

2. Tutorium

Datentypen, Operatoren, Referenzen, Strings

Tutorium 14

Péter Bohner | 09.11.2022

Tutorium 14

2022-11-07



Inhaltsverzeichnis

- 1. Wiederholung
- 2. Artemis
- 3. Klassen und Objekte
 - 3.1 Einführung
 - 3.2 Objekte
 - 3.3 Modellierung
- 4. Operatoren
- 5. Datentypen II
- 6. Referenzen
 - 6.1 Speicherung
- 7. Scanner

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o o oooo	oooo	oooo	oooo	oo	oo



2022-11-07

Inhaltsverzeichnis

- 1. Wiederholung
- 2. Artemis
- 3. Klassen und Objekte
 - 3.1 Einführung
 - 3.2 Objekte
 - 3.3 Modellierung
- 4. Operatoren
- 5. Datentypen II
- 6. Referenzen
 - 6.1 Speicherung
- 7. Scanner

Welche Befehlsformel innerhalb einer Klasse ermöglicht die Ausführung einer Java-Applikation?



2022-11-07

Tutorium 14

Wiederholung

Welche Befehlsformel innerhalb einer Klasse ermöglicht die Ausführung einer Java-Applikation?

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
●○○○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	○○○	○○○	○○	○○

Welche Befehlsformel innerhalb einer Klasse ermöglicht die Ausführung einer Java-Applikation?



KIT
Karlsruher Institut für Technologie

Die *main*-Methode ist der Haupteinstiegspunkt in die Anwendung. Hier startet die Programmausführung.

```
class JavaApp {  
    public static void main(String[] args) {  
        System.out.println("Hello world.");  
    }  
}
```

Java Dateien werden **exakt** nach der Klasse benannt: Sonst Compilerfehler.

2022-11-07

Tutorium 14

└─Wiederholung

└─Welche Befehlsformel innerhalb einer Klasse ermöglicht die Ausführung einer Java-Applikation?

Die main-Methode ist der Haupteinstiegspunkt in die Anwendung. Hier startet die Programmausführung.

```
class JavaApp {  
    public static void main(String[] args) {  
        System.out.println("Hello world.");  
    }  
}
```

Java Dateien werden **exakt** nach der Klasse benannt: Sonst Compilerfehler.

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○●○○○	○○○○○○○○○○○○○	○ ○ ○○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14

└─Wiederholung

└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○●○○○	○○○○○○○○○○○○○	○ ○ ○○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14

└─Wiederholung

└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○●○○○	○○○○○○○○○○○○○	○ ○ ○ ○○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14

└─Wiederholung

└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp
Wie kann eine Ausgabe auf der Konsole erzeugt werden?

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

System.out.println(); oder System.out.print();

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○●○○○	○○○○○○○○○○○○○	○ ○ ○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14

└─Wiederholung

└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp
Wie kann eine Ausgabe auf der Konsole erzeugt werden?
System.out.println(); oder System.out.print();

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: `javac JavaApp.java`

Ausführen: `java JavaApp`

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

`System.out.println();` oder `System.out.print();`

Sind Klassen Datentypen in Java?

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○●○○○	○○○○○○○○○○○○○	○ ○ ○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14
└─Wiederholung
└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: `javac JavaApp.java`
Ausführen: `java JavaApp`
Wie kann eine Ausgabe auf der Konsole erzeugt werden?
`System.out.println();` oder `System.out.print();`
Sind Klassen Datentypen in Java?

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

System.out.println(); oder System.out.print();

Sind Klassen Datentypen in Java?Ja...

```
class Car {  
    Body body;  
    Engine engine;  
    ...  
}
```

Wiederholung
●○○○

Artemis
○○○○○○○○○○○○○○

Klassen und Objekte
○
○
○○○○

Operatoren
○○○○

Datentypen II
○○○○

Referenzen
○○○○

Scanner
○○

Ende
○○

2022-11-07

Tutorium 14

└─Wiederholung

└─Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp
Wie kann eine Ausgabe auf der Konsole erzeugt werden?
System.out.println(); oder System.out.print();
Sind Klassen Datentypen in Java?Ja...
class Car {
 Body body;
 Engine engine;
 ...
}

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

System.out.println(); oder System.out.print();

Sind Klassen Datentypen in Java?Ja...

```
class Car {  
    Body body;  
    Engine engine;  
    ...  
}
```

Mit welchem Schlüsselwort kann ein neues Objekt einer Klasse erzeugt werden?

Wiederholung ●○○○	Artemis ○○○○○○○○○○○○	Klassen und Objekte ○ ○ ○○○○	Operatoren ○○○○	Datentypen II ○○○○	Referenzen ○○○○	Scanner ○○	Ende ○○
----------------------	-------------------------	---------------------------------------	--------------------	-----------------------	--------------------	---------------	------------

2022-11-07

Tutorium 14

Wiederholung

Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp
Wie kann eine Ausgabe auf der Konsole erzeugt werden?
System.out.println(); oder System.out.print();
Sind Klassen Datentypen in Java?Ja...
class Car {
 Body body;
 Engine engine;
 ...
}
Mit welchem Schlüsselwort kann ein neues Objekt einer Klasse erzeugt werden?

Wiederholung



Mit welchem Befehl kann ein Java-Programm kompiliert werden?

Kompilieren: javac JavaApp.java

Ausführen: java JavaApp

Wie kann eine Ausgabe auf der Konsole erzeugt werden?

