DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Tutorium Programmierung 1

21.01.2026

Jannes Kurzke und Fabian Bauriedl

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Inhalt

1. Git Basics
2. Datentypen
3. Operatoren
4. Methoden
5. Array und ArrayList
6. Kontrollstrukturen + Ternary Operator
7. Schleifen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Organisatorisches

> Steffen und Mario behandeln die Themen in unterschiedlicher Reihenfolge

Organisatorisches

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Komm in die Gruppe!

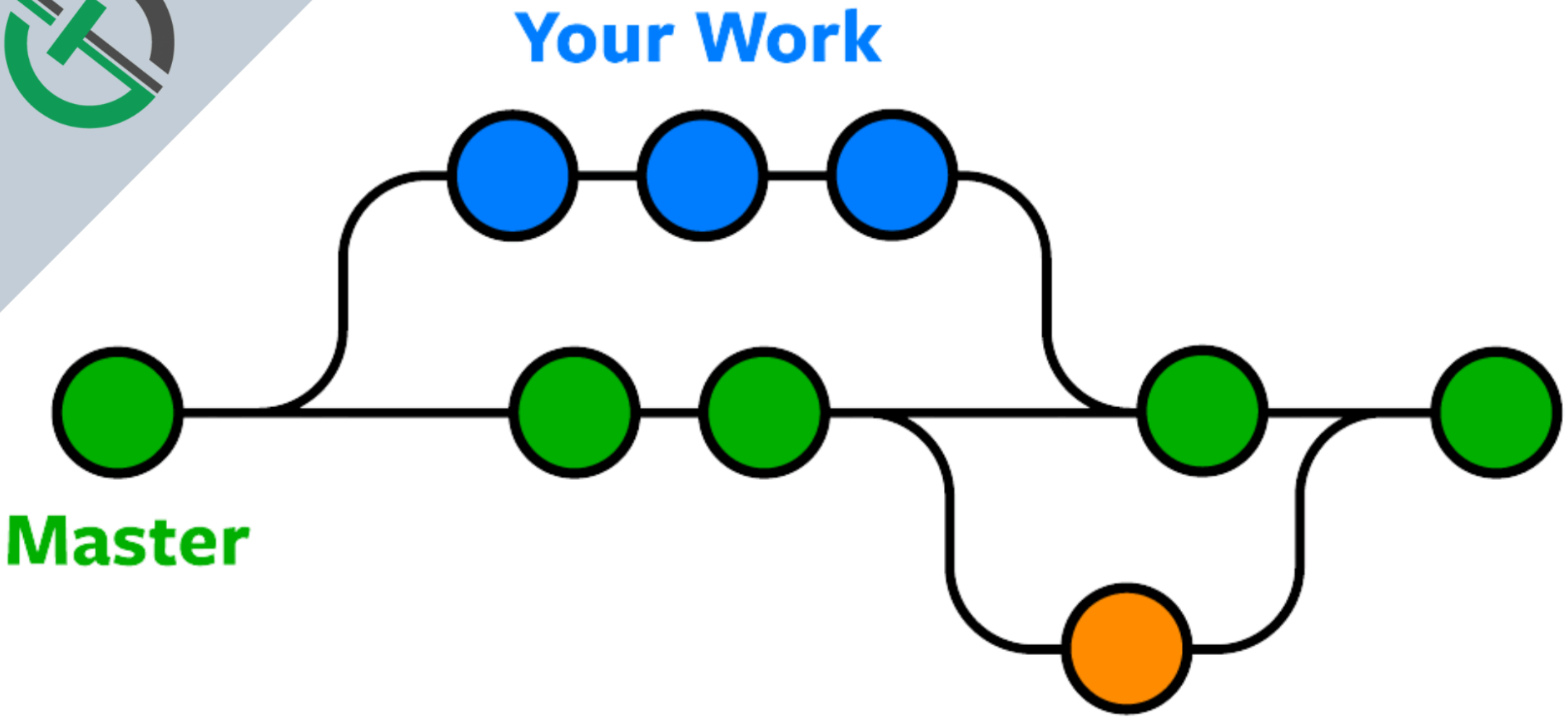
Git Basics

DHBW
Duale Hochschule

