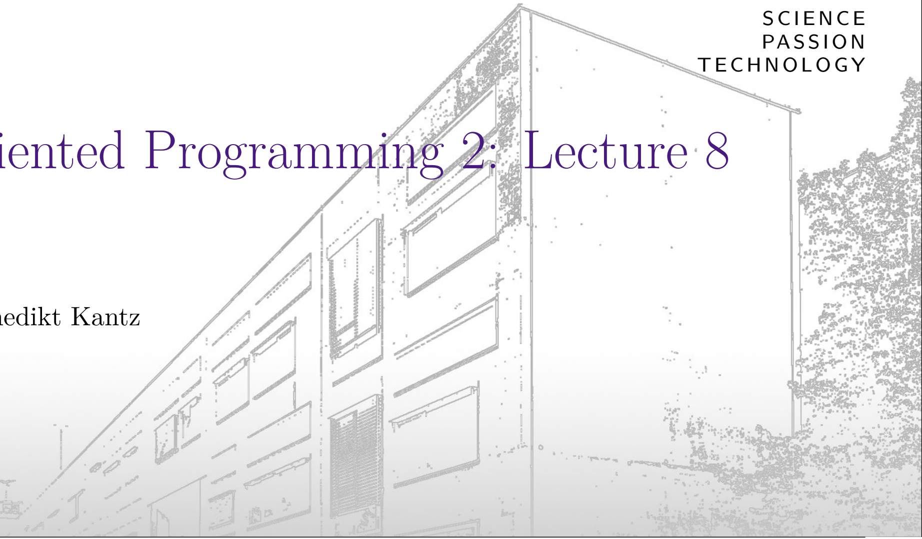
# Object-Oriented Programming 2: Lecture 8
# Libraries

Tobias Schreck, Benedikt Kantz



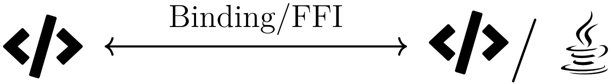
SCIENCE
PASSION
TECHNOLOGY

# Which libraries have you used for your studies?

⟹ ORMs, Machine Learning, Bindings, . . . . . .

# Bindings

## What?[1]



[1] The name is interchangeably used with Foreign Function Interfaces in Rust and Python

# Bindings

## What?[1]

- (Semi-)Automatic translation layer between langauges
- Similar to ORMs, again: impedance mismatches, memory management
- Gotchas:
  - **Marshalling**: conversion of data/object for calls ($\neq$ serialization)
  - **Memory Management**: Java/Python have Garbage Collectors (GCs)[1], C/C++ not - communicate lifetime events!

[1] Java Docs

[1] The name is interchangeably used with Foreign Function Interfaces in Rust and Python

# Bindings

## What?[1]

- (Semi-)Automatic translation layer between langauges
- Similar to ORMs, again: impedance mismatches, memory management
- Gotchas:
    - **Marshalling**: conversion of data/object for calls ($\neq$ serialization)
    - **Memory Management**: Java/Python have Garbage Collectors (GCs)[1], C/C++ not - communicate lifetime events!

[1] Java Docs

[1] The name is interchangeably used with Foreign Function Interfaces in Rust and Python

# Bindings

## What?[1]

- (Semi-)Automatic translation layer between langauges
- Similar to ORMs, again: impedance mismatches, memory management
- Gotchas:
  - **Marshalling**: conversion of data/object for calls ($\neq$ serialization)
  - **Memory Management**: Java/Python have Garbage Collectors (GCs)[1], C/C++ not - communicate lifetime events!

[1] Java Docs

[1] The name is interchangeably used with Foreign Function Interfaces in Rust and Python

# Bindings

## What?[1]

- (Semi-)Automatic translation layer between langauges
- Similar to ORMs, again: impedance mismatches, memory management
- Gotchas:
  - **Marshalling**: conversion of data/object for calls ($\neq$ serialization)
  - **Memory Management**: Java/Python have Garbage Collectors (GCs)[1], C/C++ not - communicate lifetime events!

