UNIVERSITY OF TWENTE.

APPLICATION DEVELOPMENT

LECTURE 3: DESIGN A CLASS, USING OBJECTS AND METHODS, CONDITIONS AND LOOPS

```
class AppDev {

}
```

Part of SmartProducts

INTRODUCTION

APPLICATION DEVELOPMENT

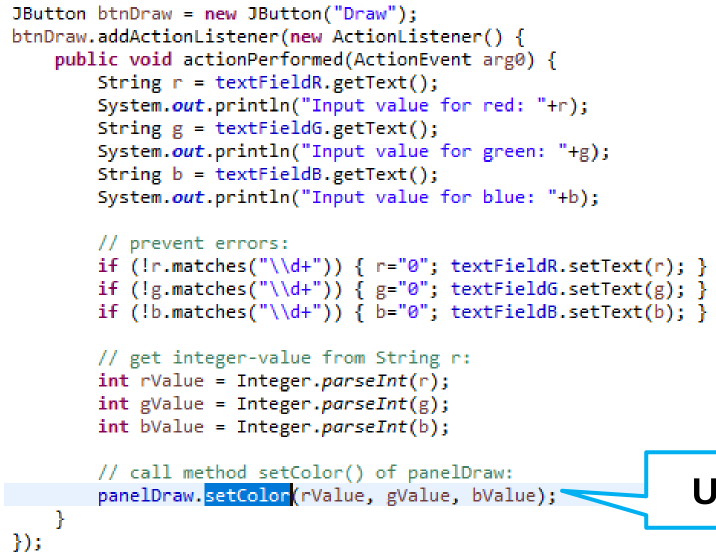
- Design a class
- Using objects and methods
- Conditions and loops
- Assignment

Fjodor van Slooten
W241 *(Horst-wing West)*
f.vanslooten@utwente.nl

slides @ vanslooten.com/appdev

ASSIGNMENT 2

- Adding a variable and a method
- A method declaration (definition) and it's use (call)



```
JButton btnDraw = new JButton("Draw");
btnDraw.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String r = textFieldR.getText();
        System.out.println("Input value for red: "+r);
        String g = textFieldG.getText();
        System.out.println("Input value for green: "+g);
        String b = textFieldB.getText();
        System.out.println("Input value for blue: "+b);

        // prevent errors:
        if (!r.matches("\\d+")) { r="0"; textFieldR.setText(r); }
        if (!g.matches("\\d+")) { g="0"; textFieldG.setText(g); }
        if (!b.matches("\\d+")) { b="0"; textFieldB.setText(b); }

        // get integer-value from String r:
        int rValue = Integer.parseInt(r);
        int gValue = Integer.parseInt(g);
        int bValue = Integer.parseInt(b);

        // call method setColor() of panelDraw:
        panelDraw.setColor(rValue, gValue, bValue);
    }
});
```

```
public class DrawingPanel extends JPanel {

  Color drawColor = Color.yellow;

  protected void paintComponent(Graphics g) {
    super.paintComponent(g);
         ...
  }

  public void setColor(int r, int g, int b) {
    drawColor = new Color(r % 256, g % 256, b % 256);
    // draw again because the color has been changed:
    repaint();
  }
```

Declaration

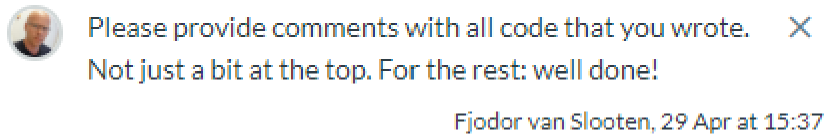
Use (call)

Find declaration? Select and press F3
(or right-click)

FEEDBACK & ASSIGNMENTS

Grade on Canvas: 1 = Pass, 0 = Fail

- Read the feedback on Canvas!



Please provide comments with all code that you wrote. Not just a bit at the top. For the rest: well done!

Fjodor van Slooten, 29 Apr at 15:37

- You *might* receive a request for an additional check:
- This does not 'mean' anything: it is just an extra check we do randomly with about 10% of the students

"You have been selected to give an additional live demo and/or answer questions via a video chat. On the next lecture day, please ask for this check. You must do this to pass the assignment!"

