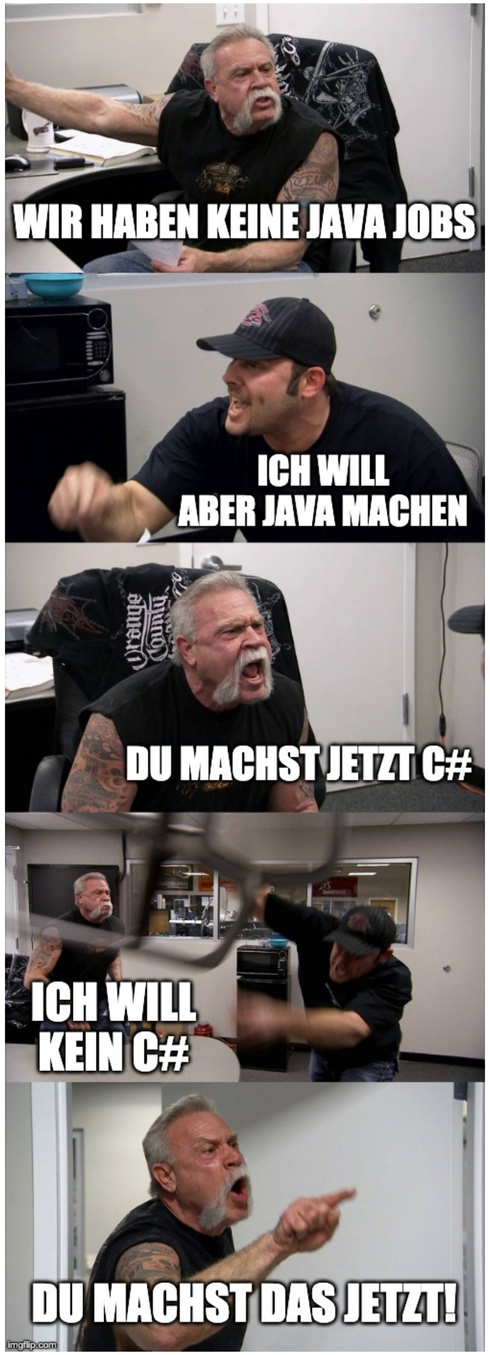
```
presentation {
    title = "Ein Kotlin Workshop"
    date = LocalDate.of(2025, 1, 27)

    speaker {
        name = "Lars Michaelis"
        company = NEUSTA SD GmbH

        social {
            email = "l.michaelis@neusta.de"
            github = "larmic"
        }
    }

    github = "github.com/larmic/workshop_kotlin"
}
```

