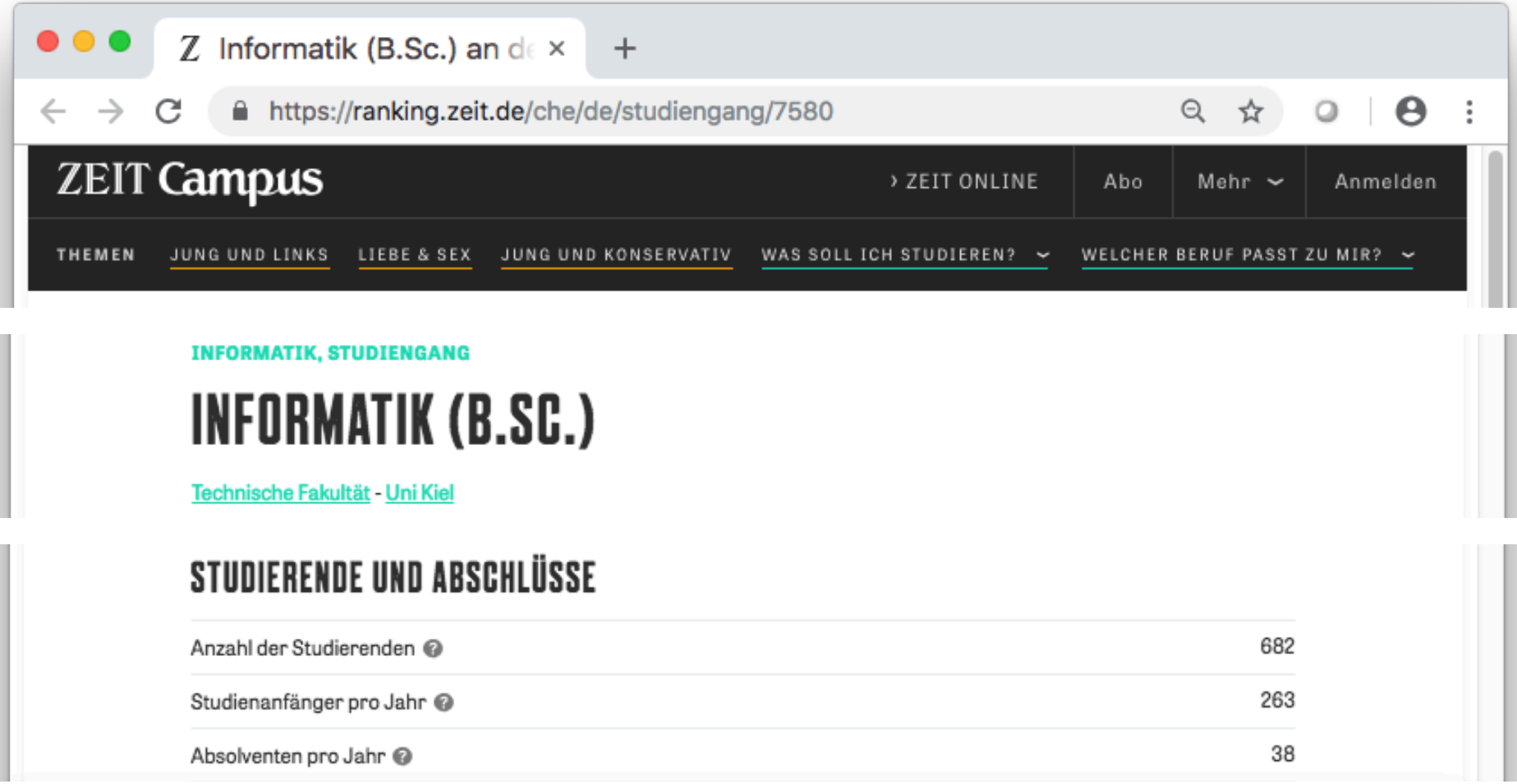
Five-Minute Review

1. What steps do we distinguish in solving a problem by computer?
2. What are essential properties of an *algorithm*?
3. What is a definition of *Software Engineering*?
4. What types of *programming error* do we distinguish?
5. What is the difference between *compilation* and *interpretation*?

Where This May Lead ...



Studienanfänger/Absolventenzahlen (lt. CHE Ranking)

Informatik:	263 / 38
Wirtschaftsinformatik:	49 / 17
Mathematik:	91 / 13

- Mag sein, dass das *Universitätsstudium* der Informatik nicht der beste Weg für *Sie* ist ...

