PROPÄDEUTIKUM INFORMATIK
SOSE 2018

Martin Mehlhose

ORGANISATORISCHES

ABLAUF PROPÄDEUTIKUM

- 14 Wochen Unterricht
- Freitags 13:30 - 15:00 Uhr
- Raum Lu4 Zi 17

ABLAUF PROPÄDEUTIKUM

- Propädeutikum kann mit 3 LP bewertet werden
- Programmieraufgabe als Leistungsnachweis
- regelmäßige Teilnahme an den Veranstaltungen vorrausgesetzt

ABLAUF PROPÄDEUTIKUM

Folien und Quellcode unter:

- https://github.com/Francisde/PropaedeutikumSoSe2018.git

Bei Fragen jeder Zeit eine E-Mail an:

- E-Mail: fm63byza@studserv.uni-leipzig.de

LITERATUR

- http://openbook.rheinwerk-verlag.de/javainsel/
- https://www.bioinf.uni-leipzig.de/Leere/WS1516/MUP/Vorlesung_I.zip
- https://www.bioinf.uni-leipzig.de/Leere/WS1516/MUP/Vorlesung_II.zip

INFOVERANSTALTUNG ZUR UNIBEWERBUNG

- Mittwoch, 18. April 2018 ab 13:30 Uhr in der Aula (Uni Leipizg)
- Mittwoch, 09. Mai 2018 ab 13:30 Uhr in der Aula (TU Dresden)

STOFFÜBERSICHT

- Einführung in die objektorientierte Programmierung mit Java
- Modellierung mit UML
- Maschinenzahlen und Zahlenkonvertierung
- Algorithmen zum arbeiten auf verschiedenen Datenstrukturen

JAVA

- 1991 - entwickeln Mike Sheridan, James Gosling, Patrick Naughton u.a. bei Sun Microsystems die Programmiersprache OAK (**Object Application Kernel)**, ursprünglich zur Steuerung und Integration von Haushaltsgeräten.
- 1993 - OAK ist klein, objektorientiert, platformunabhängig und robust und eignet sich für Internetanwendungen

JAVA

- 1994 - wird die Sprache in Java (starker Kaffee) umbenannt.
- 1995 - wird Java in die führenden Web-Browser Netscape Navigator und MicroSoft Internet Explorer integriert.

WARUM JAVA?

- **objektorientiert**, dadurch leicht zu verstehen und zu erweitern
- **robust und sicher**, durch Datenkapselung und aktive Fehlerbehandlung
- **architekturneutral**, undabhängig von Hardware und Betriebssystem
- **leistungsfähig**

EINSTIEG IN JAVA

AUFBAU

```
public class HelloWorld{
    public static void main(String[] args){
        // do anything
        System.out.println("Hello World!");
    }
}
```

- Klasse als Spezifikation für eine Menge von Objekten
- Main-Methode als Einstiegspunkt
- println Methode zur Ausgabe auf der Standardausgabe

DATENTYPEN

- im ersten Beispiel nur Ausgabe eines vorgegebenen Textes
- Ziel beliebige Daten eingeben, verarbeiten und ausgeben

GRUNDELEMENTE JAVA

GRUNDELEMENTE - IDENTIFIER

Um Variablen, Metoden, Klassen und Objekte ansprechen zu können werden diese mit Namen identifiziert.

- Regeln zum erstellen von Identifiern
  - Namen dürfen Buchstaben, Ziffern, Unterstrich "_" und Dollarzeichen "$"enthalten
  - Namen dürfen nicht mit Ziffer beginnen
  - Namen dürfen nicht mit Java Schlüsselwörtern übereinstimmen
  - Konvention: Variablen beginnen mit einem Kleinbuchstaben

JAVA SCHLÜSSELWÖRTERN

abstract	assert	boolean	break	byte
case	catch	char	class	const
continue	default	do	double	else
enum	extends	final	finally	float
for	goto	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	short	static	strictfp	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while

GRUNDELEMENTE - IDENTIFIER

Konventionen zur Namensvergabe:

- verwendung von **"sprechenden"** Namen
- **Variablen**namen beginnen mit einem Kleinbuchstaben
- Namen von **Klassen** beginnen mit Großbuchstaben
- bei zusammengesetzten Namen beginnt jedes Wort nach dem ersten mit einem Großbuchstaben
- Namen von **Konstanten** bestehen nur aus Großbuchstaben

BEISPIEL IDENTIFIER

```
public class Sample{
    public static void main(String[] args){
        int variable1 =2;
        int variable2 = 5;

        int ergebnisDerMultiplikation= variable1*variable2;

        System.out.println(ergebnisDerMultiplikation);

        final double EULERSCHEZAHL= 2.718281828459045;
    }
}
```