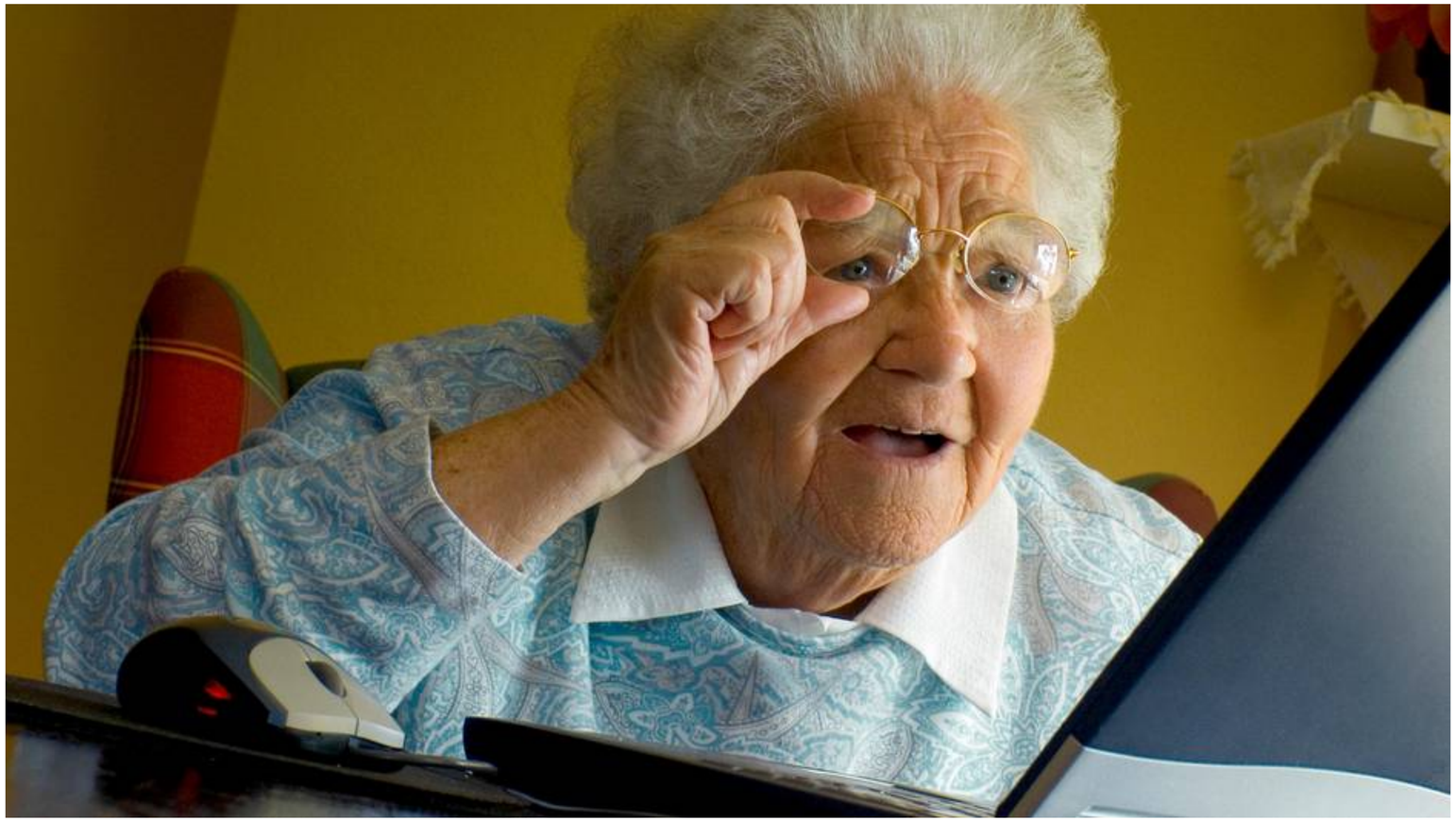
Agenda

1. Storytime
2. Einführung in die Basics
3. Programmieren & Vorstellen
4. We are going deeper
5. Programmieren & Vorstellen
6. Ausblick


WIR HABEN KEINE JAVA JOBS
ICH WILL
ABER JAVA MACHEN
DU MACHST JETZT C#
ICH WILL
KEIN C#
DU MACHST DAS JETZT!
imgflip.com

C#

```
public class SaleItem {
    public string Name { get; set; }
    public decimal Price { get; set; }
}
```



One language to rule them all

& readable by java developers

Hacking Time Basics

Properties

Functions

init {}

Multiline Strings

New

data classes

Semicolons

Strings

Ein Java-Beispiel

Address

+identifier: UUID = randomUUID
+street: Text
+houseNumber: Number
+houseNumberAffix: Text - optional
+zip: Number
+city: Text
+district: Text - optional

+readableAddress()

Definition inkl. Defaultwerte

Java Teil 1

```java
public class Address {

    private UUID identifier = UUID.randomUUID();
    private String street;
    private Integer houseNumber;
    private String houseNumberAffix;
    private Integer zip;
    private String city;
    private String district;

    public String getReadableAddress() {
        return street + " " + houseNumber
                      + " " + houseNumberAffix
                + "\n" + zip + " "
                + city + " (" + district + ")";
    }

    public void setIdentifier(UUID identifier) {
        if (identifier == null) {
            throw new IllegalArgumentException();
        }

        this.identifier = identifier;
    }

    public void setStreet(String street) {
        if (street == null || "".equals(street.trim())) {
            throw new IllegalArgumentException();
        }

        this.street = street;
    }

    public void setHouseNumber(Integer houseNumber) {
        if (houseNumber < 1) {
            throw new IllegalArgumentException();
        }
        this.houseNumber = houseNumber;
    }
```

Java Teil 2

```java
    public void setZip(Integer zip) {
        if (zip < 10000 || zip > 99999) {
            throw new IllegalArgumentException();
        }

        this.zip = zip;
    }

    public void setCity(String city) {
        if (city == null || "".equals(city.trim())) {
            throw new IllegalArgumentException();
        }

        this.city = city;
    }

    // getter + setter

}
```

Insgesamt
89 Code-Zeilen

Businesslogik

Java Teil 3

```java
final Address address = new Address();
address.setStreet("Konsul-Smidt-Str.");
address.setHouseNumber(24);
address.setZip(28217);
address.setCity("Bremen");
address.setDistrict("Häfen");

System.out.println(address.getReadableAddress());
```

Neues Pflichtfeld?

Feld vergessen?

Java Teil 1

```java
public class Address {

    private final UUID identifier;
    private String street;
    private Integer houseNumber;
    private String houseNumberAffix;
    private Integer zip;
    private String city;
    private String district;

    public Address(UUID identifier,
                   String street,
                   Integer houseNumber,
                   String houseNumberAffix,
                   Integer zip,
                   String city,
                   String district) {
        this.identifier
           = Objects.requireNonNullElseGet(identifier, UUID::randomUUID);
        this.setStreet(street);
        this.setHouseNumber(houseNumber);
        this.houseNumberAffix = houseNumberAffix;
        this.setZip(zip);
        this.setCity(city);
        this.district = district;
    }

    public String getReadableAddress() {
        return street + " " + houseNumber + " " + houseNumberAffix
                + "\n" + zip + " " + city + " (" + district + ")";
    }

    public void setStreet(String street) {
        if (street == null || "".equals(street.trim())) {
            throw new IllegalArgumentException();
        }

        this.street = street;
    }

    public void setHouseNumber(Integer houseNumber) {
        if (houseNumber < 1) {
            throw new IllegalArgumentException();
        }
        this.houseNumber = houseNumber;
    }

       // getter + setter (incl. validation)
}
```

Definition

Initialisierung (Defaultwerte)

Businesslogik

Insgesamt
99 Code-Zeilen

Java Teil 2

```java
final Address address = new Address(
        null,
        "Konsul-Smidt-Str.",
        24,
        null,
        28217,
        "Bremen",
        "Häfen");

System.out.println(address.getReadableAddress());
```

Reihenfolge?