System.out.println(); oder System.out.print();

Sind Klassen Datentypen in Java?Ja...

```
class Car {  
    Body body;  
    Engine engine;  
    ...  
}
```

Mit welchem Schlüsselwort kann ein neues Objekt einer Klasse erzeugt werden?

```
new Classname();
```

Wiederholung ●○○○	Artemis ○○○○○○○○○○○○	Klassen und Objekte ○ ○ ○○○○	Operatoren ○○○	Datentypen II ○○○	Referenzen ○○○	Scanner ○○	Ende ○○
----------------------	-------------------------	---------------------------------------	-------------------	----------------------	-------------------	---------------	------------

2022-11-07

Tutorium 14

Wiederholung

Wiederholung

Mit welchem Befehl kann ein Java-Programm kompiliert werden?
Kompilieren: javac JavaApp.java
Ausführen: java JavaApp
Wie kann eine Ausgabe auf der Konsole erzeugt werden?
System.out.println(); oder System.out.print();
Sind Klassen Datentypen in Java?Ja...
class Car {
 Body body;
 Engine engine;
 ...
}
Mit welchem Schlüsselwort kann ein neues Objekt einer Klasse erzeugt werden?
new Classname();

Wiederholung primitive Datentypen



Typ	Erklärung	Wertebereich	Beispielwerte
boolean	Wahrheitswerte	true oder false	true, false
char	16-Bit-Unicode	0x0000 ... 0xffff	'A', '\n', '\u05D0'
byte	8-Bit-Integer	$-2^7 \dots 2^7 - 1$	12
short	16-Bit-Integer	$-2^{15} \dots 2^{15} - 1$	12
int	32-Bit-Integer	$-2^{31} \dots 2^{31} - 1$	12
long	64-Bit-Integer	$-2^{63} \dots 2^{63} - 1$	12L, 14L
float	32-Bit-Gleitk.	1,40239846E-45f ... 3,40282347E+38f	9.81F, 0.3E-8F, 2f
double	64-Bit-Gleitk.	4,94065645841246544E-324 ... 1,79769131486231570E+308	9.81, 3e1

2022-11-07

Typ	Erklärung	Wertebereich	Beispielwerte
boolean	Wahrheitswerte	true oder false	true, false
char	16-Bit-Unicode	0x0000 ... 0xffff	'A', '\n', '\u05D0'
byte	8-Bit-Integer	$-2^7 \dots 2^7 - 1$	12
short	16-Bit-Integer	$-2^{15} \dots 2^{15} - 1$	12
int	32-Bit-Integer	$-2^{31} \dots 2^{31} - 1$	12
long	64-Bit-Integer	$-2^{63} \dots 2^{63} - 1$	12L, 14L
float	32-Bit-Gleitk.	1,40239846E-45f ... 3,40282347E+38f	9.81F, 0.3E-8F, 2f
double	64-Bit-Gleitk.	4,94065645841246544E-324 ... 1,79769131486231570E+308	9.81, 3e1

Wiederholung Variablen



„Platzhalter“ für Werte eines Datentyps

Deklaration

- Name und Datentyp der Variable
- Datentyp Name;

Zuweisung

- Wert der Variable
- Name = Wert;

Initialisierung

- Kombination aus Deklaration und Zuweisung
- Datentyp Name = Wert;

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○●○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	○○○	○○○	○○	○○

„Platzhalter“ für Werte eines Datentyps

Deklaration	Zuweisung
<ul style="list-style-type: none">Name und Datentyp der VariableDatentyp Name;	<ul style="list-style-type: none">Wert der VariableName = Wert;
Initialisierung	
<ul style="list-style-type: none">Kombination aus Deklaration und ZuweisungDatentyp Name = Wert;	

Attribute



Wie greift man auf Attribute von Objekten zu?

- Zugriff mit Objektname.Variablenname
- Umgang wie mit „normalen“ Variablen
- Also Initialisierung mit:
⇒ Objektname.Variablenname = Wert;

Wiederholung
○○○●

Artemis
○○○○○○○○○○○○○○

Klassen und Objekte
○
○
○
○○○○

Operatoren
○○○

Datentypen II
○○○

Referenzen
○○○

Scanner
○○

Ende
○○

2022-11-07

Tutorium 14
└─Wiederholung
└─Attribute

Wie greift man auf Attribute von Objekten zu?

- Zugriff mit Objektname.Variablenname
- Umgang wie mit „normalen“ Variablen
- Also Initialisierung mit:
⇒ Objektname.Variablenname = Wert;

Artemis



Artemis

- neues Abgabesystem
- nutzt im Hintergrund GitLab und Jenkins
- Online-Editor: Code kann direkt komfortable in Artemis bearbeitet werden
- Oder: Aufgabe mit git clonen, lokal bearbeiten und Änderungen pushen

Registrierung

- Registrierung drücken und mit KIT Account einloggen
- E-Mail mit Link zur Passwort-Setzung für Artemis (GitLab, Jenkins)
- Anmeldung mit Passwort und Kurs beitreten

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	●○○○○○○○○○○○○	○ ○ ○ ○○○○	○○○	○○○	○○○	○○	○○

2022-11-07

Tutorium 14
└─Artemis

└─Artemis

Artemis <ul style="list-style-type: none">■ neues Abgabesystem■ nutzt im Hintergrund GitLab und Jenkins■ Online-Editor: Code kann direkt komfortable in Artemis bearbeitet werden■ Oder: Aufgabe mit git clonen, lokal bearbeiten und Änderungen pushen
Registrierung <ul style="list-style-type: none">■ Registrierung drücken und mit KIT Account einloggen■ E-Mail mit Link zur Passwort-Setzung für Artemis (GitLab, Jenkins)■ Anmeldung mit Passwort und Kurs beitreten

Git Basics



Git Basics

- `git clone url` - Clont das GitLab Repository lokal auf euren Rechner
- `git add file` - „staged“ (markiert) Datei für den nächsten Commit
- `git add -A` - Alle neuen/veränderten Dateien werden „gestaged“
- `git commit -m „Nachricht“` „gestagete“ Änderungen werden in einem Commit zusammengefasst
- `git pull` - Änderungen aus GitLab Repository werden heruntergeladen
- `git push` - Änderungen aus lokalen Repository werden hochgeladen
- mehr zu Git in SWT I oder in der [Git Dokumentation](#)
- ...oder Git über eclipse oder eine der vielen anderen [GUIs](#) benutzen

Nach jedem commit führt Artemis einen build aus und führt automatisiert die public Tests durch

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○○	●○○○○○○○○○○○	○ ○ ○○○○	○○○○	○○○○	○○○○	○○	○○