ELEMENTARE DATENTYPEN

GANZE ZAHLEN

- **byte** [-128,127]
- **short** [-32768,32767]
- **int** [-2147483648,2147483647]
- **long**
  [-9223372036854775808,9223372036854775807]

ELEMENTARE DATENTYPEN

GLEITKOMMAZAHLEN

- **float** $[10^{-45},10^{38}]$
- **double** $[10^{-324},10^{308}]$

LOGISCHE WERTE

- **boolean** Werte: true, false

ELEMENTARE DATENTYPEN

TEXT

Der Datentyp für einezelne Zeichen ist in Java **char**.
Bei der expliziten Definition einer Variable ist der Wert mit **' '** anzugeben.
Beispiel: char buchstabe ='B';
Zusammenhängende Zeichen und Sätze werden mit dem Datentyp **String** gespeichert. Der Wert ist dann mit **" "** anzugeben.
Beispiel: String satz="Das ist ein Satz";

ELEMENTARE DATENTYPEN

KOMMENTARE

Kommentare sind Bereiche im Quellcode die bei der Compoilierung und Programmausführung ignoriert werden. Sie dienen der Dokumentation des Quellcodes und sollen desen Lesbarkeit erhöhen.

ELEMENTARE DATENTYPEN

KOMMENTARE

- Einzeilige Kommentare
  - Kommentarbereich wird durch // markiert

    Alles was in der selben Zeile dem // folgt wird ignoriert
- Mehrzeilige Kommentare
  - Kommentarbereich wird durch /* ... */ markiert

    Alles was zwischen /* und */ steht wird ignoriert.

BEISPIEL KOMMENTARE

```
public class Sample{
    public static void main(String[] args){
        int variable1;          // einzeiliges Kommentar

        //int a=0.5;

        /*
        * Das ist ein mehrzeiliges Kommentar
        */

    }
}
```

VARIABLEN IN JAVA

Variablen sind Platzhalter im Speicher (Ähnlich wie in der Mathematik). Jede Variable hat einen **Namen**, einen **Typ** und (eventuell) einen **Wert**.

- Deklarieren

  Zuordnung eines Namen und Typ
  Beispiel: int myInteger;

- Initialisieren

  Zuweisung eines Wertes
  Beispiel: myInteger=25;

BEISPIEL VARIABLEN

```
public class Sample{
    public static void main(String[] args){
        int variable1;          // Deklarieren der Variable
        variable1 = 25;      // Initialisieren der Variable

        // Deklarieren mehrerer Variablen ist möglich
        int a,b,c;

        // Deklarieren und Initialisieren gleichzeitig
        int variable2 =50;
    }
}
```

GRUNDLEGENDE RECHENOPERATIONEN IN JAVA

- Additions-Operator: +
- Subtraktions-Operator: -
- Divisions-Operator: /
- Multiplikations-Operator: *
- Rest-Operator: %

  Klammerung wie beim "normalem" rechnen.

JAVA MATH BIBLIOTHEK

Weitere Mathematische Operationen in der Math Bibliothek. Klammerung wie beim "normalem" rechnen.

- Wurzelfunktion: **sqrt(x);**
- Potenzfunktion: **pow(a,b);** -> repräsentiert a^b. Achtung! Für Potenzfunktion nicht a^b benutzen.
- diverse trigonometrische und andere Funktion

IOTOOL

Bibliothek zum auslesen der Standardeingabe (Tastatur)

- **readString(String x);**
- **readInt(String x);**
- **readDouble(String x);**

KONTROLLSTRUKTUREN UND SCHLEIFEN

BOOLESCHE OPERATIONEN

- Negation: **!**
- Logische "und": **&&**
- Logische "oder": ||
- auch hier wieder Klammerung für die Auswertungsreihenfolge beachten

VERGLEICHSOPERATOREN

- gleich: ==
- ungleich: !=
- größergleich: >=
- kleinergleich: <=
- strikt größer: >
- strikt kleiner: <

IF ANWEISUNG

- Syntax: **if(logischer Ausdruck) {statement}**
- Auf eine if Anweisung können mehrere Statements folgen
- Wichtig: Klammerung beachten!

- Syntax: **if(logischer Ausdruck) {statement} else {statement}**

BEISPIEL IF ANWEISUNG

```
// einfache if Anweisung
if(a<b){
    max=b;
}

// if-else Anweisung
if(a<b){
    max=b;
} else {
    max=a;
}
```