

zum Temperaturprogramm

Programmtext in Datei abspeichern

muss Namen der Klasse haben + Endung .java

class Temperatur => Dateiname Temperatur.java

Dabei kompiliert durch Aufruf

javac Temperatur.java

Erzeugt Dateien mit Java Bytecode mit Endung .class

(eine .class Datei für jede Klasse,
auch für innere Klassen)

Ausführung des Applets nur über HTML Dateien

mit Anweisung für Start des Applet

siehe Internet / Come CD

haben wichtige Begriffe kennengelernt

- Variable
- Zuweisung
- Klasse
- Methode

Variable = Name (Platzhalter) für Daten / Objekte
eines bestimmten Typs

double celsius; ← Deklaration
↑ ↑
Typ Name

Compiler legt Speicherplatz an für eine double Zahl,
der dann unter dem Namen celsius angesprochen
werden kann

TextField input;
↑ ↑
Typ Name

Zuweisung aktualisiert den Wert einer Variable
(d.h. den Inhalt des zugehörigen
Speicherplatzes)

double fahrenheit, celsius;

fahrenheit	<input type="text" value="0.0"/>
celsius	<input type="text" value="0.0"/>

Eingabe von 95 in Input Feld

fahrenheit = Double.parseDouble (input.getText());

↑
Zuweisung an
die Variable
fahrenheit

String "95"
↙
verwandelt String
in double Zahl 95.0

fahrenheit	95.0
celsius	0.0

celsius = (5.0 / 9) * (fahrenheit - 32);

↑
Zuweisung an
Variable celsius

(arithmetischer) Ausdruck
ausgewertet, ergibt
den Wert 35.0

fahrenheit	95.0
celsius	35.0

Ein Ausdruck hat einen Wert

Eine Zuweisung hat einen Effekt (Änderung im Speicher)

Klasse = Zusammenfassung von Objekten
desselben Typ mit zugehörigen
Operationen / Methoden, die mit den
Objekten etwas machen

TextField ist Klasse

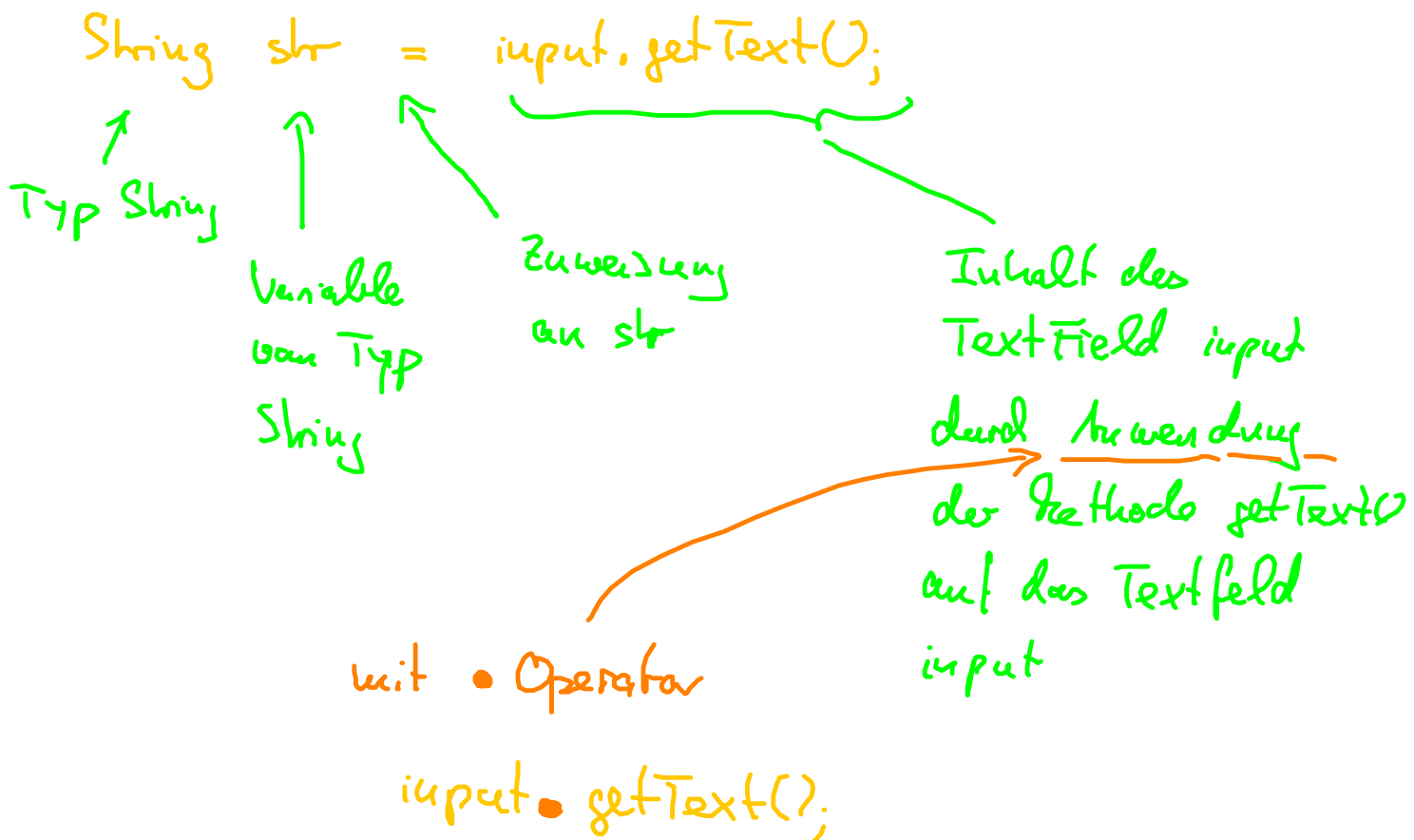
Objekte sind Textfelder

Methoden sind z.B. `getText()`;
`setText()`;