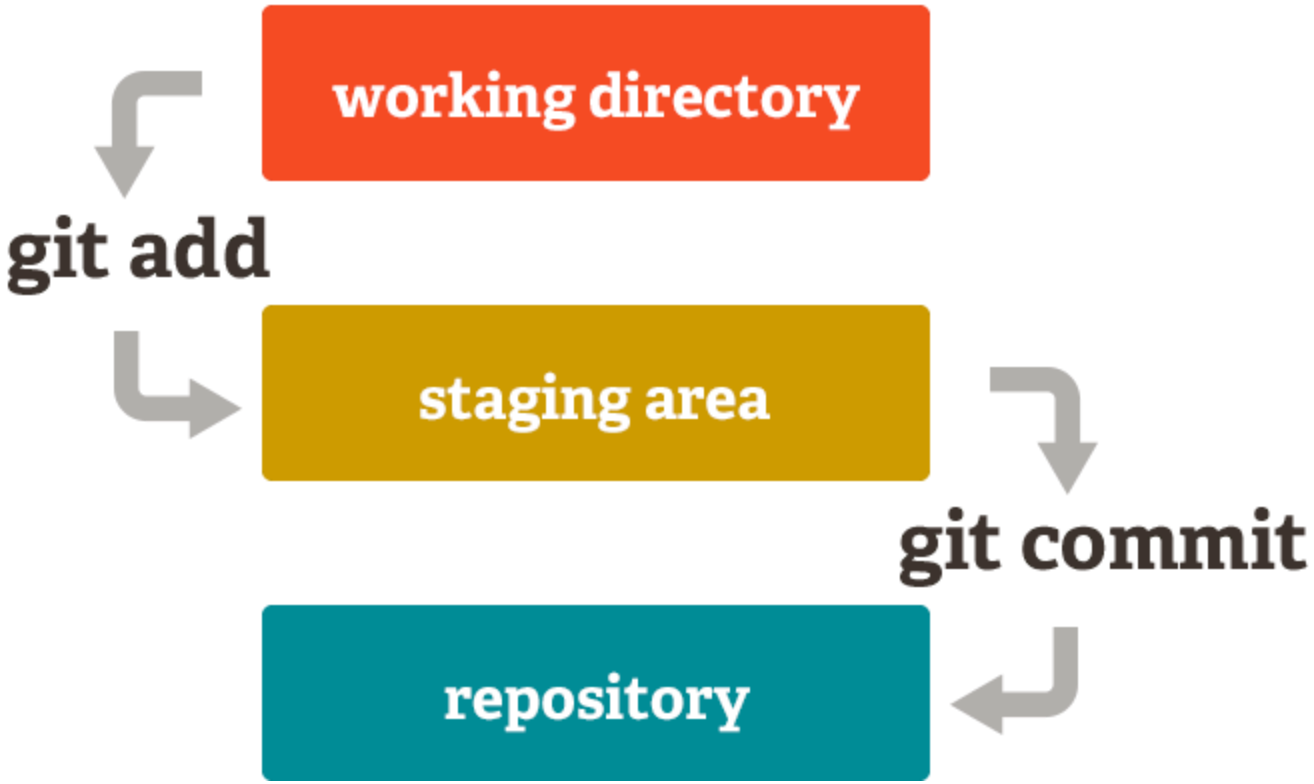
Git Basics - Branches

DHBW

www.softwaretestingo.com



Git Basics - Staging



DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Git Basics - Commands

Befehl	Beispiel
git clone	git clone https://github.com/Kona418/ProgrammierungTutorium.git
git add <>	git add ReadMe.md
git commit -m""	git commit -m"Updated Repo name in ReadMe"
git push	
git pull	
git branch	
git checkout	git checkout exercise/strings

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datentypen

- Bestimmung wie Daten gespeichert und gelesen werden
- Planbarkeit von Speicherbedarf
- Integrierte Limitierung des Wertebereichs

