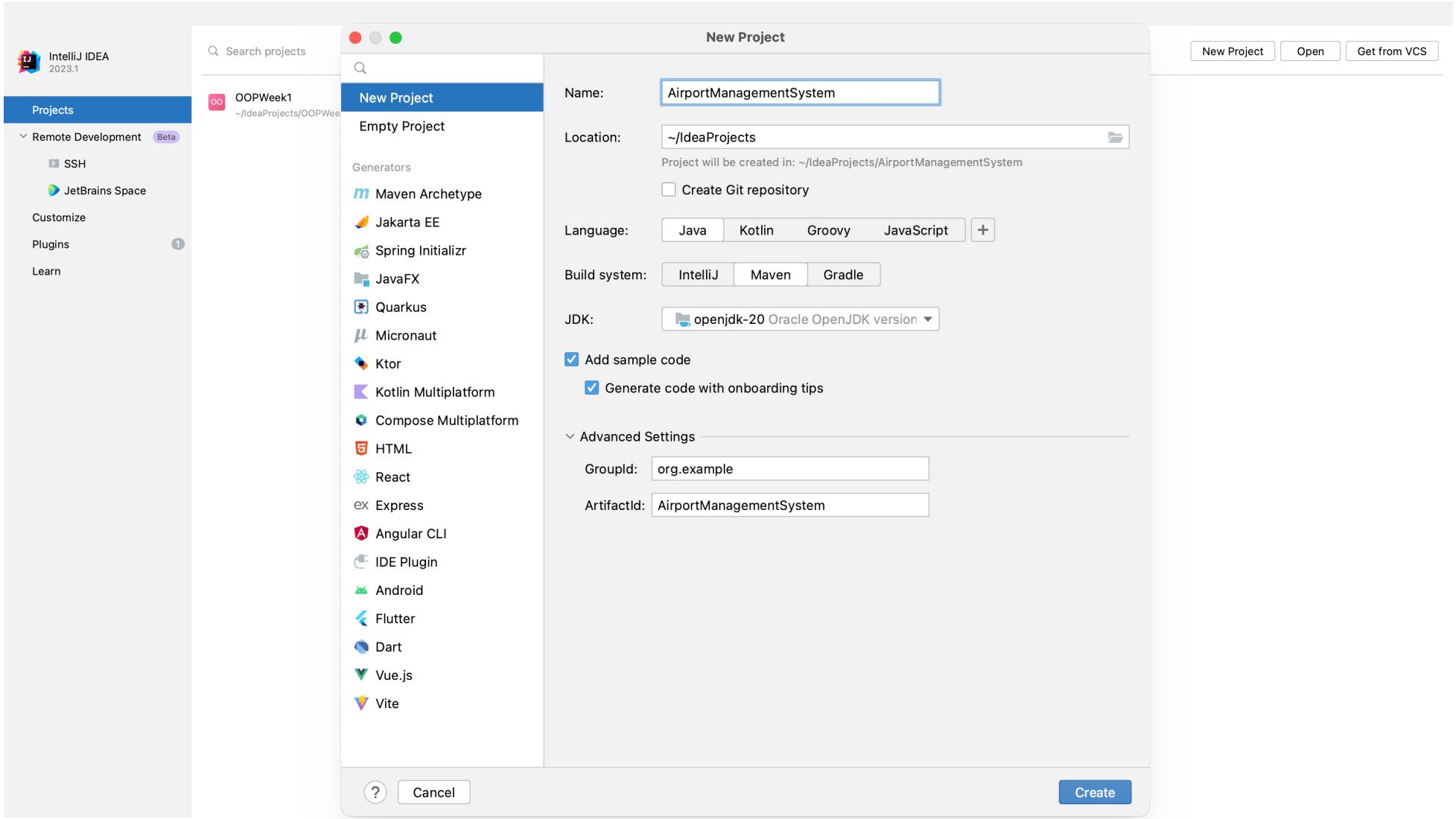
Software Technologies

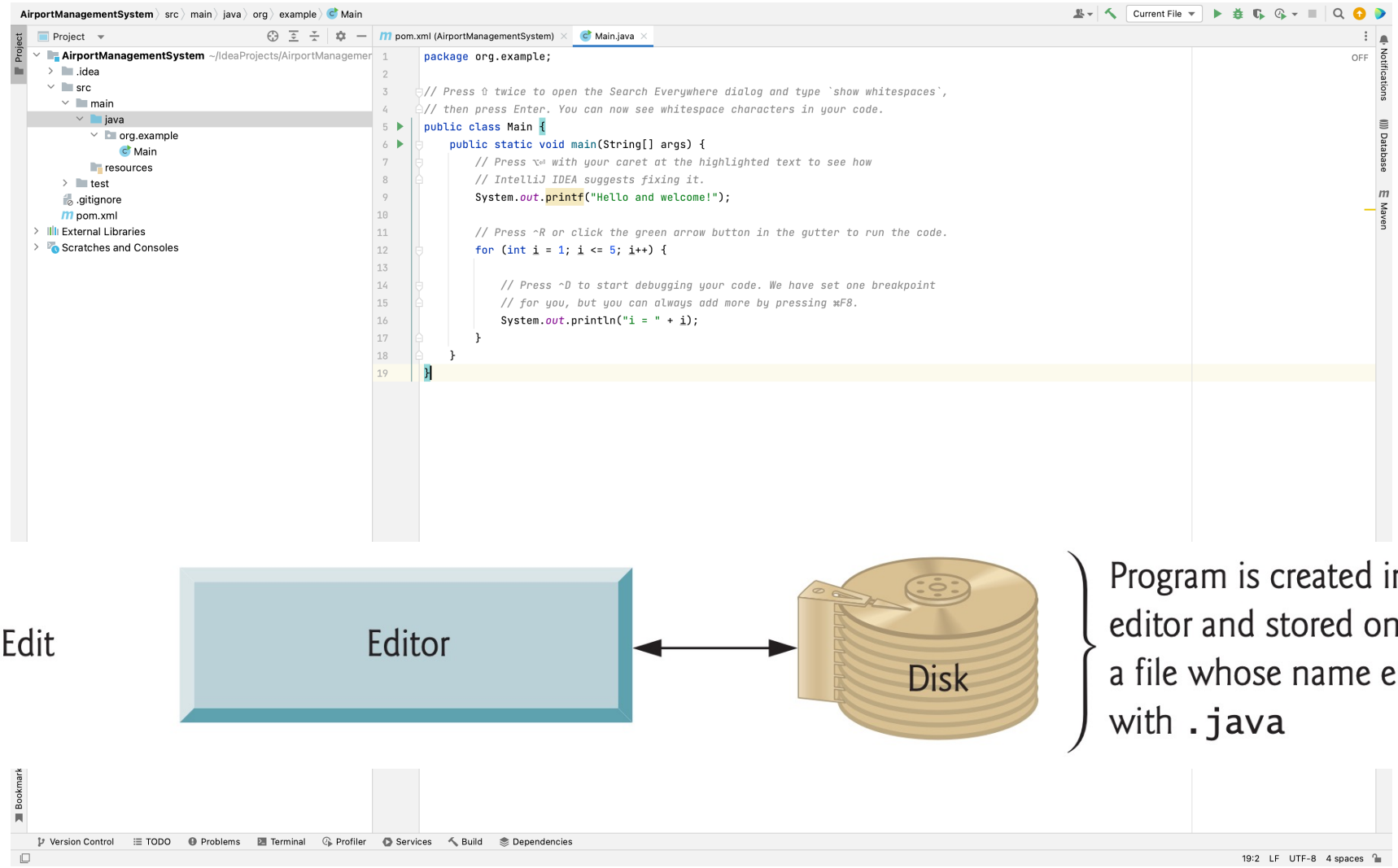
Software Technology	Description
Agile software	**Agile software development** is a set of methodologies that try to get soft-development ware implemented faster and using fewer resources.
Refactoring	**Refactoring** involves reworking programs to make them clearer and easier to maintain while preserving their correctness and functionality. Many IDEs contain built-in *refactoring tools* to do major portions of the reworking automatically.
Design patterns	**Design patterns** are proven architectures for constructing flexible and maintainable object-oriented software. The field of design patterns tries to enumerate those recurring patterns, encouraging software designers to *reuse* them to develop better-quality software using less time, money and effort.
LAMP	**LAMP** is an acronym for the open-source technologies that many developers use to build web applications—it stands for *Linux*, *Apache*, *MySQL* and *PHP* (or *Perl* or *Python*—two other scripting languages)