- Aber es gibt alternative Wege, Informatiker*innen werden **nicht** nur als Uni-Absolventen gesucht!

Wer ist Ihr Pinguin? *

Am 05.09.18 um 21:31 schrieb stuXXXXXX@mail.uni-kiel.de:

... Außerdem habe ich ein gutes Jobangebot bekommen und werde die Uni verlassen. Ich halte dies für den besten Weg mich endlich zu entfalten, da der akademische Weg für mich eine Qual war. Ich wünsche ihnen und der Arbeitsgruppe alles Gute und **weisen Sie bitte die Erstsemester auf den hohen theoretischen Aspekt der Uni hin und empfehlen ihnen auch den Blick zur Ausbildung**.

* Google „Hirschhausen Pinguin“

Login | Registrieren

Mein Lebenslauf | Jobs suchen | Firmen entdecken | Für Arbeitgeber ›

Was? fachinformatik

Wo? Kiel, Deutschland

Umkreis 50 km

Suchen

Aktuelle Suche

- fachinformatik
- Kiel, Deutschland

Neue Suche starten

Art der Anstellung

Vollzeit (27)

Unternehmen

REMONDIS GmbH & Co KG (10)
FERCHAU GmbH Niederlassung Kiel (2)
ACO Severin Ahlmann GmbH & Co. KG (1)
CITTI Handelsgesellschaft & Co. KG (1)
CoCoProPersona GmbH (1)
mehr »

Region

Kiel (10)
Melsdorf (8)
Flintbek (2)
Rendsburg (2)
Schleswig (2)
mehr »

Automatisch neue Jobs per E-Mail erhalten? Jetzt Jobs per E-Mail empfangen!

Ausbildung zum Fachinformatiker / Systemintegration (m/w/d) (Fachinformatiker/in - Systemintegration)
30.10.2020, Heinrich Karstens GmbH & Co.KG
Kiel

Ausbildung Fachinformatiker*in für Systemintegration (m/w/d) (Fachinformatiker/in - Systemintegration)
27.10.2020, Deutsche Telekom AG Ausbildungszentrum Kiel
Kiel

Fachinformatiker (m/w/d)
04.10.2020, expertum Holding GmbH
Kiel

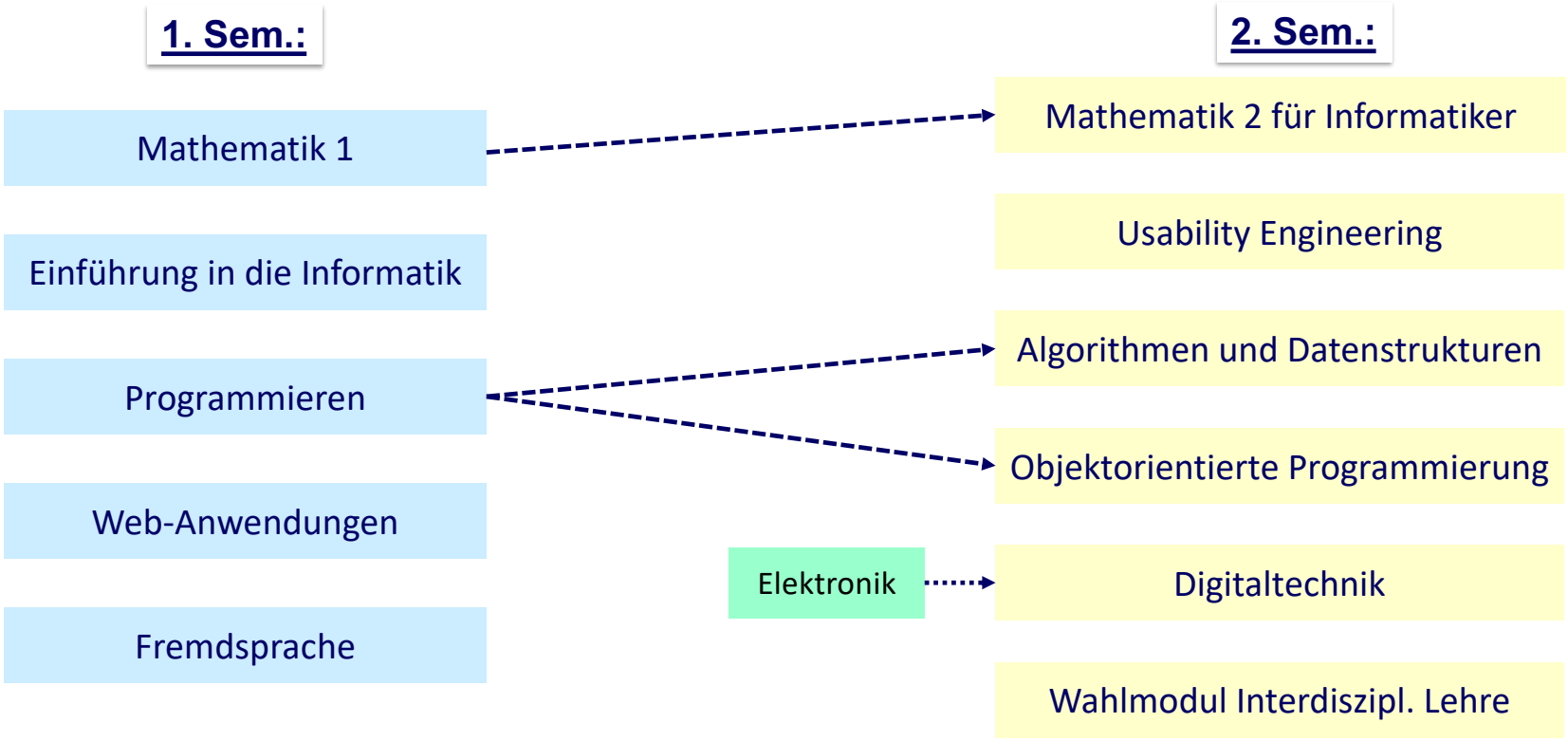
Ausbildung zum Fachinformatiker/in (m/w/d) 2021 (Fachinformatiker/in - Anwendungsentwicklung)
30.09.2020, Die Netzwerkstatt Dirk Meinke und Sven Probst GmbH & Co. KG
Rendsburg - Distanz: 30km