2022-11-07

Tutorium 14

Artemis

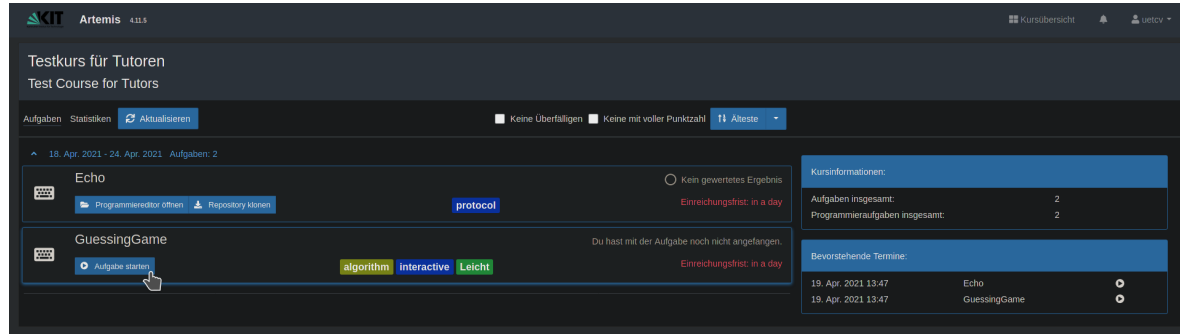
Git Basics

Git Basics

- `git clone url` - Clont das GitLab Repository lokal auf euren Rechner
- `git add file` - „staged“ (markiert) Datei für den nächsten Commit
- `git add -A` - Alle neuen/veränderten Dateien werden „gestaged“
- `git commit -m „Nachricht“` „gestagete“ Änderungen werden in einem Commit zusammengefasst
- `git pull` - Änderungen aus GitLab Repository werden heruntergeladen
- `git push` - Änderungen aus lokalen Repository werden hochgeladen
- mehr zu Git in SWT I oder in der [Git Dokumentation](#)
- ...oder Git über eclipse oder eine der vielen anderen [GUIs](#) benutzen

Nach jedem commit führt Artemis einen build aus und führt automatisiert die public Tests durch

Artemis - Aufgabe starten



1. „Aufgabe starten“ anklicken

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○○	○○●○○○○○○○○○○	○ ○ ○ ○○○○	○○○○	○○○○	○○○○	○○	○○

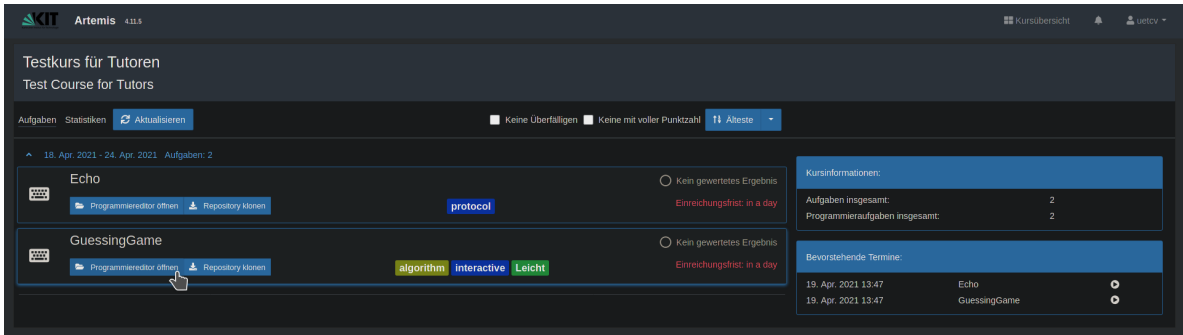
2022-11-07

Artemis - Aufgabe starten



1. „Aufgabe starten“ anklicken

Artemis - Webeditor nutzen



2. „Programmiereditor öffnen“ anklicken

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○○	○○●○○○○○○○○○	○ ○ ○○○○○	○○○○	○○○○	○○○○	○○	○○

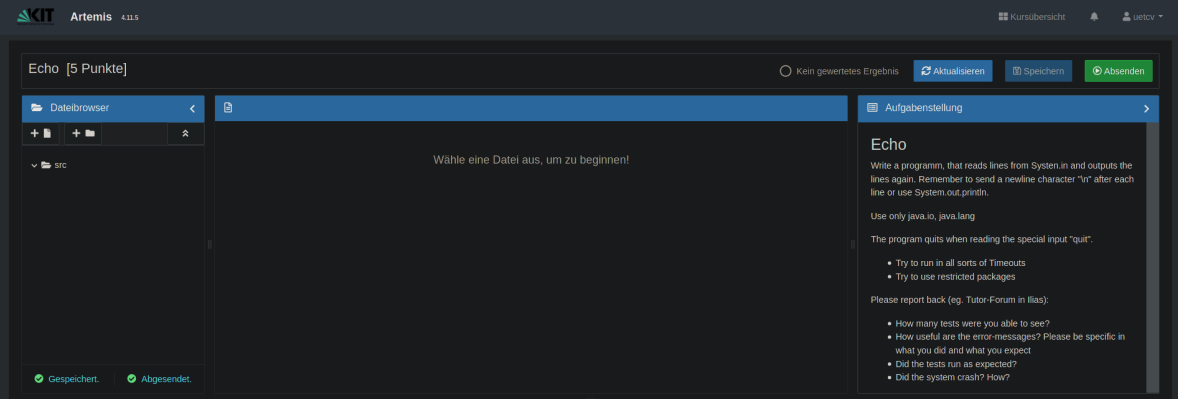
2022-11-07

Tutorium 14

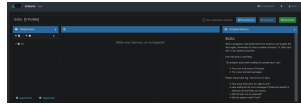
Artemis

Artemis - Webeditor nutzen

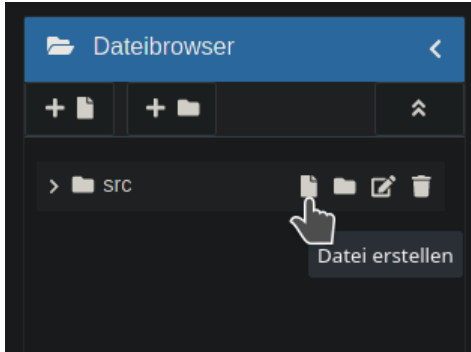




2022-11-07
Tutorium 14
Artemis



Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○○	○○○○●○○○○○○○○	○ ○ ○○○○○	○○○○	○○○○	○○○○	○○	○○