Welches Feld darf Null sein?

Neues Pflichtfeld

Java Teil 1

```java
public class AddressBuilder {

    private UUID identifier = UUID.randomUUID();
    private final String street;
    private final Integer houseNumber;
    private String houseNumberAffix = null;
    private final Integer zip;
    private final String city;
    private String district = null;

    private AddressBuilder(String street,
                           Integer houseNumber,
                           Integer zip,
                           String city) {
        this.street = street;
        this.houseNumber = houseNumber;
        this.zip = zip;
        this.city = city;
    }

    public AddressBuilder identifier(UUID identi
        this.identifier = identifier;
        return this;
    }

    public AddressBuilder houseNumberAffix(String houseNumberAffix) {
        this.houseNumberAffix = houseNumberAffix;
        return this;
    }

    public AddressBuilder district(String district) {
        this.district = district;
        return this;
    }

    public Address build() {
        return new Address(identifier, street,
                houseNumber, houseNumberAffix, zip, city, district);
    }

    public static AddressBuilder builder(String street,
                                         Integer houseNumber,
                                         Integer zip,
                                         String city) {
        return new AddressBuilder(street, houseNumber, zip, city);
    }
}
```

Definition + Defaultwerte

Wo ist die Businesslogik?

Insgesamt 140 Code-Zeilen

Java Teil 2

```java
public static void main(String[] args) {
    final AddressBuilder builder = builder(
            "Konsul-Smidt-Str.",
            24,
            28217,
            "Bremen");
    builder.district("Häfen");
    final Address address = builder.build();

    System.out.println(address.getReadableAddress());
}
```

Klare Trennung der optionalen Attribute

Reihenfolge?

User Story

[Klasse: Address]

Anforderung

Schreibe eine Adress-Klasse in Kotlin

Akzeptanzkriterien

Die Klasse Address enthält die Attribute
* street (String); Pflichtfeld; darf nicht leer sein
* houseNumber (Int); Pflichtfeld; muss größer als 0 sein
* houseNumberAffix (String), optional
* zip (Int); Pflichtfeld; muss zwischen 10000 und 99999 liegen
* city (String); Pflichtfeld, darf nicht leer sein

Es gibt eine Methode, die die Adresse schon in der Konsole darstellt

Zeit

30 min

Kein TDD, keine Tests

Address
~~+identifier: UUID = randomUUID~~ +street: Text +houseNumber: Number +houseNumberAffix: Text - optional +zip: Number +city: Text ~~+district: Text - optional~~
+readableAddress()

KOTLIN
DEVELOPER
JAVA
imgflip.com

Hacking Time Deeper

Listen

mockk

when {}

extension functions

Invoke functions

sealed classes

kotest

User Story

[Klasse: Address]

Anforderung

Erzeuge einen Usecase, der eine Adresse in eine Datenbank (Cache) speichert und vorher prüft, ob eine entsprechende Adresse schon existiert.

Akzeptanzkriterien

* Die Attribute der Adresse sind getrimmt
* Ein Repository hält alle Adressen (AddressEntity) im Speicher
* Ein fachlicher Service prüft
  * prüft, ob die Adresse schon existiert
  * enthält alle notwendigen Felder als Signatur
  * erstellt eine Address-Instanz und sammelt alle Fehler ein
  * Rückgabe ist entweder ein Fehler-Object oder die erstellte Adresse
* Es gibt ausreichend Unit-Tests für den Service

Zeit

90 min

WHY KOTLIN?
imgflip.com

Easy to learn

Properties

No more Nullpointer Exceptions

No more semicolons

Write less code - be happy

Prefers Compile Errors

Especially for Java developers

100% Java interoperable

Use Java Frameworks and Libraries

Supports >= Java 6

Ausblick

Kotlin DSL

```
@Test
fun `names are valid and skill is created`() {
    apiTestSkill<CreateSkillDto>(mockMvc) {
        post {
            prepare {
                every {
                    skillAnlageUseCaseMock.legeAn(
                        gewichtung = body!!.gewichtung,
                    )
                } returns this@post.body!!.mapToSkillCreated()
            }
            body {
                gewichtung = "RELEVANT"
            }

            expectedResult {
                status { isCreated() }
                content { contentType(MediaType.APPLICATION_JSON) }
            }

            expectedJsonContent {
                "gewichtung" hasValue body!!.gewichtung
            }
        }
    }
}
```

THANKS!
QUESTIONS?
imgflip.com