Fachinformatiker/in Fachrichtung Systemintegration (m/w/d)
29.09.2020, Schleswig-Holstein GmbH
Kiel

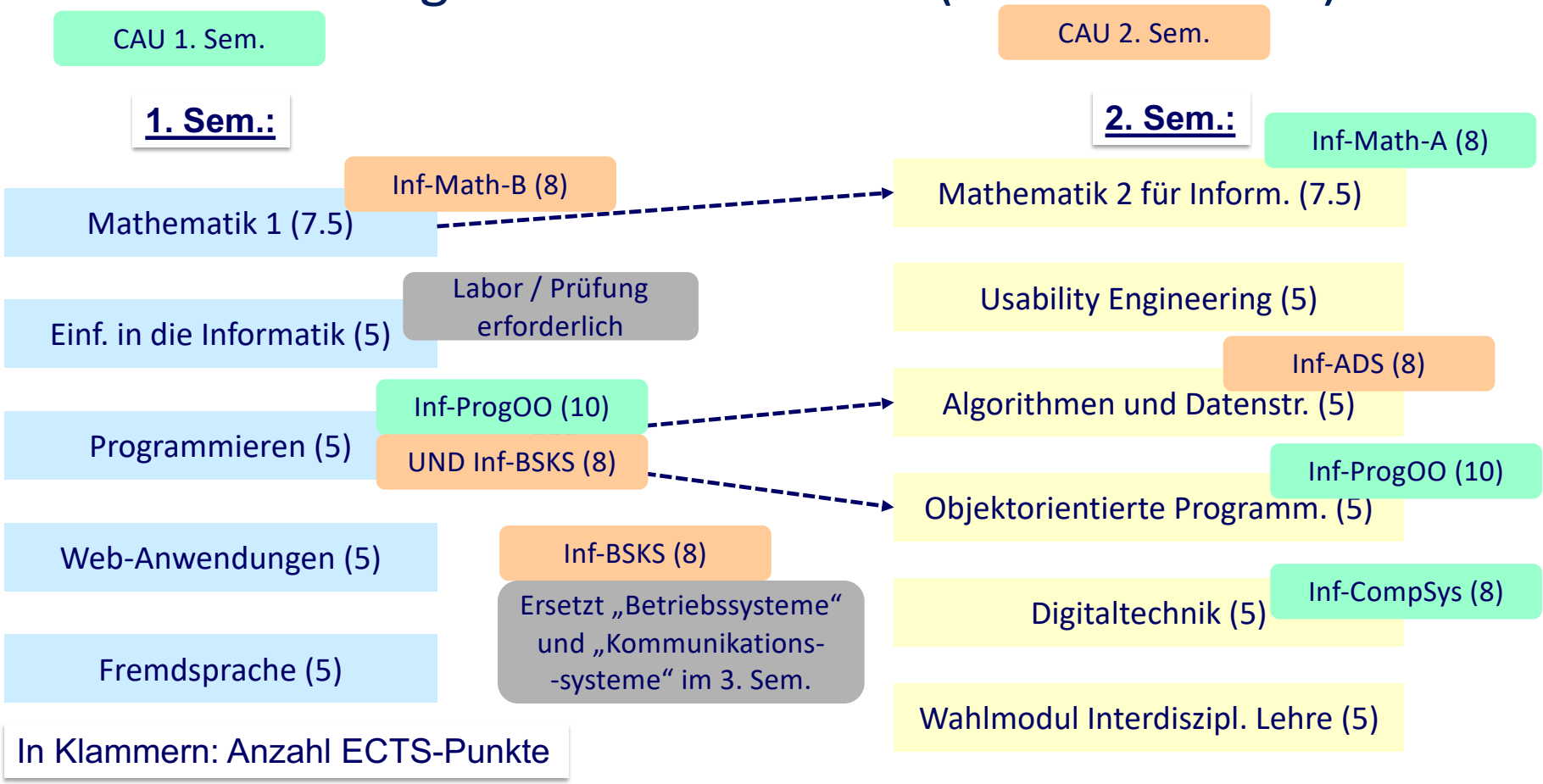
Bachelor Informationstechnologie
FH Kiel