There is no time limit for this, but **we urge you to do it on the next lecture day**: otherwise you might have forgotten the meaning of code you wrote and fail the test.

- If we question the authenticity of your work, we might request a check also, but then the message will be different

DESIGN A CLASS
ANALYZE OBJECT (IN REAL WORLD)

Properties ↓

Position (x,y)
Diameter
Color

Ball

Actions/behavior ↓

Move
Bounce (change direction)
Draw

(methods)

What (is)? → Properties
What can … do? → Methods
How can … do? → (Pseudo) code (of methods)

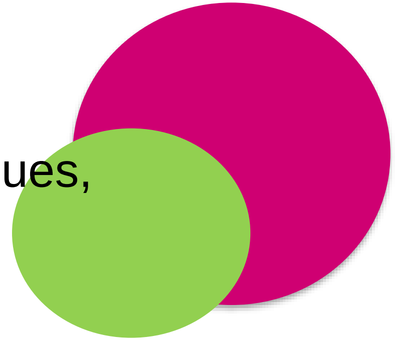
UNIVERSITY OF TWENTE.

DESIGN A CLASS

DETAIL CLASS IN (PSEUDO) CODE

Pseudo code: incomplete code, human-readable

- Types of properties
- Methods: return values, parameters



```
public class Ball {
  // properties:
  int x, y; // position
  int diameter;
  Color color;

  // methods:
  public void move();
  public void bounce();
  public void draw(Graphics g);
}
```

Parameter **g**, of type **Graphics**, needed for drawing

Standard return value: **void** (=nothing), and modifier **public**

DESIGN A CLASS

DETAIL METHODS IN (PSEUDO) CODE

Pseudo code: incomplete code, human-readable

Next step: start coding

- For each method:
- Write steps in pseudo code
- If new variables/properties are needed, alter design

```
public void bounce() {
  reverse direction:
  dx = -dx
  dy = -dy
}
```

```
public void move() {
  increase position (x,y)
  // by what? introduce dx/dy? (delta x and y)
}
```

```
public void draw(Graphics g) {
  set color
  draw filled circle at position (x,y)
}
```

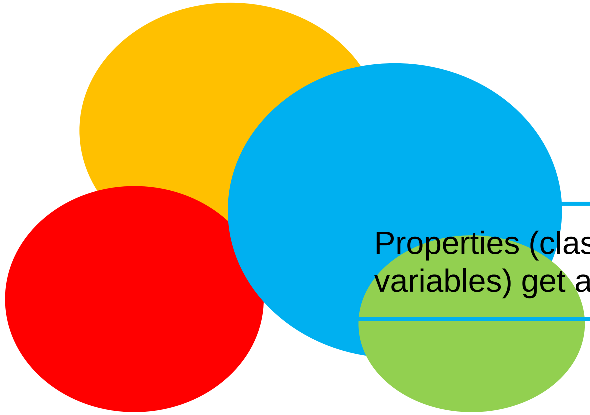
CREATE BALLS: NEW

b1 is a new object of type **Ball:** "b1 is a Ball"

```
Ball b1 = new Ball(10, Color.orange, 10, 20);
Ball b2 = new Ball(8,  Color.red,     5, 30);
Ball b3 = new Ball(15, Color.blue,   20, 25);
Ball b4 = new Ball(5,  Color.green,  30, 30);
```

