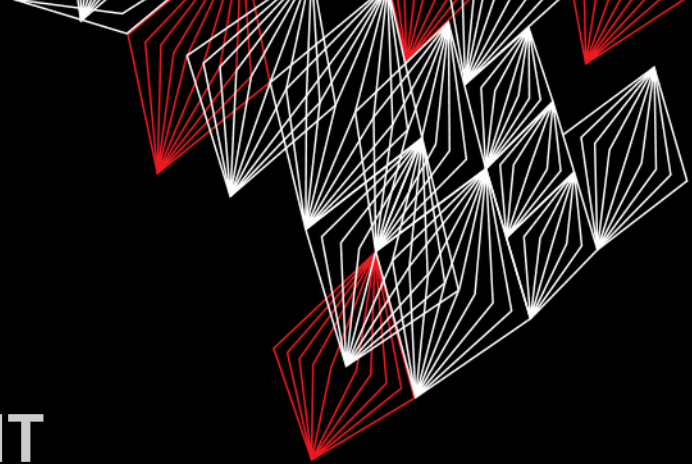


UNIVERSITY OF TWENTE.



APPLICATION DEVELOPMENT

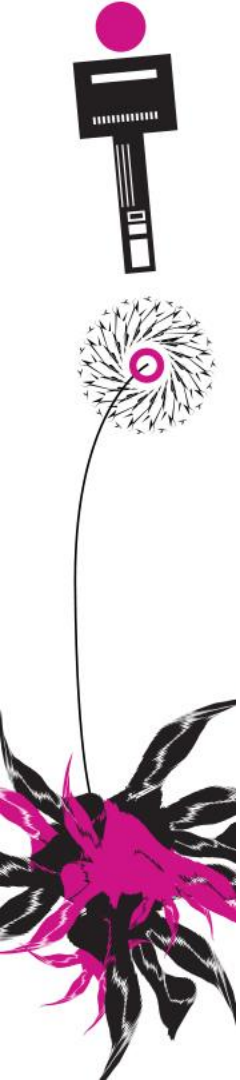
LECTURE 1: INTRODUCTION TO JAVA AND ECLIPSE

```
class AppDev {
```



Part of **SmartProducts**

```
}
```



INTRODUCTION

APPLICATION DEVELOPMENT



- Intro to:
 - Java
 - Eclipse



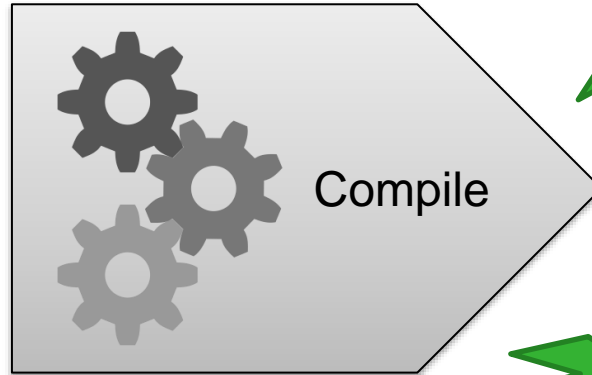
Fjodor van Slooten
W241 (*Horst-wing West*)
f.vanslooten@utwente.nl