[1] Java Docs

[1] The name is interchangeably used with Foreign Function Interfaces in Rust and Python

# Bindings

## Why?

# Bindings

## Why?

- Reuse existing libraries and code written in other languages
- Leverage performance or features of languages like C/C++
- Enable rapid prototyping using high-level languages
- Facilitate interoperability between different programming environments

# Bindings

## Why?

- Reuse existing libraries and code written in other languages
- Leverage performance or features of languages like C/C++
- Enable rapid prototyping using high-level languages
- Facilitate interoperability between different programming environments

# Bindings

## Why?

- Reuse existing libraries and code written in other languages
- Leverage performance or features of languages like C/C++
- Enable rapid prototyping using high-level languages
- Facilitate interoperability between different programming environments

# Bindings

## Why?

- Reuse existing libraries and code written in other languages
- Leverage performance or features of languages like C/C++
- Enable rapid prototyping using high-level languages
- Facilitate interoperability between different programming environments

# Bindings

## C++ ↔ Python: Boost.Python

**Boost.Python**[a] is a part of Boost for seamless interoperability between C++ and Python.

- Expose C++ classes and functions to Python.
- Automatic type conversion between C++ and Python types.

```cpp
#include <boost/python.hpp>
int add(int i, int j) { return i + j; }
BOOST_PYTHON_MODULE(example) {
    using namespace boost::python;
    def("add", add);
}
```

```python
import example
print(example.add(2, 3))  # Output: 5
```

[a] https://www.boost.org/doc/libs/release/libs/python/

# Bindings

## More about Boost

- Boost is one of the most respected and widely used C++ library collections.
- `Boost.Python` is just one example; other libraries include `Boost.Asio` (networking), `Boost.Filesystem`, `Boost.Spirit` (parsing), and `Boost.Test`.
- Many features from Boost have influenced or become part of the C++ Standard Library (e.g., `std::shared_ptr`, `std::regex`).
- Boost libraries are expert-reviewed and portable, supporting multiple platforms and compilers.

# Bindings

## Python → C++: pybind

**pybind11**[a] is a lightweight header-only library.

- Inspired by `Boost.Python`
- More lightweight, easier to operate (header-only)

[a] https://pybind11.readthedocs.io/

```
#include <pybind11/pybind11.h>
int add(int i, int j) { return i + j; }
PYBIND11_MODULE(example, m) {
    m.def("add", &add, "A function that
     ↪  adds two numbers");
}
```

```
import example
print(example.add(2, 3))  # Output: 5
```

# Bindings

## JavaScript → C++: Emscripten (WASM)

**Emscripten**[a] is a toolchain for compiling C/C++ code to WebAssembly (WASM).

- Also inspired by `Boost.Python`
- Created for WASM, can run C++ code (within a sandbox) in the browser.

```cpp
#include <emscripten/bind.h>
int add(int a, int b) { return a + b; }
EMSCRIPTEN_BINDINGS(my_module) {
    emscripten::function("add", &add);
}
```

```javascript
Module.onRuntimeInitialized = () => {
    console.log(Module.add(2, 3)); //
     ↪  Output: 5
};
```

[a] https://emscripten.org/

# Bindings

## Java ↔ C++: JNI (Java Native Interface)

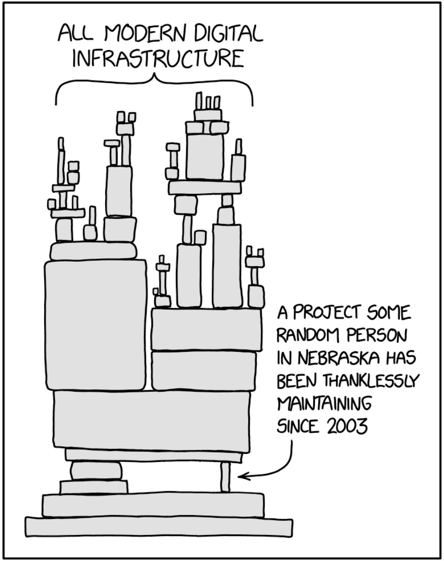
**JNI**[a] allows Java code to interoperate with native applications and libraries written in C or C++.

- Call C/C++ functions from Java and vice versa

```
#include <jni.h>
extern "C" JNIEXPORT jint JNICALL
↪   Java_Example_add(JNIEnv *, jobject,
↪   jint a, jint b)
{
    return a + b;
}
```

```
public class Example {
    static {
    ↪   System.loadLibrary("example");
    ↪   }
    public native int add(int a, int
    ↪   b);
}
```

[a] https://docs.oracle.com/en/java/javase/17/docs/specs/jni/index.html, Tutorial


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

[1] XKCD, https://xkcd.com/2347/

# Lombok[a]

- Reduces boilerplate code in Java
- Generates getters, setters, constructors, etc.
- Uses annotations for code generation

```java
import lombok.Data;
@Data
public class Person {
    private String name;
    private int age;
}
```

[a] https://projectlombok.org/

# Apache Commons[a]

- Collection of reusable Java components
- Utilities for strings, collections, IO, math, etc.
- Widely used in many Java projects

```
StringsComparator cmp = new
↪  StringsComparator("ABCFGH", "BCDEFG");
EditScript<Character> script =
↪  cmp.getScript();
int mod = script.getModifications();
```

[a] https://commons.apache.org/

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

. . . regression of code!