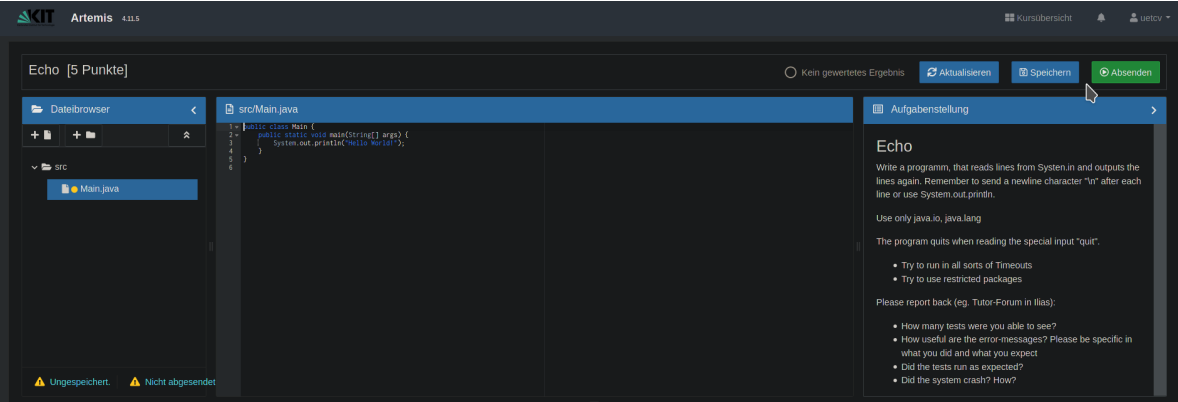
3. Über den Dateibrowser können neue Dateien angelegt werden
4. Code schreiben

2022-11-07

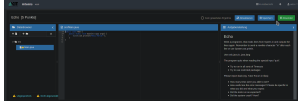
Tutorium 14 Artemis



3. Über den Dateibrowser können neue Dateien angelegt werden
4. Code schreiben



5. „Speichern“ um Änderung zu sichern
6. „Absenden“ um Projekt zu builden, testen und abzugeben



5. „Speichern“ um Änderung zu sichern
6. „Absenden“ um Projekt zu builden, testen und abzugeben

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○○	○○○○○○●○○○○○	○ ○ ○○○○	○○○○	○○○○	○○○○	○○	○○

Artemis - Lokal arbeiten



KIT

Artemis

4.11.5

Kursübersicht

uetcv

Testkurs für Tutoren

Test Course for Tutors

Aufgaben

Statistiken

Aktualisieren

Keine Überfalligen

Keine mit voller Punktzahl

Alteste

18. Apr. 2021 - 24. Apr. 2021

Aufgaben: 2

Echo

Programmiereditor öffnen

Repository klonen

protocol

Kein gewertetes Ergebnis

Einreichungsfrist: in a day

GuessingGame

Programmiereditor öffnen

Repository klonen

algorithm

interactive

Leicht

Kein gewertetes Ergebnis

Einreichungsfrist: in a day

Kursinformationen:

Aufgaben insgesamt: 2

Programmieraufgaben insgesamt: 2

Bevorstehende Termine:

19. Apr. 2021 13:47

Echo

19. Apr. 2021 13:47

GuessingGame

2. „Repository klonen“ anklicken

Wiederholung

Artemis

Klassen und Objekte

Operatoren

Datentypen II

Referenzen

Scanner

Ende

○○○○○

○○○○○○●○○○○○

○
○
○○○○

○○○○

○○○○

○○○○

○○

○○

2022-11-07

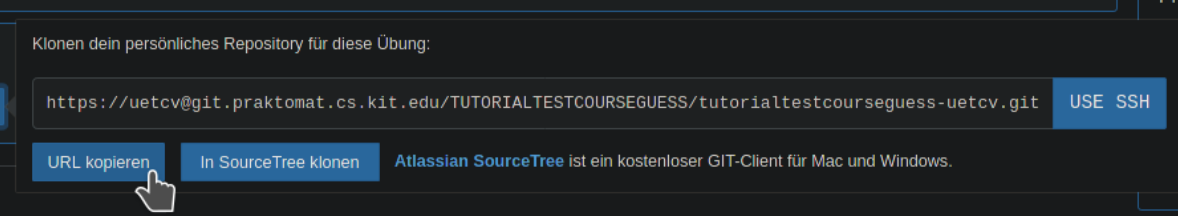
Tutorium 14

Artemis

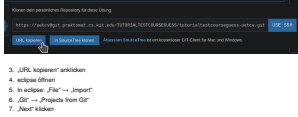
Artemis - Lokal arbeiten



2. „Repository klonen“ anklicken



- 3. „URL kopieren“ anklicken
- 4. eclipse öffnen
- 5. In eclipse: „File“ → „Import“
- 6. „Git“ → „Projects from Git“
- 7. „Next“ klicken



2022-11-07

Tutorium 14

Artemis

Source Git Repository

Enter the location of the source repository.

Location

URI:

https://uetcv@git.praktomat.cs.kit.edu/TUTORIALTESTCOURSEGUSS/tutorialtestcourseguess-uetcv.git

Local Folder...

Local Bundle File...

Host:

git.praktomat.cs.kit.edu

Repository path:

/TUTORIALTESTCOURSEGUSS/tutorialtestcourseguess-uetcv.git

Connection

Protocol:

https

Port:

Authentication

User:

uetcv

Password:

☐ Store in Secure Store

?