A Typical Java Development Environment

- Phase 1: Creating a Program
- Integrated development environments (IDEs) provide tools that support the software development process, such as editors, debuggers for locating logic errors and more.
- There are many popular Java IDEs, including:
  - Eclipse (www.eclipse.org)
  - NetBeans (www.netbeans.org)
  - IntelliJ IDEA (www.jetbrains.com)


IntelliJ IDEA
2023.1
Projects
Remote Development
Beta
SSH
JetBrains Space
Customize
Plugins
1
Learn
Search projects
New Project
Open
Get from VCS
OOPWeek1
~/IdeaProjects/OOPWeek1


IntelliJ IDEA
2023.1
Projects
Remote Development
Beta
SSH
JetBrains Space
Customize
Plugins
Learn
Search projects
OOPWeek1
~/IdeaProjects/OOPWee
New Project
Open
Get from VCS
New Project
New Project
Empty Project
Generators
Maven Archetype
Jakarta EE
Spring Initializr
JavaFX
Quarkus
Micronaut
Ktor
Kotlin Multiplatform
Compose Multiplatform
HTML
React
Express
Angular CLI
IDE Plugin
Android
Flutter
Dart
Vue.js
Vite
Name:
AirportManagementSystem
Location:
~/IdeaProjects
Project will be created in: ~/IdeaProjects/AirportManagementSystem
Create Git repository
Language:
Java
Kotlin
Groovy
JavaScript
Build system:
IntelliJ
Maven
Gradle
JDK:
openjdk-20 Oracle OpenJDK version
Add sample code
Generate code with onboarding tips
Advanced Settings
GroupId:
org.example
ArtifactId:
AirportManagementSystem
Cancel
Create

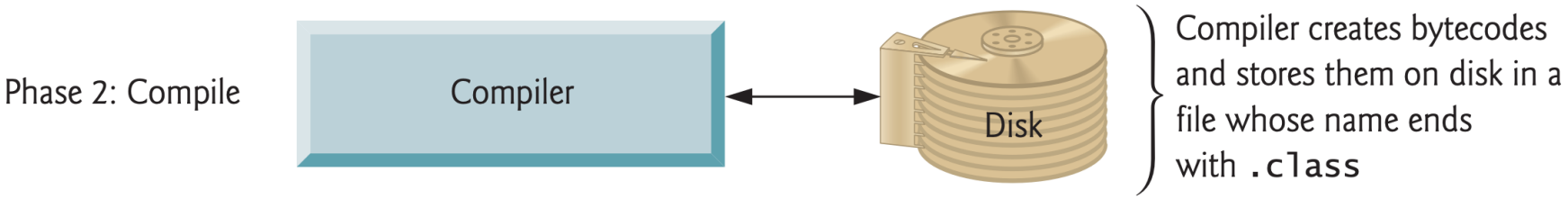


Phase 1: Edit

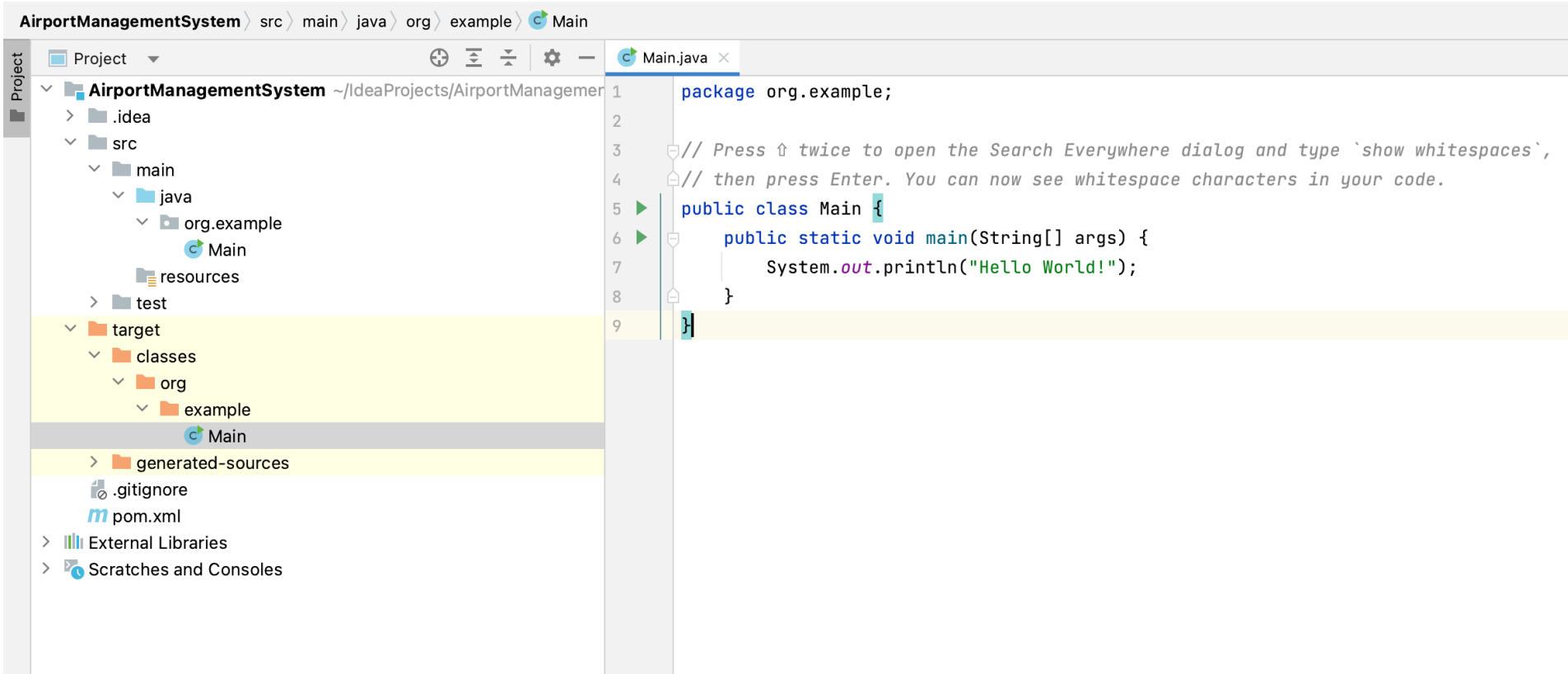
Program is created in an editor and stored on disk in a file whose name ends with `.java`