Call to constructor

Parameters determine difference in properties



Properties (class variables) get a value

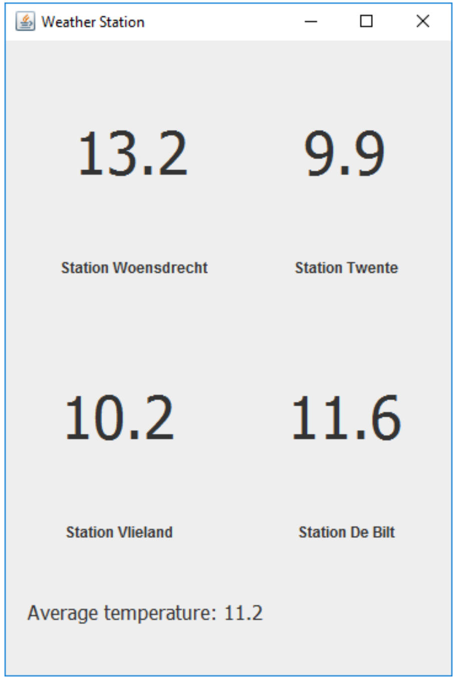
```
// constructor assigns properties:
public Ball(int d, Color c, int i, int j) {
  diameter = d;
  color = c;
  x = i;
  y = j;
}
```

Constructor: special method with same name as class and no return value

USING OBJECTS

- Assignment: weather-panels, create a class once, use 4 times

```
panel = new TemperaturePanel(6340);

panel2 = new TemperaturePanel(6290);

panel3 = new TemperaturePanel(6260);

panel4 = new TemperaturePanel(6235);
```



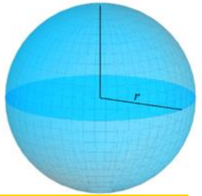
Average temperature: 11.2

METHODS: RETURN VALUE

Surface area [edit]

The surface area of a sphere is:

$$A = 4\pi r^2.$$



en.wikipedia.org/wiki/Sphere

Type of result: **double**

Parameter

```
public double calculateSurfaceArea(double r) {
   double A;
   A = 4 * Math.PI * Math.pow(r,2);
   return A;
}
```

Return the value (to caller)

Calculation

Use the method:

Parameter passed to method

```
double result = calculateSurfaceArea(10);
```

Call of the method

More info

‘CALL’ A METHOD

WITH TEXT AS A PARAMETER

Variable **temp** gets a value

‘Call’ method **readTemperature()** of object ***w***

“Dear weather station, please read the temperature for us”

```
String temp = w.readTemperature();

labelTemp.setText(temp);
```

“Show temperature in userinterface”

Call method **setText** to show String temp in a label

Variable **temp** is used as a parameter in a method-call

UNIVERSITY OF TWENTE.

CONDITIONS

IF …

Condition between (…)

```
int x = 3;
if (x == 3) {
   System.out.println("x must be 3");
}
```

(Conditional or Boolean) Operators

<	smaller?
<=	smaller or equal?
>	larger?
>=	larger or equal?
==	equal?
!=	not equal?

x=5	x gets value 5 (assignment)
x==5	does x equal 5 ?

CONDITIONS

IF … ELSE …

```
int age = 14;
int length = 110;

if (age < 10 && length > 110)
  System.out.println("You are a tall kid");
else if (age > 10 && length <= 110)
  System.out.println("Eat more bananas!");
else
  System.out.println("I guess you are Ok");
```

More info

Logical operators

&&	and
||	or
!	not

Use to build boolean expressions.
Result is *true* or *false*.

UNIVERSITY OF TWENTE.

CONDITIONS

SWITCH

```
switch(x) {
  case 1:
             soundbite = new File("cat.wav"); break;
  case 2:
             soundbite = new File("chicken.wav"); break;
  case 3:
             soundbite = new File("cow.wav"); break;
  case 4:
             soundbite = new File("dog.wav"); break;
  case 5:
             soundbite = new File("frog.wav"); break;
  default:
             soundbite = new File("bird.wav");
}
```

`default`: if none of the options complies

More info

REPEAT: LOOPS

WHILE …

More info

Condition between (…)

```
Dog rufus = new Dog();

int x = 0;
while (x < 3) {
  rufus.bark();
  x = x + 1;
}
```

How many times does rufus bark?

Condition depends on x!

UNIVERSITY OF TWENTE.