< Back

Next >

Cancel

Finish

8. Unter „URI“ kopierte URL einfügen

9. Unter „Authentication“ Artemis Passwort eintragen

10. 3 x „Next“ klicken

11. „Finish“ klicken

Wiederholung

Artemis

Klassen und Objekte

Operatoren

Datentypen II

Referenzen

Scanner

Ende

○○○○○

○○○○○○○○●○○○

○
○
○○○○○

○○○○

○○○○

○○○○

○○

○○

8. Unter „URI“ kopierte URL einfügen

9. Unter „Authentication“ Artemis Passwort eintragen

10. 3 x „Next“ klicken

11. „Finish“ klicken

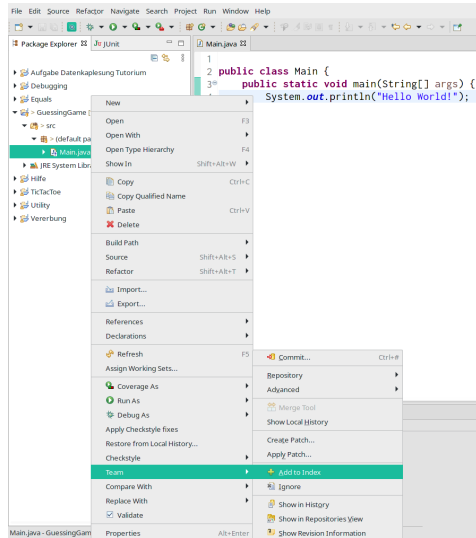
17/43

09.11.2022

Péter Bohner: Tutorium 14

Programmieren Tutorium

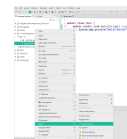
12. Repository ist nun lokal verfügbar
13. Nun kann normal lokal in eclipse programmiert werden
14. geänderte/erstellte Dateien müssen nun „gestaged“ und „committed“ werden
15. „Rechtsklick“ auf geänderte Datei → „Team“ → „Add to Index“
16. wurden alle geänderten Dateien „gestaged“:
„Rechtsklick“ auf Projekt → „Team“ → „Commit“



2022-11-07

Tutorium 14 Artemis

12. Repository ist nun lokal verfügbar
13. Nun kann normal lokal in eclipse programmiert werden
14. geänderte/erstellte Dateien müssen nun „gestaged“ und „committed“ werden
15. „Rechtsklick“ auf geänderte Datei → „Team“ → „Add to Index“
16. wurden alle geänderten Dateien „gestaged“:
„Rechtsklick“ auf Projekt → „Team“ → „Commit“



Wiederholung
○○○○○

Artemis
○○○○○○○○○●○○

Klassen und Objekte
○
○
○○○○○

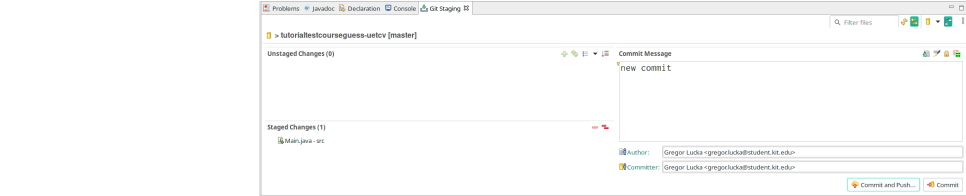
Operatoren
○○○○

Datentypen II
○○○○

Referenzen
○○○○

Scanner
○○

Ende
○○



- 17. beliebige „Commit Message“ wählen
- 18. „Commit and Push...“ klicken
- 19. „Preview“ klicken
- 20. „Push“ klicken
- 21. „Close“ klicken
- 22. Änderungen sind nun in Artemis eingebucht und die Tests werden ausgeführt
- 23. Ergebnisse der Tests sind in Artemis verfügbar





Fragen?

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo●	o o oooo	ooo	ooo	ooo	oo	oo

2022-11-07	Tutorium 14
	└─ Artemis
	└─ Sonstiges

Klassen und Objekte



Allgemein

- Modellierung der Realität
- Objekte mit Eigenschaften und Verhalten

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	● ○ oooo	oooo	oooo	oooo	oo	oo

Klassen und Objekte



Allgemein

- Modellierung der Realität
- Objekte mit Eigenschaften und Verhalten

Was ist eine Klasse?

- Objekte, die ähnliche Eigenschaften und dasselbe Verhalten haben, werden zusammengefasst
- Klasse stellt Bauplan für Objekte dar
- Attribute und Methoden werden festgelegt

Wiederholung
○○○○

Artemis
○○○○○○○○○○○○○○

Klassen und Objekte
●
○
○○○○

Operatoren
○○○○

Datentypen II
○○○○

Referenzen
○○○○

Scanner
○○

Ende
○○

2022-11-07

Tutorium 14
└─ Klassen und Objekte
 └─ Einführung
 └─ Klassen und Objekte

Allgemein
<ul style="list-style-type: none">■ Modellierung der Realität■ Objekte mit Eigenschaften und Verhalten
Was ist eine Klasse?
<ul style="list-style-type: none">■ Objekte, die ähnliche Eigenschaften und dasselbe Verhalten haben, werden zusammengefasst■ Klasse stellt Bauplan für Objekte dar■ Attribute und Methoden werden festgelegt

Was ist ein Objekt?



Instanzen, die aus dem Bauplan entstehen können

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o ● oooo	oooo	oooo	oooo	oo	oo

2022-11-07

- Tutorium 14
 - Klassen und Objekte
 - Objekte
 - Was ist ein Objekt?

Instanzen, die aus dem Bauplan entstehen können

Was ist ein Objekt?



Instanzen, die aus dem Bauplan entstehen können

Charakterisierung eines Objekts

- **Identität:** Bleibt immer gleich, egal ob sich der Zustand ändert
- **Zustand:** Menge aller Attributwerte
- **Verhalten:** Methoden

2022-11-07

Tutorium 14
└─ Klassen und Objekte
 └─ Objekte
 └─ Was ist ein Objekt?

Instanzen, die aus dem Bauplan entstehen können

Charakterisierung eines Objekts

- **Identität:** Bleibt immer gleich, egal ob sich der Zustand ändert
- **Zustand:** Menge aller Attributwerte
- **Verhalten:** Methoden

Klasse



Wie bauen die Klasse **Laptop**:

Eigenschaften

2022-11-07

Wie bauen die Klasse **Laptop**:

Eigenschaften

Klasse

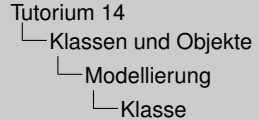


Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

2022-11-07



Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

Klasse



Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

Verhalten

Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

Verhalten

Klasse



Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

Verhalten

- Hochfahren
- Aufklappen
- Programm ausführen

Wie bauen die Klasse **Laptop**:

Eigenschaften

- Arbeitsspeicher
- Tastatur
- Display
- Ist es ein Convertible?