Compiling a Java Program into Bytecodes

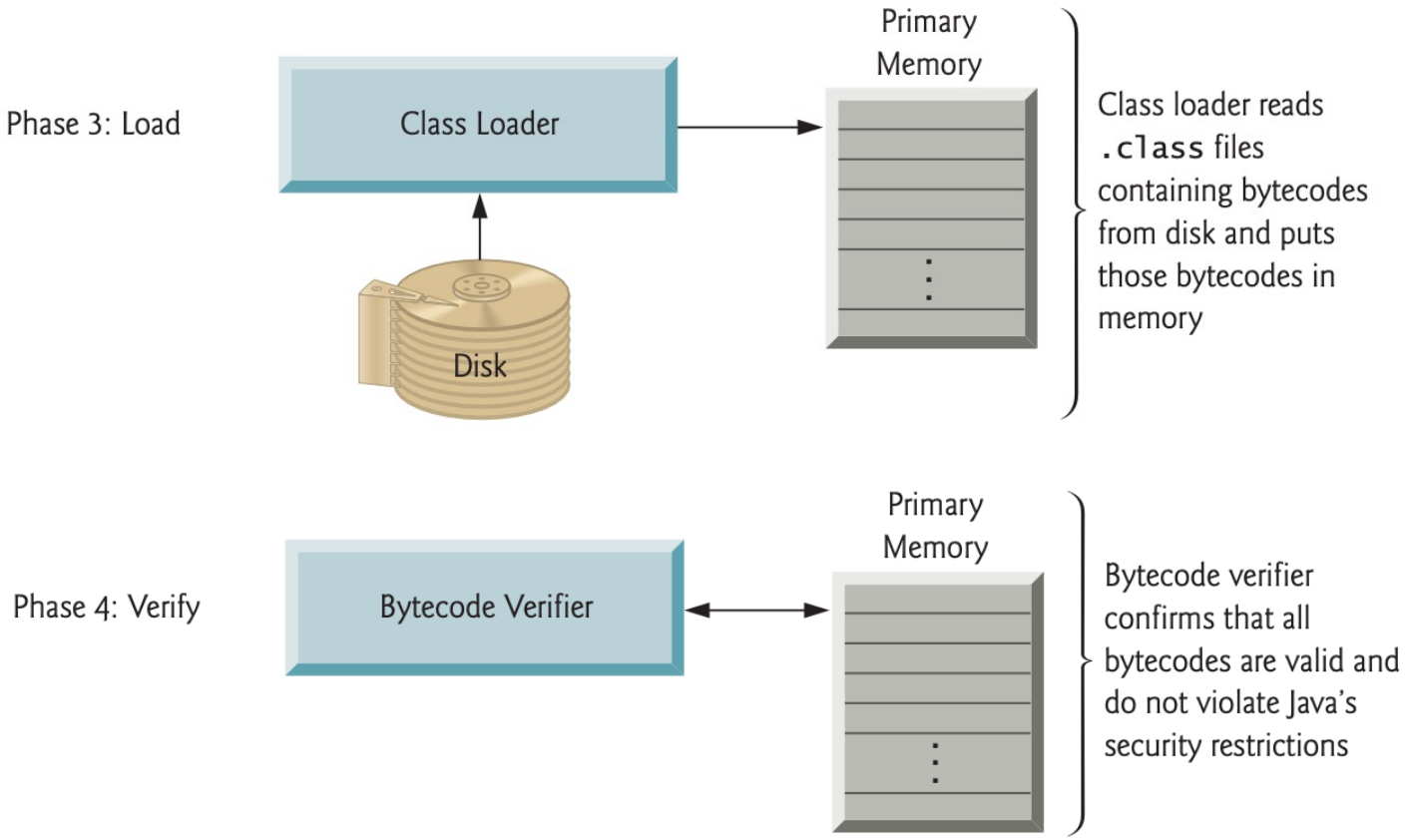
- The Java compiler translates Java source code into bytecodes that represent the tasks to execute in the execution phase
- Bytecode instructions are platform independent.