REPEAT: LOOPS

FOR …

Control variable

Condition

Step for control variable

More info

```
for (int l=0; l<4; l++) {
  System.out.println("Line "+l);
}
```

What is the output of these loops?

```
for (int l=0; l<8; l++) {
  for (int c=0; c<l; c++)
         System.out.print("#");
  System.out.println("");
}
```

increase a variable x by one: x++
same as: x = x + 1

FORMAT OUTPUT

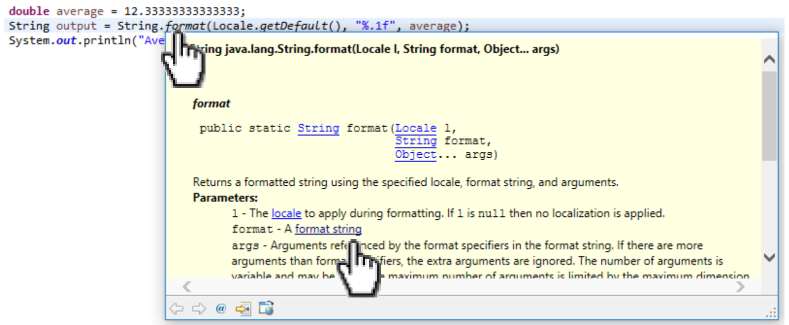
STRING.FORMAT

Format according to default locale settings of computer

```
double average = 12.33333333333333;

String output = String.format(Locale.getDefault(), "%.1f", average);

System.out.println("Average: "+output);
```

Format as a decimal number with one digit after the . (.1f)

Input parameter(s)



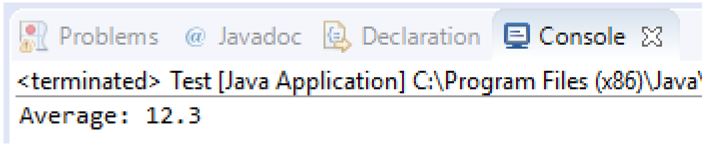
```
double average = 12.33333333333333;
String output = String.format(Locale.getDefault(), "%.1f", average);
System.out.println("Ave
```

ring java.lang.String.format(Locale l, String format, Object... args)

format

```
public static String format(Locale l,
                            String format,
                            Object... args)
```

Returns a formatted string using the specified locale, format string, and arguments.
Parameters:
l - The locale to apply during formatting. If l is null then no localization is applied.
format - A format string
args - Arguments ref[illegible]nced by the format specifiers in the format string. If there are more arguments than form[illegible]ifiers, the extra arguments are ignored. The number of arguments is variable and may be [illegible] maximum number of arguments is limited by the maximum dimension



```
<terminated> Test [Java Application] C:\Program Files (x86)\Java\
Average: 12.3
```

SCOPE OF VARIABLES

{ Scope: region in code where a variable (or object) is valid. }

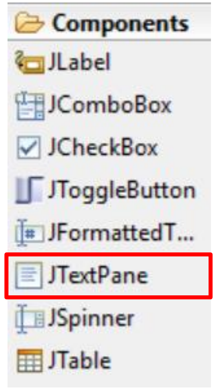
```
public class TemperaturePanel extends JPanel {
  WeatherStation w;

  public TemperaturePanel(int id) {
      JLabel labelTemp = new JLabel("25.7");

      w = new WeatherStation(id);

      String temp = w.readTemperature();

      labelTemp.setText(temp);
  }

  public double getTemperature() {
      String temperature = w.readTemperature();
      ...
  }
}
```

Object **`w`** is a *class-variable* in class **`TemperaturePanel`**

`temp` is a *local variable* valid in the constructor

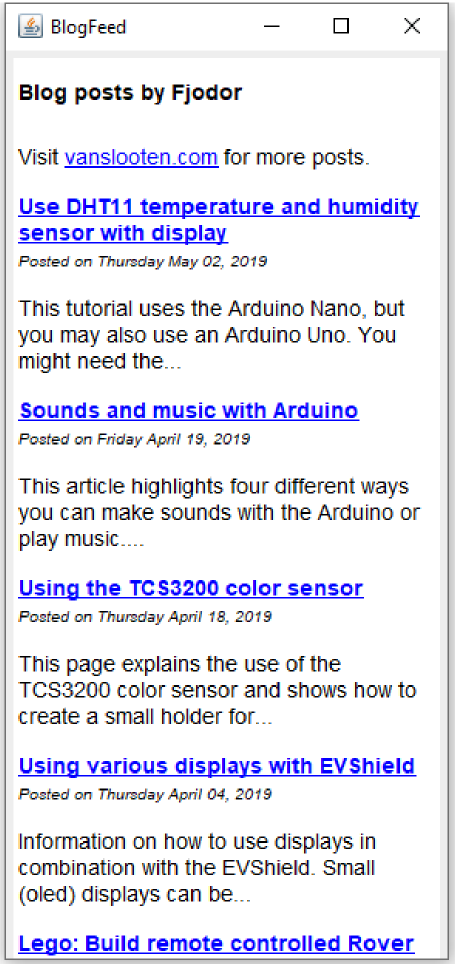
Object **`w`** can be used by all methods