slides @ vanslooten.com/appdev

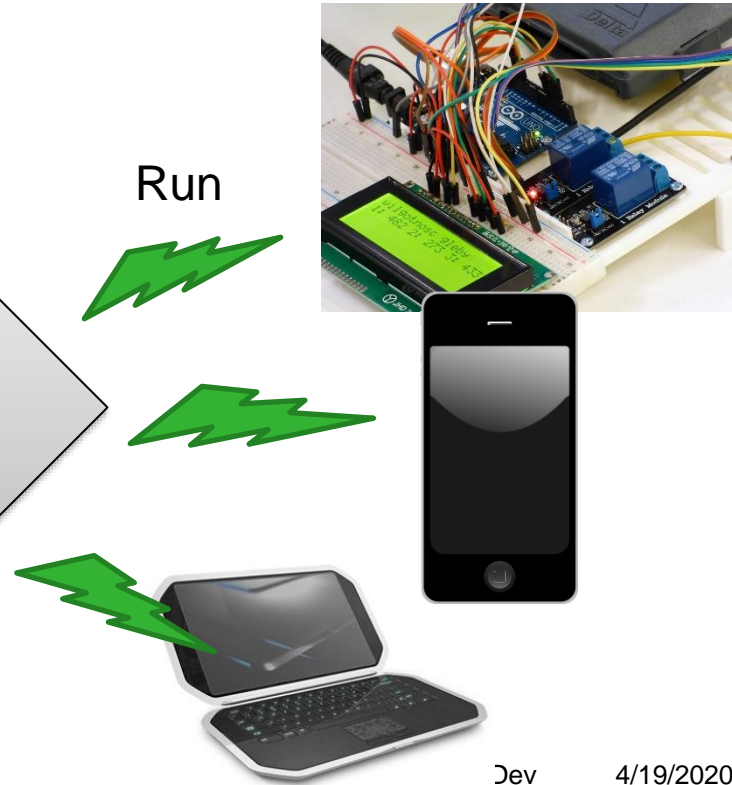
INTRODUCTION TO JAVA

Java

```
source code;  
class LetsParty {  
do interactive;  
while  
(wholeNight)  
}
```



Run



CLASS: DEFINES OBJECT

```
class Dog {  
    // properties:  
    int hairLength;  
    int age;  
  
    // methods:  
    run();  
    bark();  
    sit();  
}
```

Properties:
things an
object
has/knows/
stores

In Java, these
properties can
be variables or
objects

Methods: things
an object can **do**

Map



Object

A NEW DOG

```
// let's create a new Dog:  
Dog rufus = new Dog();
```

rufus is an object,
defined by a class



```
// let's make rufus do something:  
rufus.sit();  
rufus.bark();
```

METHODS

```
type name() {  
    // body  
}
```

- Describe what an object can do
- Contains set of instructions (code)
- May have input- and output parameters



head
body
between { }

```
// method which takes care of running:  
void run(int speed) {  
    // code which drives motors  
}  
  
// method which takes care of barking:  
void bark(int volume) {  
    speaker.play("BARK.mp3", volume);  
}
```

CODE STRUCTURE

[Java Cheat Sheet on site](#) →

```
/**
 * Comment
 * @author
 */

import java-library;

class MyFirstApp {

    public static void main(String[] args) {
        // body of method
        // contains instructions
        System.out.println("Hello world!");
    }
}
```

Opening bracket: start of code

main-method is start of program

Each line of code ends with a **semicolon ;**

Closing bracket: ends code block

source file (.java)

class

```
method1
statement;
```

```
method2
statement;
statement;
```

What goes in a source file?

```
public class Dog {

} // class
```

What goes in a class?

```
public class Dog {
    void bark() {

} // method
}
```

What goes in a method?

```
public class Dog {
    void bark() {
        statement1;
        statement1;
    }
}
```

NAMING CONVENTIONS

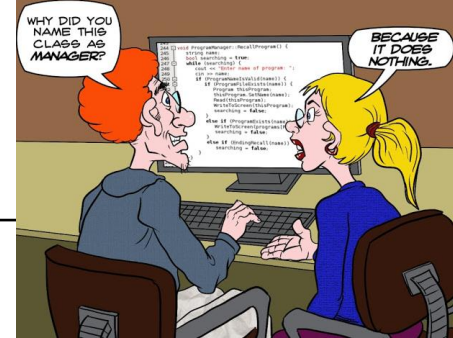
NAMES SHOULD...

Class name: starts uppercase (capital)

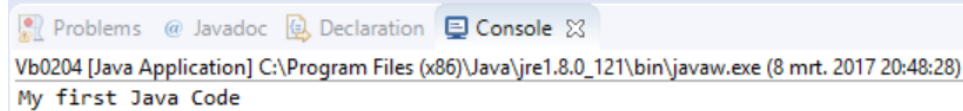
```
class TemperatureSensor {  
    // properties:  
    double temperature;  
    int age;  
  
    // methods:  
    double getTemperature() {}  
    void probe() {}  
    void reset() {}  
}
```

Variable & method
names: start
lowercase

- be short yet meaningful
- not start with a number
- not contain special-characters or spaces



TEXT OUTPUT



Head First: p6-13 Aan de slag met: 5.2.3

- Print a line of text:

```
System.out.println("My first Java Code");
```

Print (a line) to standard output

Text between quotes is a String (of characters)