Verhalten

- Hochfahren
- Aufklappen
- Programm ausführen

Attribute

```
class Laptop {  
    int ram;  
    Keyboard keyboard;  
    Display display;  
    boolean isConvertible;  
  
    void powerOn() { ... }  
    void open(){ ... }  
    void runProgram(String name) { ... }  
}
```

Wiederholung
○○○○Artemis
○○○○○○○○○○○○○○Klassen und Objekte
○
○
○●○○Operatoren
○○○○Datentypen II
○○○○Referenzen
○○○○Scanner
○○Ende
○○

2022-11-07

Tutorium 14
└─ Klassen und Objekte
 └─ Modellierung
 └─ Code

```
Attribute  
class Laptop {  
    int ram;  
    Keyboard keyboard;  
    Display display;  
    boolean isConvertible;  
  
    void powerOn() { ... }  
    void open(){ ... }  
    void runProgram(String name) { ... }  
}
```

Wie erzeugt man ein Objekt?



Der Ausdruck `new Classname();` erzeugt ein neues Objekt.
Mit Laptop `laptop = new Laptop();` wird das Objekt `laptop` vom Typ `Laptop` erstellt.
`laptop.ram = 16;` ändert den Wert des Arbeitsspeichers.
Mehr zu Konstruktoren kommt bald.

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○●○○	○○○	○○○	○○○	○○	○○

2022-11-07

- Tutorium 14
 - Klassen und Objekte
 - Modellierung
 - Wie erzeugt man ein Objekt?

Der Ausdruck `new Classname();` erzeugt ein neues Objekt.
Mit Laptop `laptop = new Laptop();` wird das Objekt `laptop` vom Typ `Laptop` erstellt.
`laptop.ram = 16;` ändert den Wert des Arbeitsspeichers.
Mehr zu Konstruktoren kommt bald.

Aufgaben



Teil A

Modelliert ein Klasse Motor. Ein Motor besitzt eine gewisse Anzahl an PS und Zylindern. Er hat außerdem ein Gewicht und Hubraum (in l). Wählt geeignete Datentypen.

Teil B

Modelliert eine Klasse Karosserie. Sie hat ein Gewicht, Anzahl an Sitzplätzen und eine bestimmte Form (Name der Form).

Teil C

Überlegt euch, wie man aus den beiden gegebenen Klassen in Java-Code eine Klasse Auto entwerfen könnte. Zur Info: Ein Auto hat eine Karosserie und einen Motor.

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○○●○	○○○	○○○	○○○	○○	○○

2022-11-07

Tutorium 14
└─ Klassen und Objekte
└─ Modellierung
└─ Aufgaben

Teil A
Modelliert ein Klasse Motor. Ein Motor besitzt eine gewisse Anzahl an PS und Zylindern. Er hat außerdem ein Gewicht und Hubraum (in l). Wählt geeignete Datentypen.
Teil B
Modelliert eine Klasse Karosserie. Sie hat ein Gewicht, Anzahl an Sitzplätzen und eine bestimmte Form (Name der Form).
Teil C
Überlegt euch, wie man aus den beiden gegebenen Klassen in Java-Code eine Klasse Auto entwerfen könnte. Zur Info: Ein Auto hat eine Karosserie und einen Motor.

Lösungsvorschlag



Teil A und B

```
class Engine {  
    int horsepower;  
    int numberOfCylinders;  
    int weight;  
    double engineDisplacement;  
}
```

```
class Body {  
    int weight;  
    int numberOfSeats;  
    String type;  
}
```

Teil C

```
class Car {  
    Body body;  
    Engine engine;  
    ...  
}
```

- Klassen können als eigener Datentyp genutzt werden
- Attribute können also als Typ eine bestimmte Klasse haben
- Sie nutzen deren Eigenschaften und Verhalten

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o o oooo●	oooo	oooo	oooo	oo	oo

2022-11-07

Tutorium 14

- └ Klassen und Objekte
 - └ Modellierung
 - └ Lösungsvorschlag



Arithmetische Operationen



		Präzedenz
unäres -/+		1
+	Addition	3
-	Subtraktion	3
*	Multiplikation	2
/	Division	2
%	Modulo (Rest bei Ganzzahldivision)	2

Präzedenz

Der Wert gibt an wie stark der Operator bindet.

Je **kleiner** der Wert, desto **stärker** bindet der Operator.

Beispiel aus der Mathematik: $5 \cdot 3 + 2 = 17 \neq 25$ da \cdot stärker als $+$ bindet

Wiederholung ○○○○	Artemis oooooooooooo	Klassen und Objekte ○ ○ ○○○○	Operatoren ●○○○	Datentypen II ○○○○	Referenzen ○○○○	Scanner ○○	Ende ○○
----------------------	-------------------------	---------------------------------------	--------------------	-----------------------	--------------------	---------------	------------

		Präzedenz
unäres -/+		1
+	Addition	3
-	Subtraktion	3
*	Multiplikation	2
/	Division	2
%	Modulo (Rest bei Ganzzahldivision)	2

Präzedenz
Der Wert gibt an wie stark der Operator bindet.
Je **kleiner** der Wert, desto **stärker** bindet der Operator.
Beispiel aus der Mathematik: $5 \cdot 3 + 2 = 17 \neq 25$ da \cdot stärker als $+$ bindet

In-/Dekrement-Operator



- ++ erhöht Variable um eins
- verringert Variable um eins

Notationen:

Präfix	y = ++x	x = x + 1;	y = x;
Postfix	y = x++	y = x;	x = x + 1;

Analog für --

Vergleichs Operationen



Liefern Wahrheitswert (boolean):

		Präzedenz
<	kleiner	5
<=	kleiner-gleich (entspricht \leq)	5
>	größer	5
>=	größer-gleich (entspricht \geq)	5
==	Gleichheit	6
!=	Ungleichheit	6

2022-11-07

Liefern Wahrheitswert (boolean):		Präzedenz
<	kleiner	5
<=	kleiner-gleich (entspricht \leq)	5
>	größer	5
>=	größer-gleich (entspricht \geq)	5
==	Gleichheit	6
!=	Ungleichheit	6

Operationen auf boolean-Werten



		Präzedenz
!	Negation	1
&&	logisches Und	10
	logisches Oder	11

2022-11-07

		Präzedenz
!	Negation	1
&&	logisches Und	10
	logisches Oder	11

Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;  
Wert von c:
```



Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	●○○○	○○○	○○	○○

2022-11-07

Tutorium 14
└─ Datentypen II
└─ Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;  
Wert von c:
```