> Unterscheidung zwischen primitiven und komplexen Datentypen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datentypen - Wahrheitswerte

Datentyp	Größe	Wertebereich
boolean	1 Bit	true, false

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datentypen - Ganzzahlen

Datentyp	Größe	Wertebereich
byte	1 Byte	-128 bis +127
short	2 Byte	-32.768 bis +32.767
int	4 Byte	-2.1 Mrd bis +2.1 Mrd
long	8 Byte	-9.2 Trill bis + 9.2 Trill

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datentypen - Gleitkommazahlen

Datentyp	Größe	Wertebereich
float	4 Byte	-3,4*10^38 bis 3,4*10^38
double	8 Byte	-1,7*10^308 bis 1,7*10^308

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datentypen - Zeichen

Datentyp	Größe	Wertebereich
char	2 Byte	\u0000 bis \uFFFF
String	variable Größe	jedes einzelne Zeichen wie bei char

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datenobjekte/ Variablen

- Variablen werden als Zwischenspeicher für Werte verwendet
- Variablen haben einen festen Datentyp
- Deklaration legt eine Variable an und reserviert Speicherplatz
- Initialisierung befüllt den reservierten Speicherplatz mit einem Wert

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Datenobjekte/ Variablen - Anlegen

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

    // Deklaration der Variable "name"
    String name;

    // Initialisierung der Variable "name" mit dem Wert "Thorsten"
    name = "Thorsten";

    // Deklaration und Initialisierung der Variable "alter" mit dem Wert "19"
    int alter = 19;
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Operatoren

- Berechnungen bzw. Umwandlung der in den Variablen gespeicherten Werte
- Verschiedene Operationen je nach Datentyp erlaubt
- Verschiedene Ergebnisse bei gleicher Operation je nach Datentyp

Addition, Subtraktion, Multiplikation


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

    int numA = 10;
    int numB = 5;
    int result;

    result = numA + numB;
    System.out.println(result);
    // 15

    result = numA - numB;
    System.out.println(result);
    // 5

    result = numA * numB;
    System.out.println(result);
    // 50
}
```

Division

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

    int numAInt = 10;
    int numBInt = 4;
    int resultInt;

    resultInt = numAInt / numBInt;
    System.out.println(resultInt);
    // 2

    double numADouble = 10;
    double numBDouble = 4;
    double resultDouble;

    resultDouble = numADouble / numBDouble;
    System.out.println(resultDouble);
    //2.5
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

String Addition

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

    String text1 = "Hello ";
    String text2 = "World!";

    System.out.println(text1 + text2);
    // Hello World!
}
```

Vergleichsoperatoren


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

    int numA = 10;
    int numB = 5;
    boolean result;

    result = numA > numB;
    System.out.println(result);
    // true

    result = numA < numB;
    System.out.println(result);
    // false

    result = numA == numB;
    System.out.println(result);
    // false

    result = numA != numB;
    System.out.println(result);
    // true
}
```

Boolische Operatoren


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

    boolean boolA = true;
    boolean boolB = false;
    boolean result;

    result = boolA && boolA;
    System.out.println(result);
    // true

    result = boolB || boolB;
    System.out.println(result);
    // false

    result = boolA || boolB;
    System.out.println(result);
    // true

    result = !boolA;
    System.out.println(result);
    // false
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Konvertierung von Variablentypen

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

    int numA = 10;
    System.out.println(numA);
    // 10

    double numB;
    numB = (double) numA;
    System.out.println(numB);
    // 10.0
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Methoden

- Main-Methode ist der Einstiegspunkt in das Programm
- Code-Abschnitte die mehrfach verwendbar sind
- Java kommt mit vielen nützlichen Standard Methoden
- Methoden lassen sich selbst definieren

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Hilfsmethoden komplexer Datentypen

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

    String name = "Thorsten";
    char firstLetter;

    firstLetter = name.charAt(0);
    System.out.println(firstLetter);
    // "T"
}
```

