Secure OOP with Java

Lecture - Unit 01

Claudia Maderthaner <claudia.maderthaner@fh-hagenberg.at>

Programming Paradigms

Programming Paradigm is a style, technique, or way of writing a program.

Types of Programming Paradigms

- Imperative
  - Procedural
  - Object-oriented
- Declarative
  - Functional
  - Logic

Imperative

- Focus on the **how**
- Specify every step to reach a desired state

Procedural

- Combine multiple commands to a **procedure**
- Procedures are executed as a unit
- A procedure has side effects

Object-Oriented

- Organize programs as interacting objects
- An object encapsulates data and algorithms

Declarative

- Focus on the **what**
- Description of a desired state

Functional

- Treat programs as mathematical functions
- Avoids states and mutable data
- A function has no side effects

Logic

- A program consists of a set of axioms and a goal statement
- Mathematical concept called a relation from a set theory

Java

Characteristics

- General-purpose programming language
- Supports
  - **object-oriented**
  - procedural, and
  - functional paradigms

JAVA is a programming language and environment that was designed to solve a number of problems in modern programming practice.

Java: an Overview (1995)
— James Gosling

Design Goals

- Simple
- Object-Oriented
- Distributed
- Robust
- Secure
- Architecture Neutral
- Portable
- Interpreted
- High Performance
- Multithreaded
- Dynamic

History

1996

JDK 1.0

- First stable version of Java
  - Java Virtual Machine
  - Java Compiler
  - Class Libraries

1997

JDK 1.1

- Inner classes
- JavaBeans
- Java Database Connectivity (JDBC)
- Java remote method invocation (Java RMI) and Serialization
- Reflection

1998

J2SE 1.2

- **Collections Framework**
- Swing
- `strictfp`
- Just-in-Time compiler (JIT)
- Java Interface Definition Language (Java IDL)

2000

J2SE 1.3

- HotSpot JVM
- Java Naming and Directory Interface (JNDI)
- Java Platform Debugger Architecture (JPDA)

2002

J2SE 1.4

- `assert` keyword
- Regular expressions
- Non-blocking I/O (NIO)
- Logging API
- Integrated security and cryptography extensions (JCE, JSSE, JAAS)
- Java Web Start

2004

J2SE 5.0

- Generics
- Annotations
- Enumerations
- Autoboxing/unboxing
- Varargs
- Enhanced for-each loop
- Static imports
- Concurrency utilities

2006

Java SE 6

- Scripting Language Support
- JAX-WS (Java API for XML Web Services)
- JDBC 4.0
- Java Compiler API
- Synchronization and compiler performance optimization

Java was released as free and open-source software.

2009/2010

Sun was sold to Oracle Cooperation.

2011

Java SE 7

- JVM support for dynamic languages (`invokedynamic`)
- Project **Coin**
  - Strings in `switch`
  - Automatic resources management in `try`-statement
  - Improved type inference for generic instance creation (diamond operator <>)
  - Binary integer literals
  - Allowing underscores in numeric literals
  - Catching multiple exception types
- New I/O

2014

Java SE 8 (LTS)

- Lambda expressions
- Date and time API
- Annotation on Java Type
- Repeating Annotations
- Unsigned integer arithmetic

2017

Java SE 9

- **Java Platform Module System** (Project **Jigsaw**)
- `jshell` (Read-Eval-Print Loop)
- Compact Strings
- `jlink` (The Java Linker)
- Ahead-of-Time Compilation

2018

Java SE 10 - März 2018

- Local-Variable Type Inference (`var`)
- Garbage-Collector Interface
- Parallel Full GC for G1
- Thread-Local Handshakes
- Heap Allocation on Alternative Memory Devices
- Root Certificates

Java SE 11 (LTS)

- Launch Single-File Source-Code Programs
- HTTP Client
- Unicode 10
- Flight Recorder
- Epsilon: A No-Op Garbage Collector
- Transport Layer Security (TLS) 1.3