Ungenauigkeit double

double a = 2.0;

double b = 1.9;

double c = a - b;

Wert von c: 0.10000000000000009



Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	●○○○	○○○	○○	○○

2022-11-07

Tutorium 14

Datentypen II

Ungenauigkeit double

```
double a = 2.0;
double b = 1.9;
double c = a - b;
Wert von c: 0.10000000000000009
```

Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;
```

Wert von c: 0.10000000000000009

Zahlen sind nur **endlich** genau \Rightarrow *Rundungsfehler*



Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	●○○○	○○○	○○	○○

2022-11-07

Tutorium 14

Datentypen II

Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;  
Wert von c: 0.10000000000000009  
Zahlen sind nur endlich genau  $\Rightarrow$  Rundungsfehler
```

Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;
```

Wert von c: 0.10000000000000009

Zahlen sind nur **endlich** genau \Rightarrow *Rundungsfehler*

Achtung bei Vergleichen

Statt Vergleich mit `actual == expected`, geeignetes Delta verwenden:
`Math.abs(expected - actual) < delta`



2022-11-07

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;  
Wert von c: 0.10000000000000009  
Zahlen sind nur endlich genau  $\Rightarrow$  Rundungsfehler  
Achtung bei Vergleichen  
Statt Vergleich mit actual == expected, geeignetes Delta verwenden:  
Math.abs(expected - actual) < delta
```


Ungenauigkeit double

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;
```

Wert von c: 0.10000000000000009

Zahlen sind nur **endlich** genau \Rightarrow *Rundungsfehler*

Achtung bei Vergleichen

Statt Vergleich mit `actual == expected`, geeignetes Delta verwenden:
`Math.abs(expected - actual) < delta`

Hier: Statt `c == 0.1`, verwende `Math.abs(c - 0.1) < 0.0001`

Wiederholung ○○○○	Artemis ○○○○○○○○○○○○	Klassen und Objekte ○ ○ ○○○○	Operatoren ○○○	Datentypen II ●○○○	Referenzen ○○○	Scanner ○○	Ende ○○
----------------------	-------------------------	---------------------------------------	-------------------	-----------------------	-------------------	---------------	------------



2022-11-07

```
double a = 2.0;  
double b = 1.9;  
double c = a - b;  
Wert von c: 0.10000000000000009  
Zahlen sind nur endlich genau  $\Rightarrow$  Rundungsfehler  
Achtung bei Vergleichen  
Statt Vergleich mit actual == expected, geeignetes Delta verwenden:  
Math.abs(expected - actual) < delta  
Hier: Statt c == 0.1, verwende Math.abs(c - 0.1) < 0.0001
```



Was ist die Ausgabe des folgenden Codes?

```
byte a = 127;
Sytem.out.println(a++ + ", " + a);
```

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○	○ ○ ○○○○	○○○	●○○	○○○	○○	○○

Was ist die Ausgabe des folgenden Codes?

```
byte a = 127;
System.out.println(a++ + ", " + a);
```



Was ist die Ausgabe des folgenden Codes?

```
byte a = 127;
Sytem.out.println(a++ + ", " + a);
```

127, -128

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o o oooo	oooo	o●oo	oooo	oo	oo

Was ist die Ausgabe des folgenden Codes?

byte a = 127;
System.out.println(a++ + ", " + a);

127, -128

Was ist die Ausgabe des folgenden Codes?

```
byte a = 127;  
System.out.println(a++ + ", " + a);
```

127, -128

Warum ist das so?

Was ist die Ausgabe des folgenden Codes?

```
byte a = 127;  
System.out.println(a++ + ", " + a);
```

127, -128

Warum ist das so?

Der Wertebereich von byte geht von -2^7 (-128) bis $2^7 - 1$ (127).

Inkrementieren vom größten Wert

⇒ kleinster Wert

Dekrementieren vom kleinsten Wert

⇒ größter Wert

Wie ist bei Gleitkommazahlen?



Overflow:

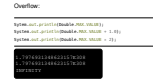
```
System.out.println(Double.MAX_VALUE);  
System.out.println(Double.MAX_VALUE + 1.0);  
System.out.println(Double.MAX_VALUE * 2);
```

```
1.7976931348623157E308  
1.7976931348623157E308  
INFINITY
```

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
○○○○	○○○○○○○○○○○○○○	○ ○ ○○○○	○○○○	○○●○	○○○○	○○	○○

2022-11-07

Tutorium 14
└─ Datentypen II
└─ Wie ist bei Gleitkommazahlen?



Wie ist bei Gleitkommazahlen?



Overflow:

```
System.out.println(Double.MAX_VALUE);  
System.out.println(Double.MAX_VALUE + 1.0);  
System.out.println(Double.MAX_VALUE * 2);
```

```
1.7976931348623157E308  
1.7976931348623157E308  
INFINITY
```

Underflow:

```
System.out.println(Math.pow(2, -1074));  
System.out.println(Math.pow(2, -1075));
```