- Testing: pay-off time only after a certain number of runs!

- Granularity![2]

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

... regression of code!

- Testing: pay-off time only after a certain number of runs!
- Granularity![2]

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

... regression of code!

- Testing: pay-off time only after a certain number of runs!
- Granularity![2]

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

... regression of code!

- Testing: pay-off time only after a certain number of runs!
- Granularity![2]

System Testing

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

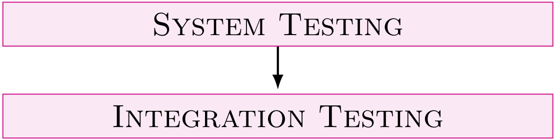
[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

... regression of code!

- Testing: pay-off time only after a certain number of runs!
- Granularity![2]



System Testing

Integration Testing

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

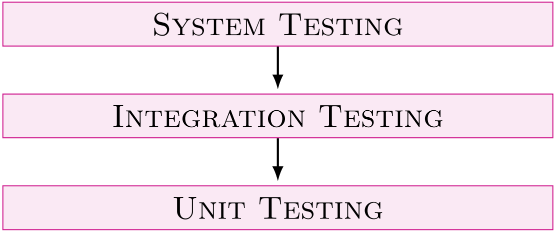
[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# (Java) Testing [1]

- Automated tests are essential to verify functionality &

. . . regression of code!

- Testing: pay-off time only after a certain number of runs!
- Granularity![2]



SYSTEM TESTING

INTEGRATION TESTING

UNIT TESTING

[1] J. Link and P. Fröhlich, **Unit Testing in Java**

[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**

# JUnit[a]

[a] https://junit.org/

- Standard framework for unit testing in Java
- Supports test annotations and assertions
- Integrates with build tools and IDEs

```
import org.junit.jupiter.api.Test;
import static
↪  org.junit.jupiter.api.Assertions.*;
class MyTest {
    @Test
    void testSum() {
        assertEquals(4, 2 + 2);
    }
}
```

# Mockito

- Popular mocking framework for unit tests
- Create mock objects and verify interactions
- Supports stubbing method calls

```
import static org.mockito.Mockito.*;

List<String> mockedList =
↪  mock(List.class);
when(mockedList.get(0))
    .thenReturn("first");
System.out.println(mockedList.get(0));
↪  // prints "first"
```

# Spring Boot[a]

- Framework for rapid development of Java applications
- Auto-configuration and embedded servers
- Minimal setup for production-ready apps

```
import org.springframework.*;
@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args)
     ↪  {
        SpringApplication.run(
                DemoApplication.class,
                 ↪  args);
    }
}
```

[a] https://spring.io/projects/spring-boot

# Spring Framework (REST API)[a]

- Uses annotations for request mapping.
- Supports JSON/XML serialization out of the box.

```java
import org.springframework.*;
@RestController
@RequestMapping("/api")
public class HelloController {
    @GetMapping("/hello")
    public String sayHello() {
        return "Hello, World!";
    }
}
```

[a] https://spring.io/projects/spring-framework

# Spring Data[a]

- Simplifies data access in Spring applications.
- Supports JPA, MongoDB, and more
- Reduces boilerplate for repositories.

```java
import org.springframework.*;
import javax.persistence.*;
@Entity
public class Person {
    @Id
    private Long id;
    private String name;
}
public interface PersonRepository
    extends JpaRepository<Person, Long>
    ↪  {
}
```

[a] https://spring.io/projects/spring-data

# Spring Security[a]

- Powerful authentication and authorization framework.
- Supports OAuth2, JWT, LDAP, etc.
- Easily integrates with Spring applications.

```java
import org.springframework.*;
public class SecurityConfig
    extends WebSecurityConfigurerAdapter {
    @Override
    protected void
        configure(HttpSecurity http){
        http.authorizeRequests()
            .anyRequest()
            .authenticated()
            .and()
            .formLogin();
    }
}
```

[a] https://spring.io/projects/spring-security

# Apache Jena[a]

- Semantic Web and Linked Data applications.
- Supports reading/writing RDF in various formats.

```
import org.apache.jena.rdf.model.*;
//...
Model model =
↪   ModelFactory.createDefaultModel();
Resource person = model

    ↪   .createResource("http://example.org/Person
    .addProperty(model

    ↪   .createProperty("http://xmlns.com/foaf/0.1
            "Alice");
model.write(System:out, "TURTLE");
```