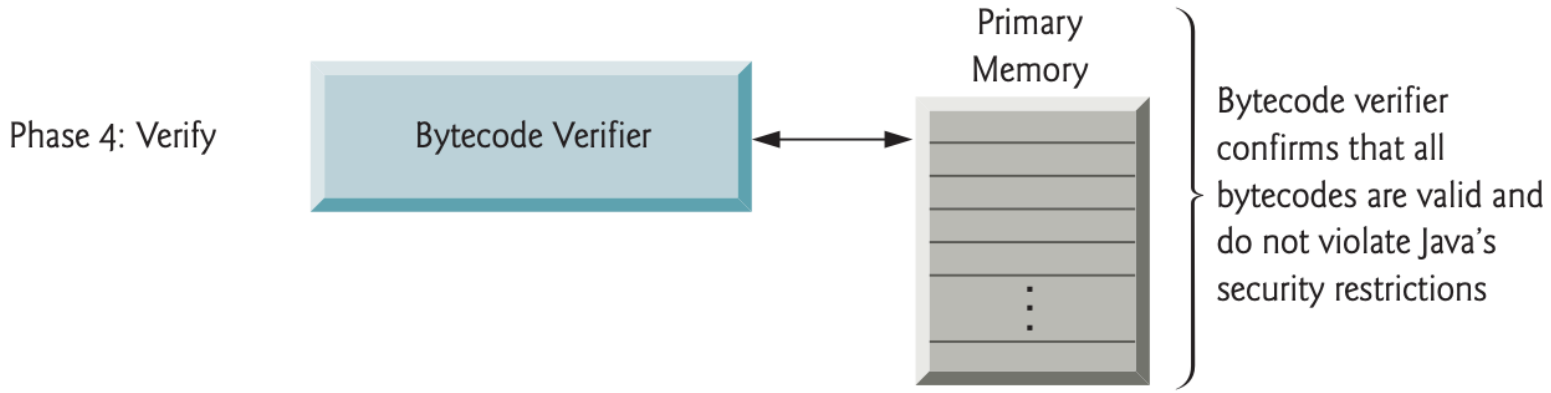
Compiling a Java Program into Bytecodes



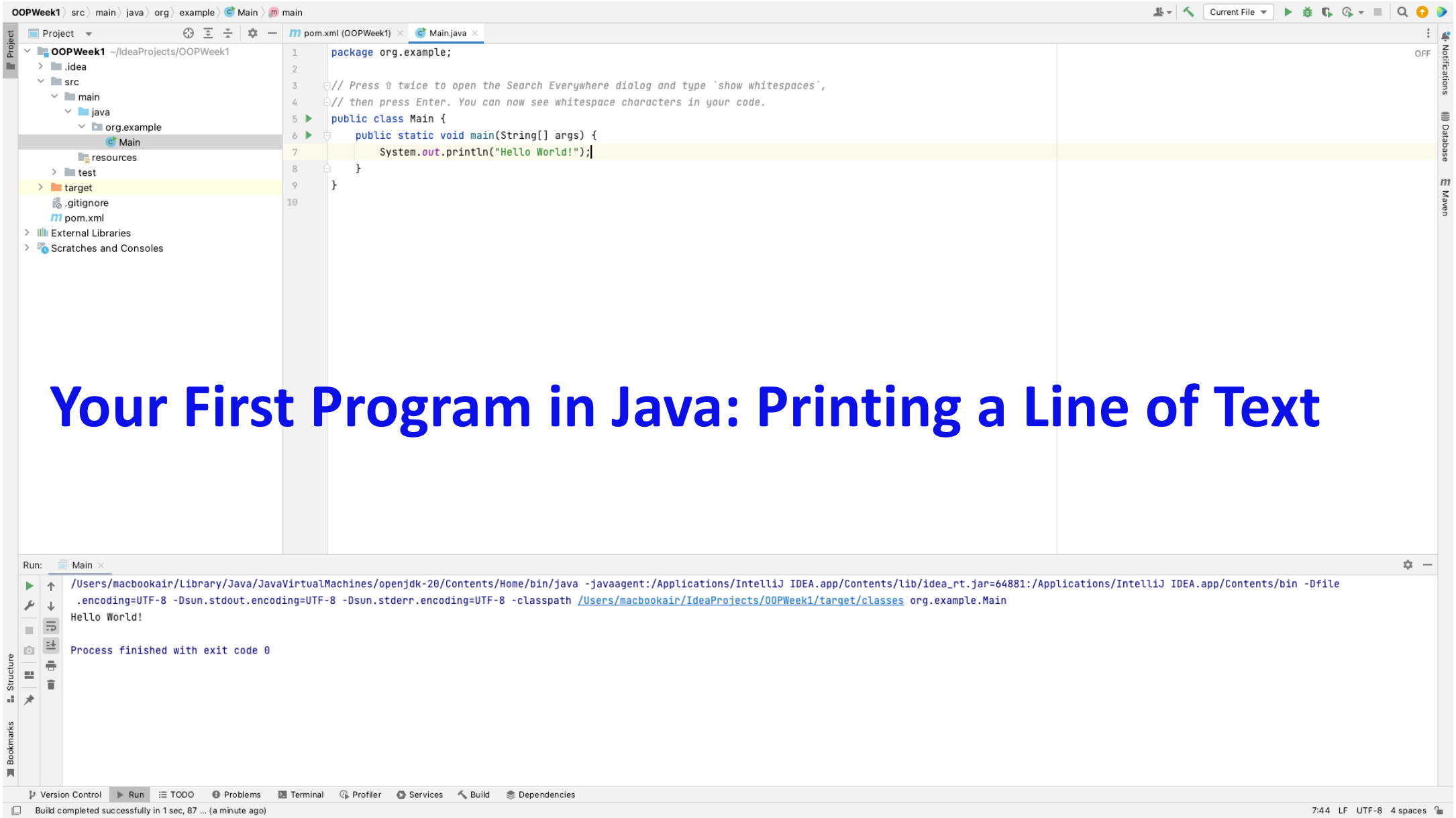
Compiling a Java Program into Bytecodes



Compiling a Java Program into Bytecodes




OOPWeek1 〉 src 〉 main 〉 java 〉 org 〉 example 〉 Main
Current File
Project
OOPWeek1 ~/IdeaProjects/OOPWeek1
.idea
src
main
java
org.example
Main
resources
test
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
pom.xml (OOPWeek1)
Main.java
OFF
1 package org.example;
2
3 // Press ⇧ twice to open the Search Everywhere dialog and type `show whitespaces`,
4 // then press Enter. You can now see whitespace characters in your code.
5 public class Main {
6 public static void main(String[] args) {
7 // Press ⌥⏎ with your caret at the highlighted text to see how
8 // IntelliJ IDEA suggests fixing it.
9 System.out.printf("Hello and welcome!");
10
11 // Press ^R or click the green arrow button in the gutter to run the code.
12 for (int i = 1; i <= 5; i++) {
13
14 // Press ^D to start debugging your code. We have set one breakpoint
15 // for you, but you can always add more by pressing ⌘F8.
16 System.out.println("i = " + i);
17 }
18 }
19 }
20
Run: Main
/Users/macbookair/Library/Java/JavaVirtualMachines/openjdk-20/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=64866:/Applications/IntelliJ IDEA.app/Contents/bin -Dfile
.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/macbookair/IdeaProjects/OOPWeek1/target/classes org.example.Main
Hello and welcome!i = 1
i = 2
i = 3
i = 4
i = 5
Process finished with exit code 0
Version Control
Run
TODO
Problems
Terminal
Profiler
Services
Build
Dependencies
Build completed successfully in 1 sec, 443 ... (a minute ago)
LF
UTF-8
4 spaces
Notifications
Database
Maven
Structure
Bookmarks