Hilfsmethoden komplexer Datentypen

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

    String userInput = "19";
    System.out.println(userInput * 2);
    // **error**

    int age;
    age = Integer.parseInt(userInput);
    System.out.println(age * 2);
    // 38
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Methoden - Aufbau

Bestandteil	Inhalt
Rückgabetyp	void, int, String, double, ...
Bezeichner	Name der Methode
Parameter	Eingabevariablen
Methodenrumpf	Auszuführender Code

Methoden Beispiel Addition

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

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

    int result;
    result = intAdder(10, 5);
    System.out.println(result);
    // 15
}

public static int intAdder(int numA, int numB){

    int internalResult = numA + numB;
    return internalResult;
}
```

Methoden Beispiel Alter Check


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

    boolean result;
    result = ageCheck(19);
    System.out.println(result);
    // true

    result = ageCheck(15);
    System.out.println(result);
    // false
}

public static boolean ageCheck(int age){

    boolean isAdult = age >= 18;
    return isAdult;
}
```

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Kontrollstrukturen

- Ausführung bestimmter Code-Abschnitte bei Erfüllung einer Bedingung
- Angbae eines alternativen Code-Abschnitts wenn Bedingung nicht erfüllt ist
- Vielseitiges Werkezug zum Erreichen eines komplexen Programm-Ablaufs

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Aufbau von If-Anweisungen

- if Schlüsselwort
- Bedingung
- Code Block

```
public static void main(String[] args) {
    int age = 19;
    if (age >= 18) {
        System.out.println("Come in");
    }
}
```

Aufbau von If-Else-Anweisungen

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

- if Schlüsselwort
- Bedingung
- Erfolg Code Block
- Misserfolg Code Block

```
public static void main(String[] args) {
    int age = 19;
    if (age >= 18) {
        System.out.println("Come in");
    } else {
        System.out.println("Adults only!");
    }
}
```

If-Else-Ketten

DHBW
Duale Hochschule
Baden-Württemberg

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

    int age = 18;
    if (age > 18) {
        System.out.println("Come in");

    } else if (age == 18){
        System.out.println("You're just old enough");

    }
    else {
        System.out.println("Adults only!");
    }
}
```

Switch-Case

- Ähnlich zu If-Else Anweisungen
- Entscheidung basiert hier auf dem Wert einer Variable
- Kann je nach Anwendung und Programmiersprache schneller sein als If-Else

Switch Aufbau

- Switch Schlüsselwort
- Variable die geprüft wird
- Case Schlüsselwort mit Wert
- Code Block
- Break Schlüsselwort
- Code Block

Switch-Case Beispiel


```java
public static void main(String[] args) {
    char gender = 'j';
    switch (gender) {
        case 'm':
            System.out.println("männlich");
            break;
        case 'w':
            System.out.println("weiblich");
            break;
        case 'd':
            System.out.println("divers");
            break;
        default:
            System.out.println("nur \"m|w|d\" zugelassen");
            break;
    }
}
```

Ternary Operator

- Kurzform der If-Else Anweisung


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

    int numA = 10;
    int numB = 5;
    String result;

    if (numA > numB) {
        result = "greater";
    } else {
        result = "less";
    }

    result = numA > numB ? "greater" : "less";
    System.out.println(result);
    // greater
}
```

Übungen (Stand 21.01.2026)

Vorbereitung:

- In VSCode/IntelliJ Terminal öffnen
- Repository klonen:
- 'git clone https://github.com/Kona418/ProgrammierungTutorium.git'
- 'cd ProgrammierungTutorium'
- Alternativ: Menüpunkt 'Neues Projekt aus Versionskontrolle' aufrufen und Repository Link einfügen

Verfügbare Themen:

- 'datatypes'
- 'methoden'
- 'kontrollstrukturen'

Übungen öffnen:

- nacheinander im Terminal ausführen (Thema aus Liste für THEMA einsetzen):
- 'git stash'
- 'git checkout semester1/THEMA'
- Im Ordner THEMAExercises sind die Aufgaben, in THEMASolutions die Lösungen
- Die Kommentare im Code beschreiben, was zu tun ist!

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg

Zeit zum Üben