2019

Java SE 12

- JVM Constants API
- Switch Expressions (Preview)
- Shenondoah: A Low-Pause-Time Garbage Collector (Experimental)

Java SE 13

- Switch Expressions (Preview)
- Text Blocks (Preview)

2020

Java SE 14

- Switch Expressions
- Text Block (Second Preview)
- Pattern Matching for `instanceof` (Preview)
- Records (Preview)
- Helpful NullPointerExceptions

Java SE 15

- Text Blocks
- Hidden Classes
- Sealed Classes (Preview)
- Records (Second Preview)
- Pattern Matching for `instanceof` (Second Preview)
- Edwards-Curve Digital Signature Algorithm (EdDSA)
- Shenondoah: A Low-Pause-Time Garbage Collector

2021

Java SE 16

- Records
- Sealed Classes (Second Preview)
- Pattern Matching for `instanceof`
- Strongly Encapsulate JDK internals by Default
- Packaging Tool

Java SE 17 (LTS)

- Sealed Classes
- Enhanced Pseudo-Random Number Generators
- Pattern Matching for `switch` (Preview)
- Deprecate the Security Manager for Removal

2022

Java SE 18

- UTF-8 by Default
- Code Snippets in the Java API Documentation
- Pattern Matching for `switch` (Second Preview)
- Deprecate Finalization for Removal

Java SE 19

- Record Patterns (Preview)
- Virtual Threads (Preview)
- Pattern Matching for `switch` (Third Preview)

2023

Java SE 20

- Record Patterns (Second Preview)
- Virtual Threads (Second Preview)
- Pattern Matching for `switch` (Fourth Preview)

Java SE 21 (LTS)

→ tbd

Java Community Process (JCP)

- Established in 1998
- Formalized mechanism to develop standard technical specifications for Java

Java Specification Requests (JSRs)

- Describe proposed specifications and technologies

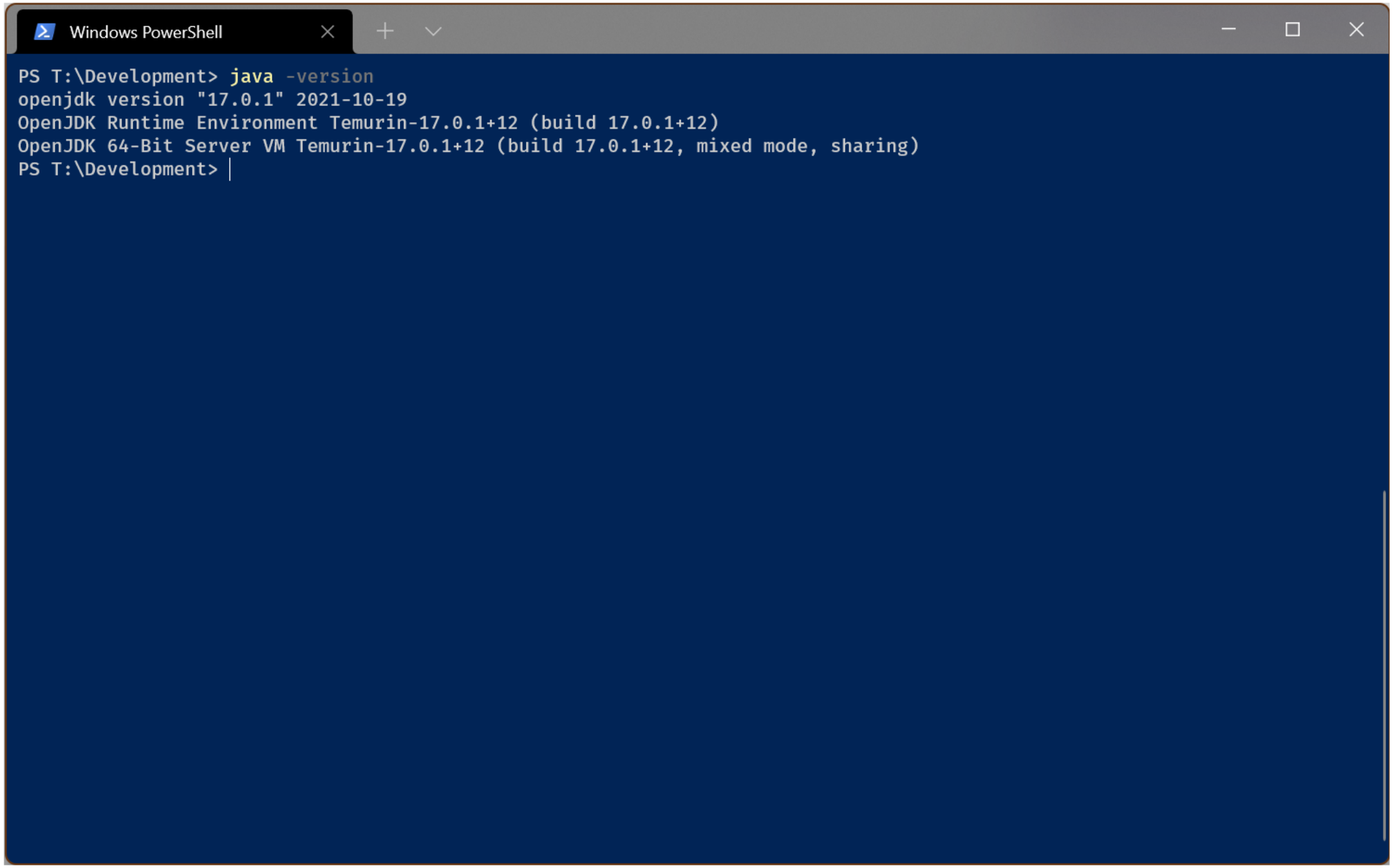
1. Initiation
2. Draft Releases
3. Final Release
4. Maintenance