[a] https://jena.apache.org/

# Apache Jena

## Querying with SPARQL

- Supports SPARQL for querying RDF data.
- Programmatic API for executing queries and processing results.
- Integrates with Jena models and datasets.

```
import org.apache.jena.query.*;
//...
String sparql =
    "SELECT ?name WHERE { ?p
    ↪   <http://xmlns.com/foaf/0.1/name>
    ↪   ?name }";
Query query =
    QueryFactory.create(sparql);
try {
    QueryExecution qexec =
    ↪   QueryExecutionFactory
            .create(query, model);
    ResultSet results = qexec.execSelect();
    //...
}
```

# Apache Solr[a]

[a] https://solr.apache.org/

- Open-source search platform built on Apache Lucene.
- Supports full-text search, faceting, and filtering.
- REST-like HTTP/XML and JSON APIs.

```
SolrClient solr = //...
SolrInputDocument doc = new
↪  SolrInputDocument();
doc.addField("id", "1");
doc.addField("name", "Alice");
solr.add(doc);
solr.commit();
SolrQuery query = new
↪  SolrQuery("name:Alice");
QueryResponse response =
↪  solr.query(query);
```

# Apache Spark[a]

## Overview

- Unified analytics engine for large-scale data processing.
- Supports batch and streaming workloads.
- APIs for Java, Scala, Python, and R.
- Distributed processing (RDDs, DataFrames).

```
JavaSparkContext sc = //...
List<Integer> data = Arrays.asList(1, 2, 3,
↪  4, 5);
JavaRDD<Integer> rdd =
↪  sc.parallelize(data);
JavaRDD<Integer> squared = rdd.map(x -> x *
↪  x);
squared.collect().forEach(n->log(n));
```

[a] https://spark.apache.org/

# Apache Spark

## MLLib

- Machine learning library for Apache Spark.
- Scalable algorithms for classification, regression, clustering, etc.
- Tools for feature extraction, transformation, and evaluation.

```java
LogisticRegression lr = new
↪  LogisticRegression();
lr.setMaxIter(10).setRegParam(0.01);
```

# Some Closing Notes

- ☕ Libraries:
  - Broad ecosystem,
  - Applicable to a *lot* of fields
- Use them in you job/uni/etc.
- (And also other languages!)

# Some Closing Notes

- ☕ Libraries:
  - Broad ecosystem,
  - Applicable to a *lot* of fields
- Use them in you job/uni/etc.
- (And also other languages!)

# Some Closing Notes

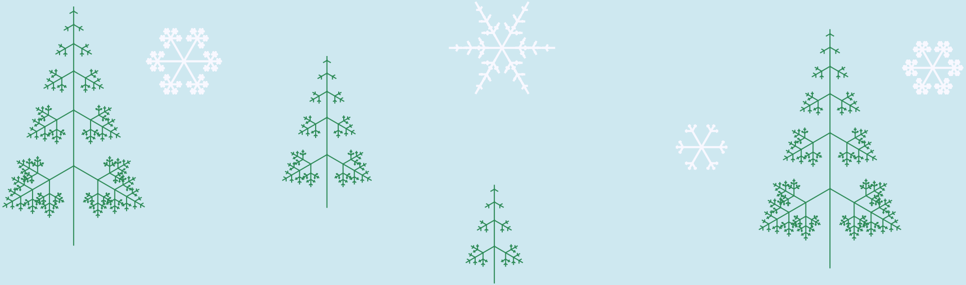
- ☕ Libraries:
  - Broad ecosystem,
  - Applicable to a *lot* of fields
- Use them in you job/uni/etc.
- (And also other languages!)

# Some Closing Notes

- ☕ Libraries:
  - Broad ecosystem,
  - Applicable to a *lot* of fields
- Use them in you job/uni/etc.
- (And also other languages!)

# Some Closing Notes

- ☕ Libraries:
  - Broad ecosystem,
  - Applicable to a *lot* of fields
- Use them in you job/uni/etc.
- (And also other languages!)



You are cordially invited to the

# Object-Oriented Programming 2: Christmas Lecture on the 18.12, 15:00

We will have: Lindenmayer Systems, Minecraft Programming, and Snacks!

# Feedback

# Bibliography I

## References

[1] J. Link and P. Fröhlich, **Unit Testing in Java**. Morgan Kaufmann, ISBN: 978-1-55860868-9. [Online]. Available: https://learning.oreilly.com/library/view/unit-testing-in/9781558608689/?sso_link=yes&sso_link_from=tugraz-at.

[2] P. C. Jorgensen, **Chapter 1: A Perspective on Testing**. Boca Raton, FL, USA: Auerbach Publications, ISBN: 978-1-46656068-0. [Online]. Available: https://learning.oreilly.com/library/view/software-testing-4th/9781466560680/ch01.html#ch00lev1sec13.