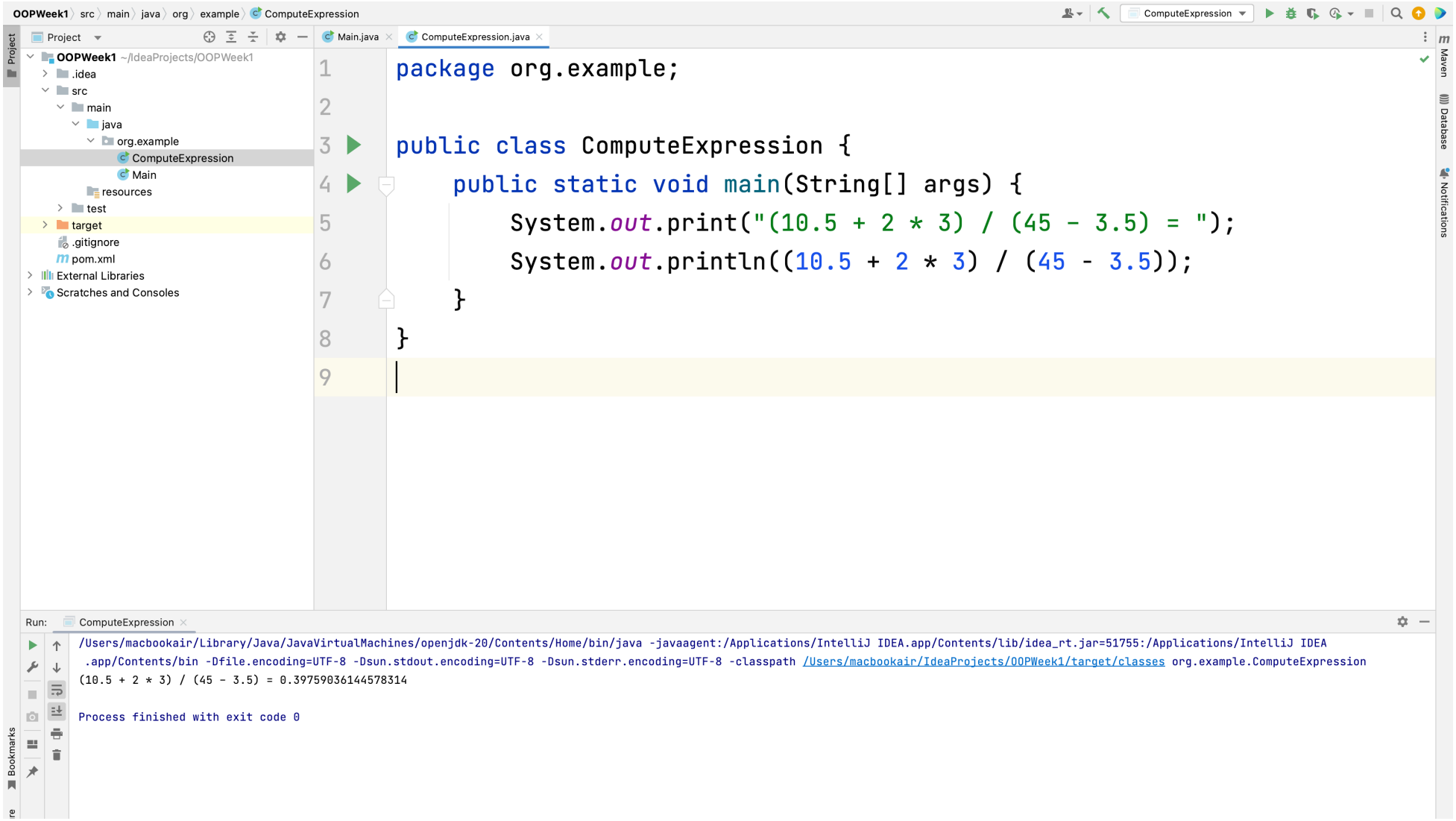
Your First Program in Java: Printing a Line of Text

Program Structure

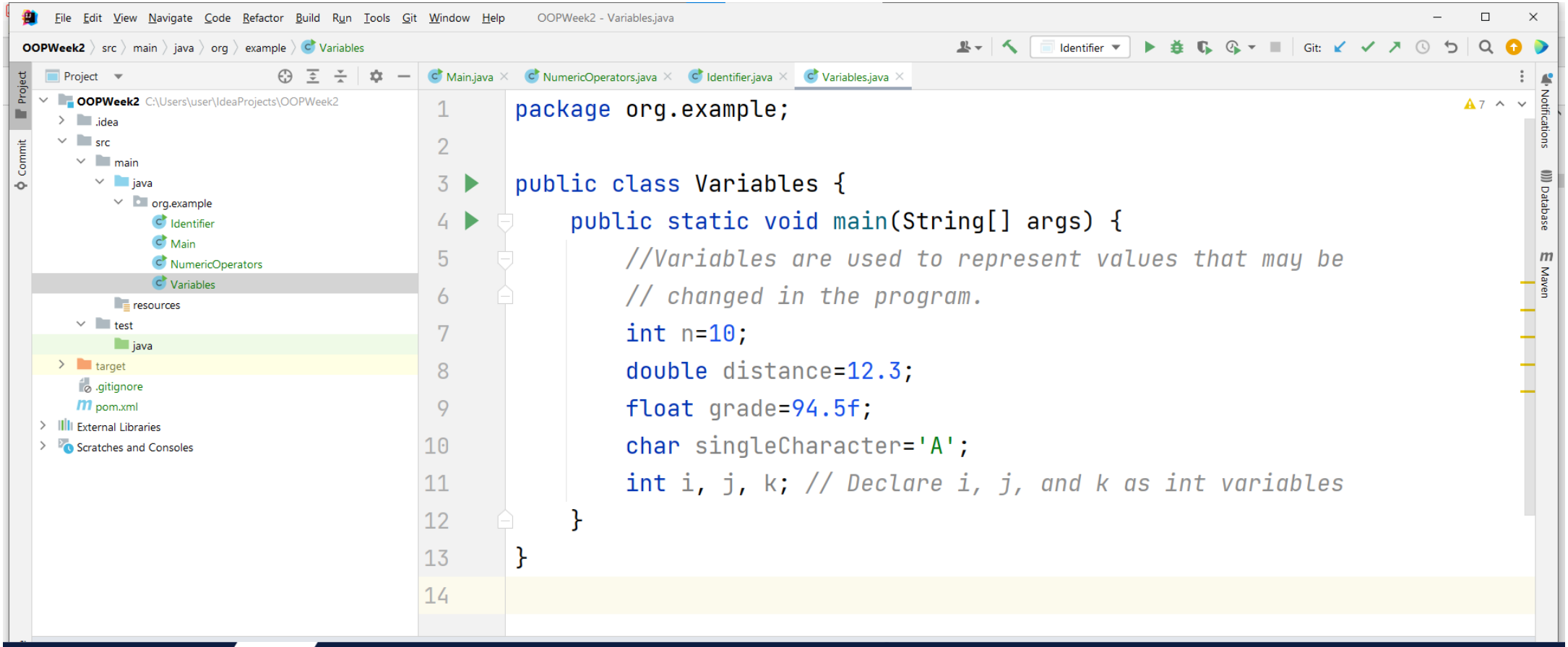
```
class CLASSNAME {
    public static void main(String[] args) {
        //STATEMENTS
    }
}
```

Compute Expression

$$\frac{10.5 + 2 \times 3}{45 - 3.5}$$


OOPWeek1 src main java org example ComputeExpression
ComputeExpression
Project
Main.java
ComputeExpression.java
OOPWeek1 ~/IdeaProjects/OOPWeek1
.idea
src
main
java
org.example
ComputeExpression
Main
resources
test
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
1 package org.example;
2
3 public class ComputeExpression {
4 public static void main(String[] args) {
5 System.out.print("(10.5 + 2 * 3) / (45 - 3.5) = ");
6 System.out.println((10.5 + 2 * 3) / (45 - 3.5));
7 }
8 }
9
Run: ComputeExpression
/Users/macbookair/Library/Java/JavaVirtualMachines/openjdk-20/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=51755:/Applications/IntelliJ IDEA
.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/macbookair/IdeaProjects/OOPWeek1/target/classes org.example.ComputeExpression
(10.5 + 2 * 3) / (45 - 3.5) = 0.39759036144578314
Process finished with exit code 0
Maven
Database
Notifications
Bookmarks