```
4.9E-324  
0.0
```

Overflow:	Underflow:
<pre>System.out.println(Double.MAX_VALUE); System.out.println(Double.MAX_VALUE + 1.0); System.out.println(Double.MAX_VALUE * 2);</pre>	<pre>System.out.println(Math.pow(2, -1074)); System.out.println(Math.pow(2, -1075));</pre>
<pre>1.7976931348623157E308 1.7976931348623157E308 INFINITY</pre>	<pre>4.9E-324 0.0</pre>

Benennungs-Koventionen



Variablen, Attribute und Methoden	lowerCamelCase	userCounter
Klassen, Enums und Interfaces	UpperCamelCase	StringBuilder
Klassenkonstanten, Einträge in Enums	GROSS_GESCHRIEBEN	GRAVITATION_EARTH
Pakete	kleinbuchstaben (umgedrehte Domain)	com.java.util

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooooooo	o o o oooo	oooo	ooo●	oooo	oo	oo

2022-11-07

Variablen, Attribute und Methoden	lowerCamelCase	userCounter
Klassen, Enums und Interfaces	UpperCamelCase	StringBuilder
Klassenkonstanten, Einträge in Enums	GROSS_GESCHRIEBEN	GRAVITATION_EARTH
Pakete	kleinbuchstaben (umgedrehte Domain)	com.java.util

Speicherung



Abstrakter Speicheraufbau:

x	y	z	a	b	c
0	2	-2	1.6f	-2.5f	50
int	int	int	float	float	int
01	02	03	04	05	06

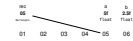
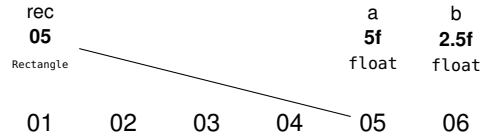
- Variablen werden an aufeinanderfolgenden Adressen im Speicher gespeichert
- Pro Adresse werden immer nur 8 Bits gespeichert
- int braucht eigentlich 4 Adressen (hier vereinfacht)



Speicherung von Objekten



- Für ein Objekt wird immer nur eine Referenz auf die Speicheradressen der Objektattribute gespeichert
- Die Objektvariable zeigt auf die Objektidentität
- Beispiel: Rectangle rec mit rec.a = 5f und rec.b = 2.5f



Zuweisung von Objekten

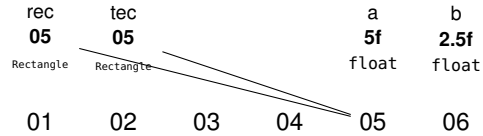


An der Situation von eben (Rectangle rec mit `rec.a = 5f` und `rec.b = 2.5f`) wird folgendes geändert:
`Rectangle tec = rec;`
Wie sieht der Speicher anschließend aus?

Zuweisung von Objekten



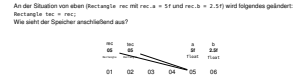
An der Situation von eben (Rectangle rec mit `rec.a = 5f` und `rec.b = 2.5f`) wird folgendes geändert:
`Rectangle tec = rec;`
Wie sieht der Speicher anschließend aus?



2022-11-07

Tutorium 14

- Referenzen
 - Speicherung
 - Zuweisung von Objekten



Spezialfall



null

- Wichtiges Element!
- `Rectangle rec = null;`
- Referenz auf nichts
- Es wird auf „kein Objekt“ referenziert
- Zugriff auf das Objekt führt zur Exception

null

- Wichtiges Element!
- `Rectangle rec = null;`
- Referenz auf nichts
- Es wird auf „kein Objekt“ referenziert
- Zugriff auf das Objekt führt zur Exception

Spezialfall

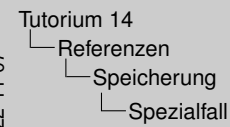


null

- Wichtiges Element!
- `Rectangle rec = null;`
- Referenz auf nichts
- Es wird auf „kein Objekt“ referenziert
- Zugriff auf das Objekt führt zur Exception

```
rec.a = 10;
```

2022-11-07



null

- Wichtiges Element!
- `Rectangle rec = null;`
- Referenz auf nichts
- Es wird auf „kein Objekt“ referenziert
- Zugriff auf das Objekt führt zur Exception

```
rec.a = 10;
```

Spezialfall



null

- Wichtiges Element!
- `Rectangle rec = null;`
- Referenz auf nichts
- Es wird auf „kein Objekt“ referenziert
- Zugriff auf das Objekt führt zur Exception

`rec.a = 10;`

⇒ `java.lang.NullPointerException`

2022-11-07

Tutorium 14
└─ Referenzen
 └─ Speicherung
 └─ Spezialfall

```
null
■ Wichtiges Element!
■ Rectangle rec = null;
■ Referenz auf nichts
■ Es wird auf „kein Objekt“ referenziert
■ Zugriff auf das Objekt führt zur Exception
rec.a = 10;
⇒ java.lang.NullPointerException
```

Scanner



Für Eingaben während das Programm läuft, verwende die Klasse Scanner:

Importieren mit `import java.util.Scanner`; vor der Klassendefinition!
Neuen Scanner erstellen mit: `new Scanner(System.in)`

Einlesen mit:

<code>nextLine()</code>	für Strings
<code>nextInt()</code>	für Ganzzahlen
<code>nextDouble()</code>	für Gleitkommazahlen
...	...

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o o oooo	oooo	oooo	oooo	●o	oo

2022-11-07

Tutorium 14
Scanner
Scanner

Für Eingaben während das Programm läuft, verwende die Klasse Scanner:
Importieren mit `import java.util.Scanner`; vor der Klassendefinition!
Neuen Scanner erstellen mit: `new Scanner(System.in)`
Einlesen mit:
`nextLine()` für Strings
`nextInt()` für Ganzzahlen
`nextDouble()` für Gleitkommazahlen
...

Scanner - Beispiel



```
import java.util.Scanner;

class ReadTerminal {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Gib eine Zahl ein!");
        int input = scanner.nextInt();
        System.out.println("Deine Zahl war " + input + ".");
    }
}
```

```
$ java ReadTerminal
Gib eine Zahl ein!
> 5
Deine Zahl war 5.
```

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooooooo	o o o oooo	oooo	oooo	oooo	o●	oo



Beispiel in Eclipse zeigen, import extra ansprechen

Wie weit sind wir?



02.11.2022
Orga
Java
Klassen und Objekte

09.11.2022
Datentypen
Operatoren
Referenzen

16.11.2022
Schleifen
Konstruktoren
JavaDoc
Kontrollstrukturen

Wiederholung	Artemis	Klassen und Objekte	Operatoren	Datentypen II	Referenzen	Scanner	Ende
oooo	oooooooooooo	o o oooo	oooo	oooo	oooo	oo	●o

2022-11-07

Tutorium 14
Ende
Wie weit sind wir?

Orga
02.11.2022
Klassen und Objekte
Datentypen
09.11.2022
Operatoren
Referenzen
Schleifen
16.11.2022
Konstruktoren
JavaDoc
Kontrollstrukturen

Bis zum nächsten Tutorium am 16.11.2022!

Wiederholung ○○○○	Artemis ○○○○○○○○○○○○○○	Klassen und Objekte ○ ○ ○○○○	Operatoren ○○○	Datentypen II ○○○	Referenzen ○○○	Scanner ○○	Ende ○●
----------------------	---------------------------	---------------------------------------	-------------------	----------------------	-------------------	---------------	------------