- Erstes Studienjahr -

Studienverlauf



Anrechnung von CAU-Modulen (Unverbindlich!)



Gaststudium

Regulärer Studienanfang: nur zum WS. Alternativen, für WS und SS:

Gasthörer

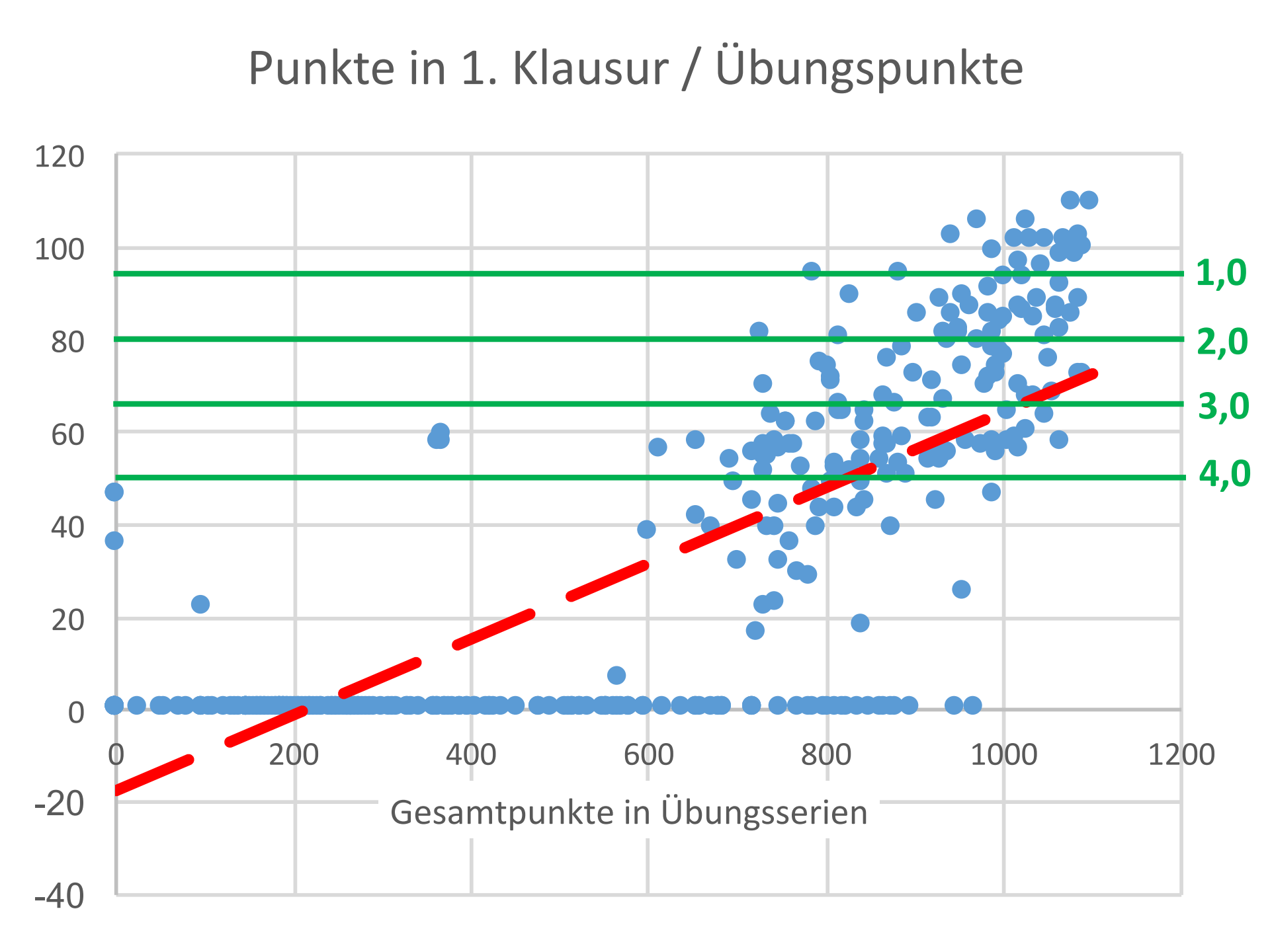
- Kann **keine** Prüfungsleistungen erbringen