Variables


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
OOPWeek2 - Variables.java
OOPWeek2 src main java org example Variables
Identifier
Git:
Project
Main.java
NumericOperators.java
Identifier.java
Variables.java
OOPWeek2 C:\Users\user\IdeaProjects\OOPWeek2
.idea
src
main
java
org.example
Identifier
Main
NumericOperators
Variables
resources
test
java
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
1 package org.example;
2
3 public class Variables {
4 public static void main(String[] args) {
5 //Variables are used to represent values that may be
6 // changed in the program.
7 int n=10;
8 double distance=12.3;
9 float grade=94.5f;
10 char singleCharacter='A';
11 int i, j, k; // Declare i, j, and k as int variables
12 }
13 }
14
Notifications
Database
Maven

Named Constants

- final datatype CONSTANTNAME = value;

```
public class NamedConstants {
    public static void main(String[] args) {
        //A named constant is an identifier that represents a permanent value.
        final double pi=3.14;
    }
}
```

Numeric Data Types and Operations

Name	*Range*	*Storage Size*	
`byte`	-2^7 to $2^7 - 1$ (-128 to 127)	8-bit signed	byte type
`short`	-2^{15} to $2^{15} - 1$ (-32768 to 32767)	16-bit signed	short type
`int`	-2^{31} to $2^{31} - 1$ (-2147483648 to 2147483647)	32-bit signed	int type
`long`	-2^{63} to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed	long type
`float`	Negative range: $-3.4028235E + 38$ to $-1.4E - 45$ Positive range: $1.4E - 45$ to $3.4028235E+38$	32-bit IEEE 754	float type
`double`	Negative range: $-1.7976931348623157E+308$ to $-4.9E - 324$ Positive range: $4.9E - 324$ to $1.7976931348623157E+308$	64-bit IEEE 754	double type

Numeric Operators

Name	*Meaning*	*Example*	*Result*
+	Addition	34 + 1	35
-	Subtraction	34.0 − 0.1	33.9
*	Multiplication	300*30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

Numeric Operators

```java
public class ArithmeticOperators {
    public static void main(String[] args) {
        double num1 = 10;
        double num2 = 3;
        double res1 = num1 / num2;
        double res2 = num1 * num2;
        double res3 = num1 - num2;
        double res4 = num1 + num2;
        System.out.println(res1); // 3.3333
        System.out.println(res2);  // 30.0
        System.out.println(res3); // 7.00
        System.out.println(res4); // 13.00
        double res5 = num1 % num2; // 1.0
        System.out.println(res5);
    }
}
```

Bitwise Operators

- Bitwise AND (&)
- Bitwise OR (|)
- Bitwise XOR (^)

```java
class BitwiseOperators {
    public static void main(String[] args) {
        int a = 5;
        int b = 7;
        int res1  = a & b ;
        int res2 = a | b;
        int res3 = a ^ b;
         System.out.println(res1); // 5
         System.out.println(res2); // 7
         System.out.println(res3); // 2
    }
}
```

```java
 int[] arr = {3, 3, 2, 8, 8, 2, 7};

static int oddNumber(int[] arr){
    int result =0;
    for (int i =0; i < arr.length; i++){
        result ^= arr[i];
    }
    return result;
}
```

Reading Input from the Console

Reading Input from the console enables program the accept from the user.

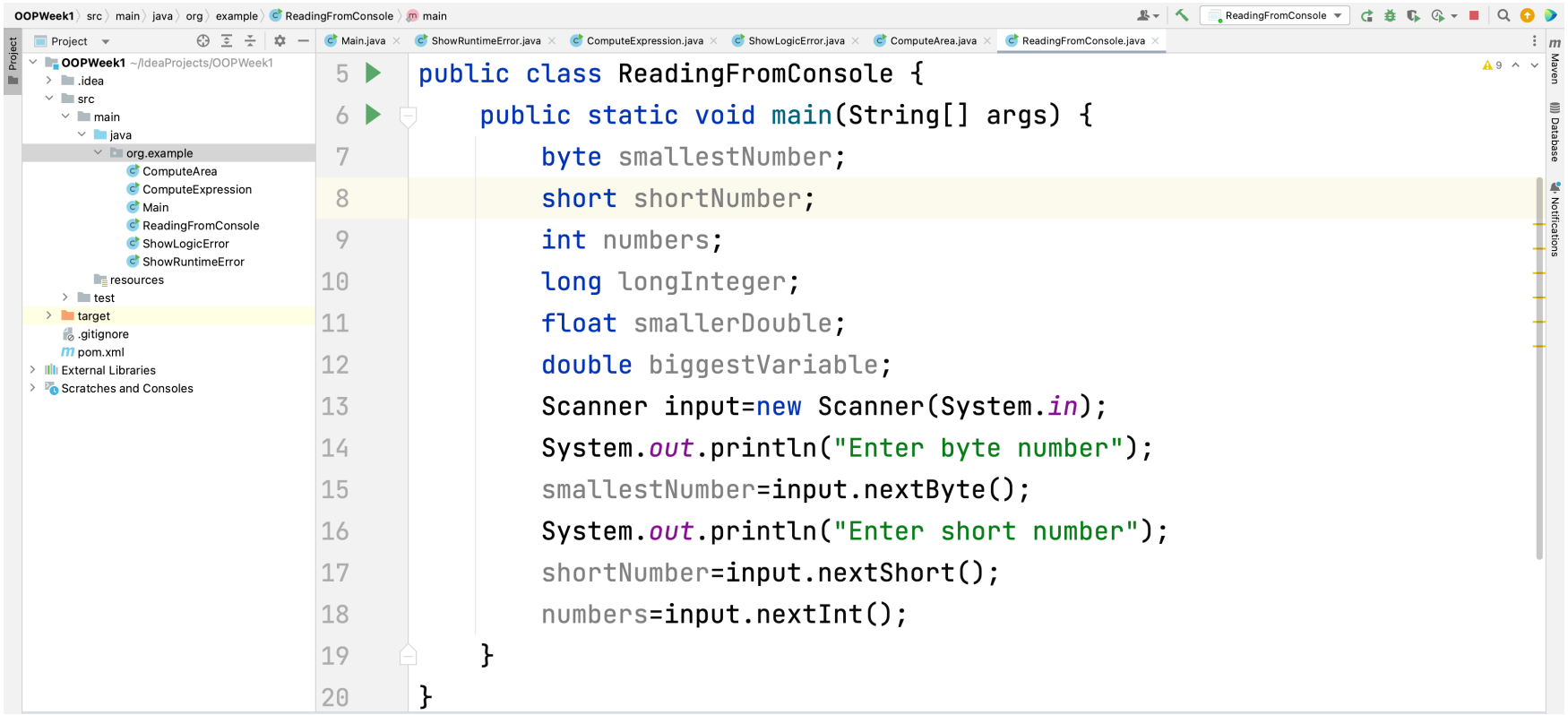
Reading Input from the Console

```
public class ReadingFromConsole {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        String message = input.next();

        System.out.println(message);
    }
}
```

Reading Numbers from the Keyboard

Method	*Description*
`nextByte()`	reads an integer of the `byte` type.
`nextShort()`	reads an integer of the `short` type.
`nextInt()`	reads an integer of the `int` type.
`nextLong()`	reads an integer of the `long` type.
`nextFloat()`	reads a number of the `float` type.
`nextDouble()`	reads a number of the `double` type.