`TextField input;` deklariert Textfeld

`input = new TextField(10);`

erzeugt neues Textfeld mit 10 Zeichen Länge
dieses ist unter Namen `input` ansprechbar



input.setText(str);

input Hallo

Konstruktor und Methoden aus der Klasse nehmen ohne sie auf Objekte anzuwenden

Math.round(...);

↑ ↑
Klasse Math Methode round aus dieser Klasse

long z = Math.round(3.14);

↑ z 3

Datentyp für lange ganze Zahlen

2.2 Einkommensteuerberechnung

2.2.1. Das Problem

einfaches Stufenmodell à la Herz

Einkommen E

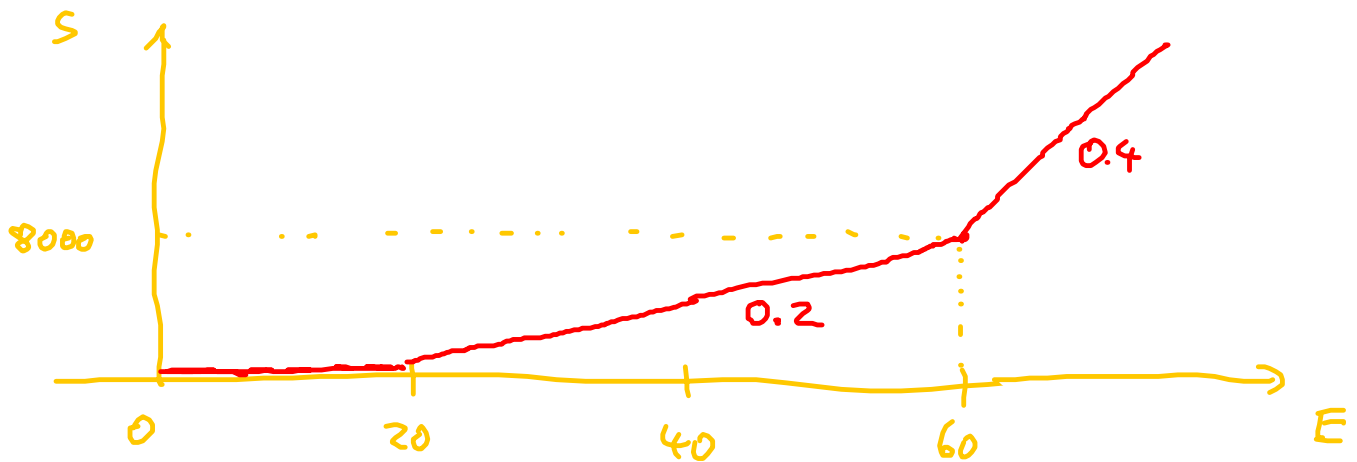
$E \leq 20.000 \text{ €}$	→ 0 Steuern	12%
$20.000 < E \leq 60.000 \text{ €}$	→ 20% von $E - 20.000$	24%

$$E > \overset{40.000}{60.000} \text{ €} \quad \rightarrow \quad 8.000 + 40\% \text{ von } \overset{35\%}{E - 60.000}$$

Merz Modell:

2.2.2. Der Algorithmus

Modell \Rightarrow Steuer S ist stückweise lineare Fkt des Einkommens E



Algorithmus

1. Einlesen von E
2. Berechnung von S
3. Ausgabe von S

Programm siehe Zettel / Cone CD / Internet

neue Elemente:

- Konstanten
- if Anweisung
- Panel

- Layout - Anweisungen

Konstanten immer über Bezeichner (Namen) ausgedr.
in Java mit Schlüsselwort final gekennzeichnet

Bezeichner: Strings in Unicode

↑
16 bit
erlaubt sehr viele Sprachen
müssen mit Buchstaben beginnen
und nur aus Buchstaben + Ziffern + _
bestehen

input_1, Möhning,
~~3~~ ~~Felder~~, ~~no~~ ~~Tax~~ ~~Bound~~;

if Anweisung

if (Bedingung) Anweisung₁ else Anweisung₂

↑
Boolesche Bedingung
ergibt Wert von
Typ boolean
einzigste Werte sind true, false
bei true wird Anweisung₁ ausgeführt,
sonst Anweisung₂ darf fehlen

Sowst Anweisung₂

Anweisung₁ bzw. Anweisung₂ kann wieder eine
if Anweisung sein \Rightarrow Schleife

daher { ... } verwenden

Panel : unterteilt Applet in verschiedene Teile
die mit Layout Anweisungen geeignet
auf Applet verteilt werden können

Kann ihr Layout und Farbe usw. ändern

Ausgabe erfolgt über Label

Branden Methode validate() damit Applet das Label
neu zeichnet, wenn sich die Länge ändert.