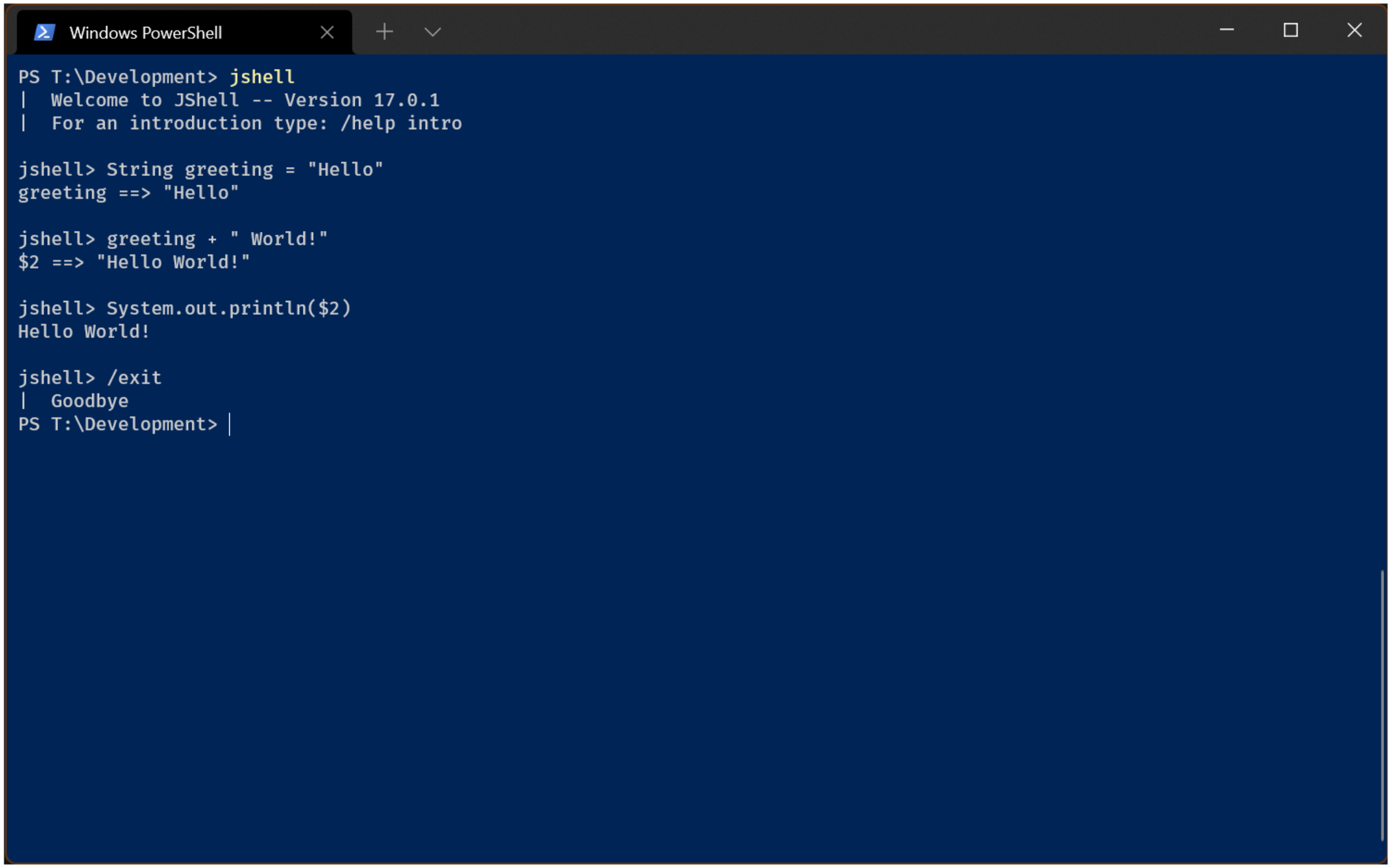
Java Development Kit

- Oracle Java SE
- OpenJDK
- OpenJDK Builds
  - Adoptium Temurin
  - RedHat build of OpenJDK
  - Amazon Coretto
  - Azul Zulu

Installation Notes

- Download JDK archive
- Extract to folder, e. g. `C:\Program Files\Java\`
- Set or update `JAVA_HOME`
- Set or update `PATH`


Windows PowerShell
PS T:\Development> java -version
openjdk version "17.0.1" 2021-10-19
OpenJDK Runtime Environment Temurin-17.0.1+12 (build 17.0.1+12)
OpenJDK 64-Bit Server VM Temurin-17.0.1+12 (build 17.0.1+12, mixed mode, sharing)
PS T:\Development>


Windows PowerShell
PS T:\Development> jshell
| Welcome to JShell -- Version 17.0.1
| For an introduction type: /help intro
jshell> String greeting = "Hello"
greeting ==> "Hello"
jshell> greeting + " World!"
$2 ==> "Hello World!"
jshell> System.out.println($2)
Hello World!
jshell> /exit
| Goodbye
PS T:\Development>
PS T:\Development>

Java Virtual Machine (JVM)

Write once - run anywhere

Writing Java Programs

A GEEK IS BORN

Hello World

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3     System.out.println("Hello World!");
4     }
5 }
```

```java
package hello;

/**
 * A first java class.
 *
 * @author Claudia Maderthaner
 */
public class HelloWorld {

    /**
     * The main application method.
     * <p>
     * This is the first method which is executed.
     *
     * @param args no used
     */
    public static void main(String[] args) {
    System.out.println(sayHello());
    }

    /**
     * This method returns the application message.
     *
     * @return the greeting message
     */
    private static String sayHello() {
    return "Hello World!";
    }

}
```

main-Method

- is called `main`
- declared `public` and `static`,
- returns `void`,
- has a argument as `String` array

```
public static void main(String[] args) {
    // ...
}
```

Contact

Moodle Discussion Board

claudia.maderthaner@fh-hagenberg.at