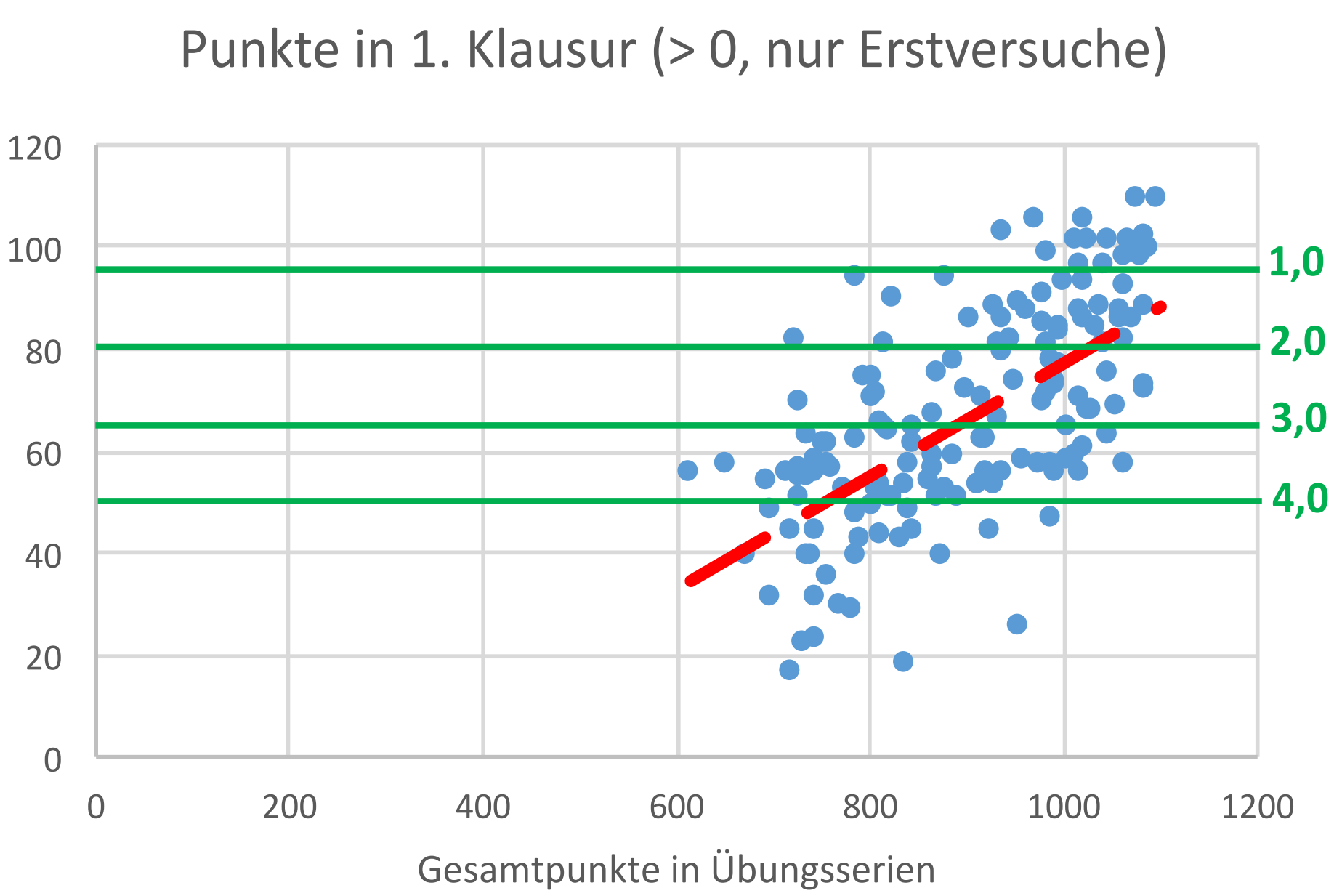
Zweithörer

- Muss an anderer Hochschule (z.B. CAU) eingeschrieben sein
- **Kann** Prüfungsleistungen erbringen
- 10 Euro Gebühr bei Antragstellung

Antragsfristen

- Für Sommersemester: bis 15.1.
- Für Wintersemester: bis 15.7.


Punkte in 1. Klausur / Übungspunkte
120
100
80
60
40
20
0
-20
-40
0
200
400
600
800
1000
1200
Gesamtpunkte in Übungsserien
1,0
2,0
3,0
4,0



Correlation coefficient: 0.78 ("highly significant")

Strategy

- Attend class
- Work through script (“Art and Science of Java”, see previous lecture)
- If there are remaining questions, ask
  1. Fellow students
  2. Tutors, during practical exercises
  3. Lecturer in *Globalübung*

Verblüffender Effekt

Wer Bier trinkt, bricht seltener das Studium ab

Wissenschaftler haben einen Zusammenhang zwischen dem Genuss von Alkohol und einem erfolgreichen Studienabschluss gefunden. Doch die Promille sind gar nicht entscheidend.