Reading Numbers from the Keyboard



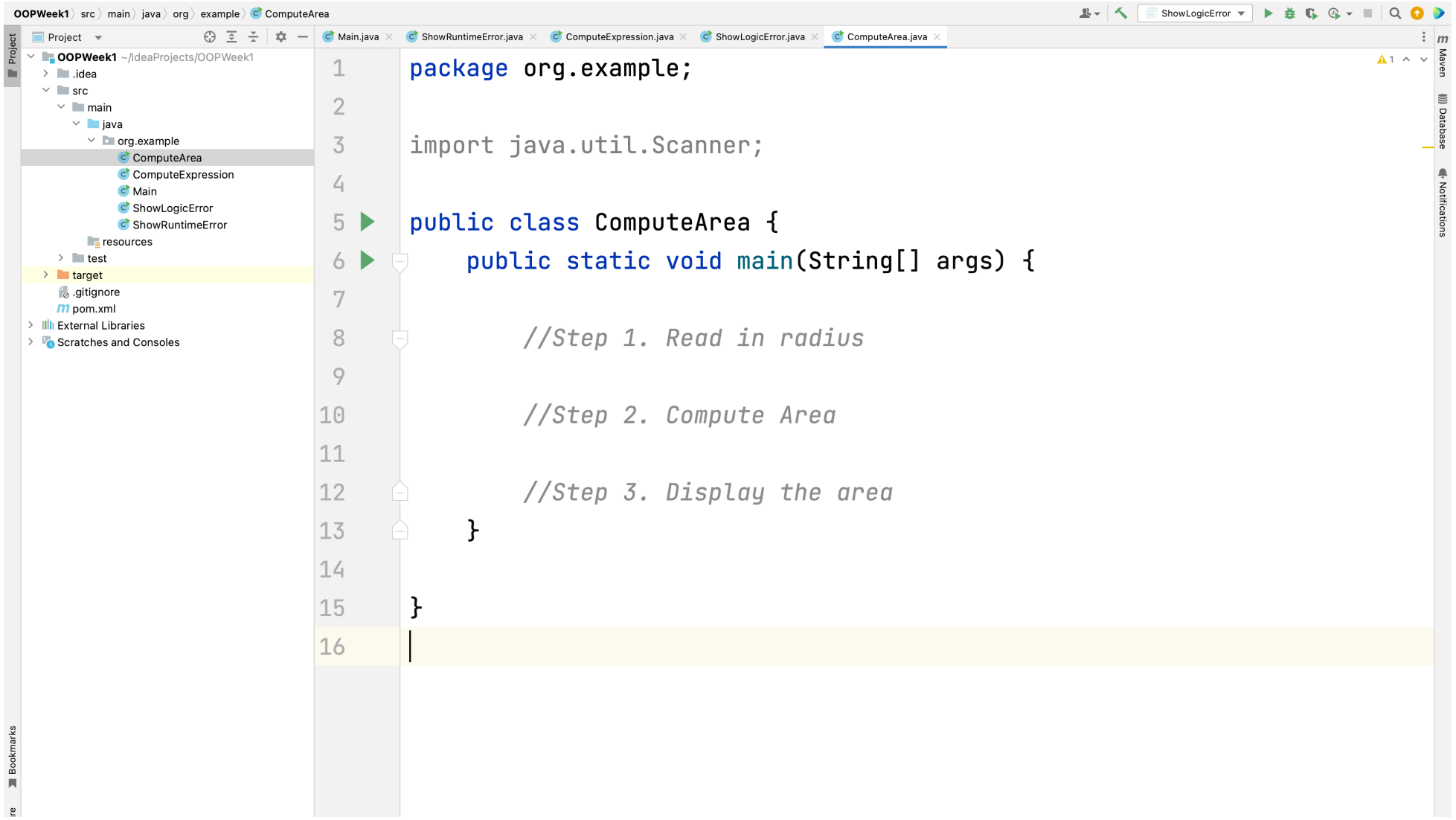
Note

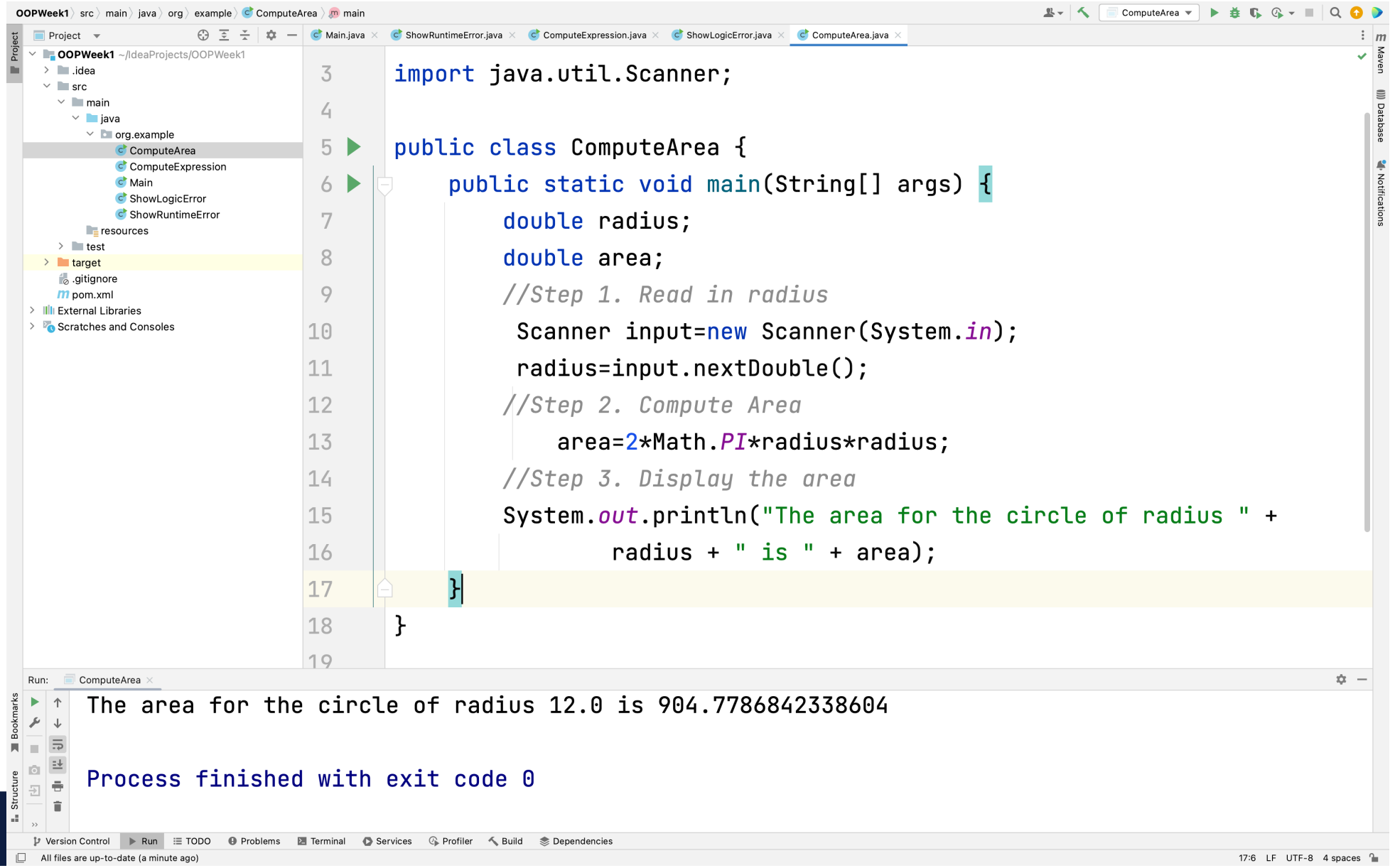
- To improve readability, Java allows you to use underscores between two digits in a number literal. For example, the following literals are correct.
- int ssn = 232_45_4519;
- However, 45_ or _45 is incorrect. The underscore must be placed between two digits.