- Print *good morning...* if hour is smaller than 12:

```
if ( LocalDateTime.now().getHour() < 12 )  
    System.out.println("Good morning");
```

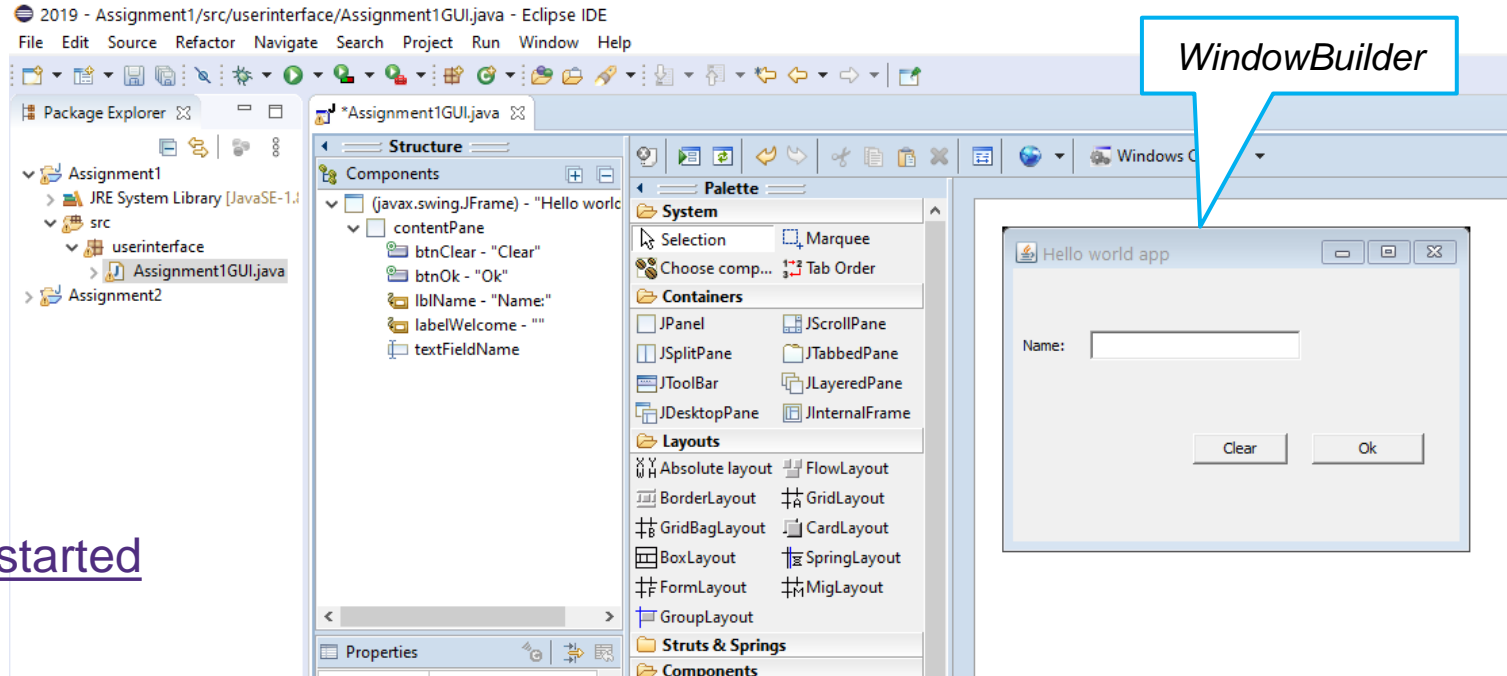


ECLIPSE

INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

- Create and organize applications
- WindowBuilder: design graphical userinterfaces

 [Watch "Getting started with Eclipse"](#)

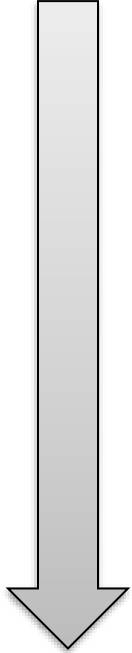


DESIGN AN APPLICATION

analyze > design >> implement

- Analyze: sketch ideas, think (out loud), find (sub) questions, (re) search, specifications
- Design: sketch (concepts), describe (functions, methods), **map (real world→class)**, pseudo code
- Implement : **write code**, draw (with computer), detail

top



down

Next week: more on designing apps