LIBRARIES



- **`java.net, org.w3c, javax.xml`**: Libraries for internet applications & XML
- XML: Extensible Markup Language (Standard Data Exchange)
- Show webpage in textPane:

```
try {
   textPane.setPage("https://home.et.utwente.nl/slootenvanf/feed.php");
} catch (IOException e) {
   e.printStackTrace();
}
```



BlogFeed

Blog posts by Fjodor

Visit vanslooten.com for more posts.

Use DHT11 temperature and humidity sensor with display

Posted on Thursday May 02, 2019

This tutorial uses the Arduino Nano, but you may also use an Arduino Uno. You might need the...

Sounds and music with Arduino

Posted on Friday April 19, 2019

This article highlights four different ways you can make sounds with the Arduino or play music....

Using the TCS3200 color sensor

Posted on Thursday April 18, 2019

This page explains the use of the TCS3200 color sensor and shows how to create a small holder for...

Using various displays with EVShield

Posted on Thursday April 04, 2019

Information on how to use displays in combination with the EVShield. Small (oled) displays can be...

Lego: Build remote controlled Rover

CREATE (DESIGN) A METHOD YOURSELF

1. Think of a name (**what** should the method do?)
2. Write Pseudo code
3. Think of things the method should do (**how** does the method …?)
4. Does the method have to return something? (a result)

CREATE (DESIGN) A METHOD YOURSELF

TODAYS ASSIGNMENT

Add a method to TemperaturePanel which gets the temperature as a number

Step 1:

1. Think of a name for the method (**what** should the method do?) → get temperature
2. Write Pseudo code →
3. Think of things the method should do (**how** does the method …?) →

```
get temperature() {
    get temperature from weather station
    convert temperature to number
}
```

4. Does the method have to return something? (a result) → yes, a number

What is the type of the number?

Step 2:

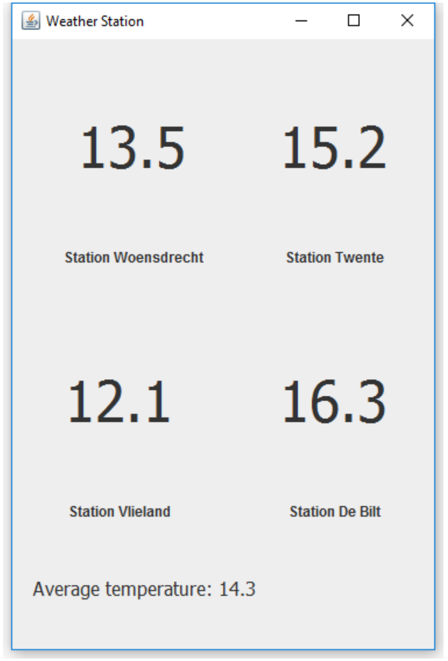
```
public ??? getTemperature() {
    // get temperature from weather station
    // convert temperature to number
    return number;
}
```

Step 3: put method into the class

add code at the 2 lines of comments

ASSIGNMENT #3

Deadline of each assignment is the next session
This one: May 8



13.5 15.2

Station Woensdrecht Station Twente

12.1 16.3

Station Vlieland Station De Bilt

Average temperature: 14.3

- “Create an application that can show weather-data from multiple weather stations”
- Extra challenge & appendix: get temperature from connected Arduino
- Try examples/self-study

Slides, assignments etc @ vanslooten.com/appdev