Exponent Operations

- The Math.pow(a, b) method can be used to compute a^b.
- The pow method is defined in the Math class in the Java API.
- You invoke the method using the syntax Math.pow(a, b) (e.g., Math.pow(2, 3)), which returns the result of ab (2^3).

```
System.out.println(Math.pow(2, 3)); // Displays 8.0
System.out.println(Math.pow(4, 0.5)); // Displays 2.0
System.out.println(Math.pow(2.5, 2)); // Displays 6.25
System.out.println(Math.pow(2.5, -2)); // Displays 0.16
```


OOPWeek1 src main java org example ComputeArea
ShowLogicError
Project
Main.java
ShowRuntimeError.java
ComputeExpression.java
ShowLogicError.java
ComputeArea.java
OOPWeek1 ~/IdeaProjects/OOPWeek1
.idea
src
main
java
org.example
ComputeArea
ComputeExpression
Main
ShowLogicError
ShowRuntimeError
resources
test
target
.gitignore
pom.xml
External Libraries
Scratches and Consoles
1 package org.example;
2
3 import java.util.Scanner;
4
5 public class ComputeArea {
6 public static void main(String[] args) {
7
8 //Step 1. Read in radius
9
10 //Step 2. Compute Area
11
12 //Step 3. Display the area
13 }
14
15 }
16
Maven
Database
Notifications
Bookmarks



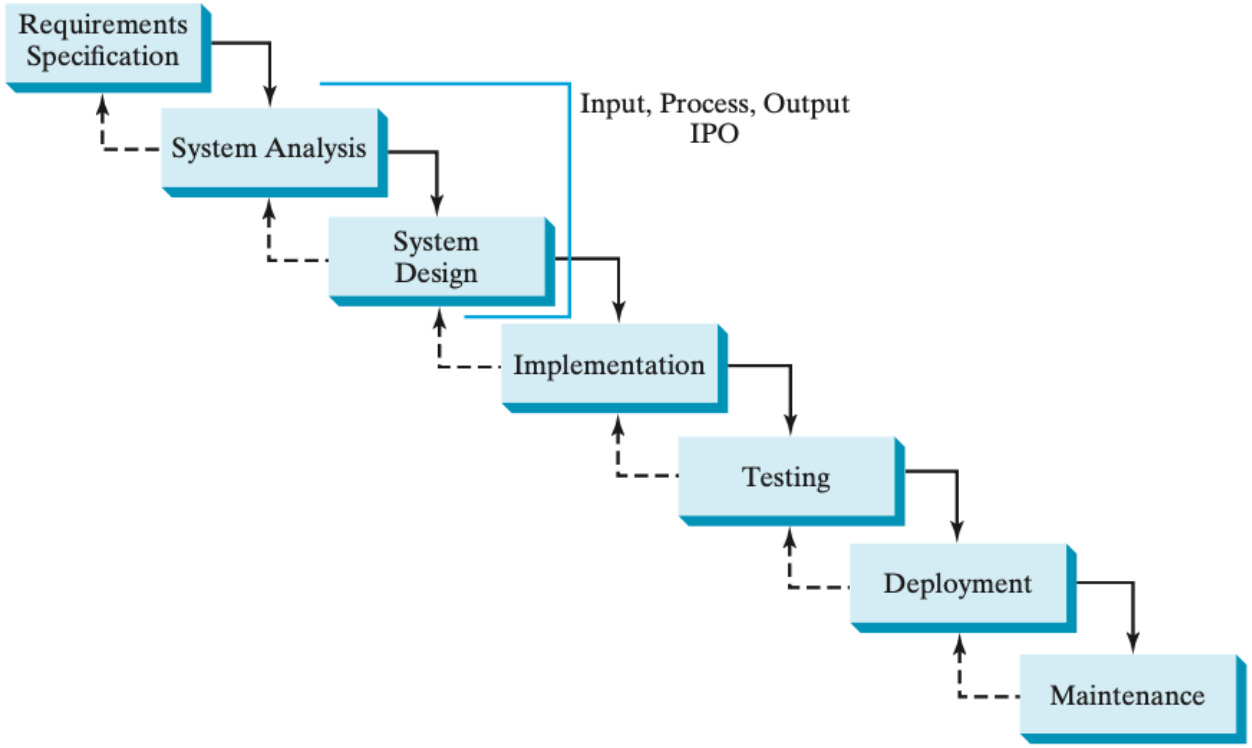
New Uzbekistan University

```
 Scanner input=new Scanner(System.in);
 radius=input.nextDouble();
//Step 2. Compute Area
    area=2*Math.PI*radius*radius;
//Step 3. Display the area
System.out.println("The area for the circle of radius " +
        radius + " is " + area);
System.out.printf("The area for the circle of radius %.2f is %.2f",radius,area);
```

Software Development Process

- The software development life cycle is a multistage process that includes requirements specification, analysis, design, implementation, testing, deployment, and maintenance.

Software Development Process




ANY
QUESTIONS