DPA

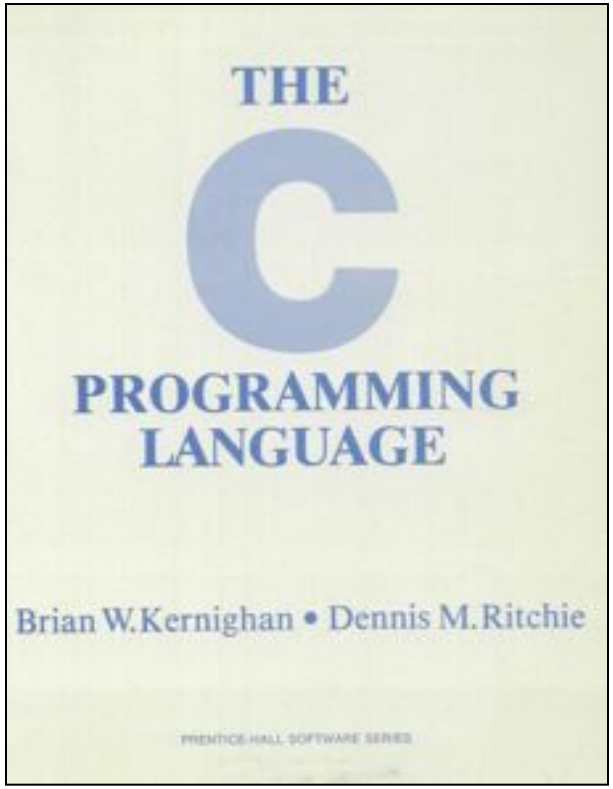
Männer mit Bierkasten

Programming – Lecture 2

Programming by example (Chapter 2)

- Hello world
- Patterns
- Classes, objects
- Graphical programming

1.1 Getting Started



The only way to learn a new programming language is to write programs in it. The first program to write is the same for all languages:

Print the words

```
hello, world
```

This is the big hurdle; to leap over it you have to be able to create the program text somewhere, compile it, load it, run it, and find out where your output went. With these mechanical details mastered, everything else is comparatively easy.

In C, the program to print "`hello, world`" is

```
#include <stdio.h>

main() {
    printf("hello, world");
}
```

Hello World – "Standard Java"

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

Hello World – "Standard Java"

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

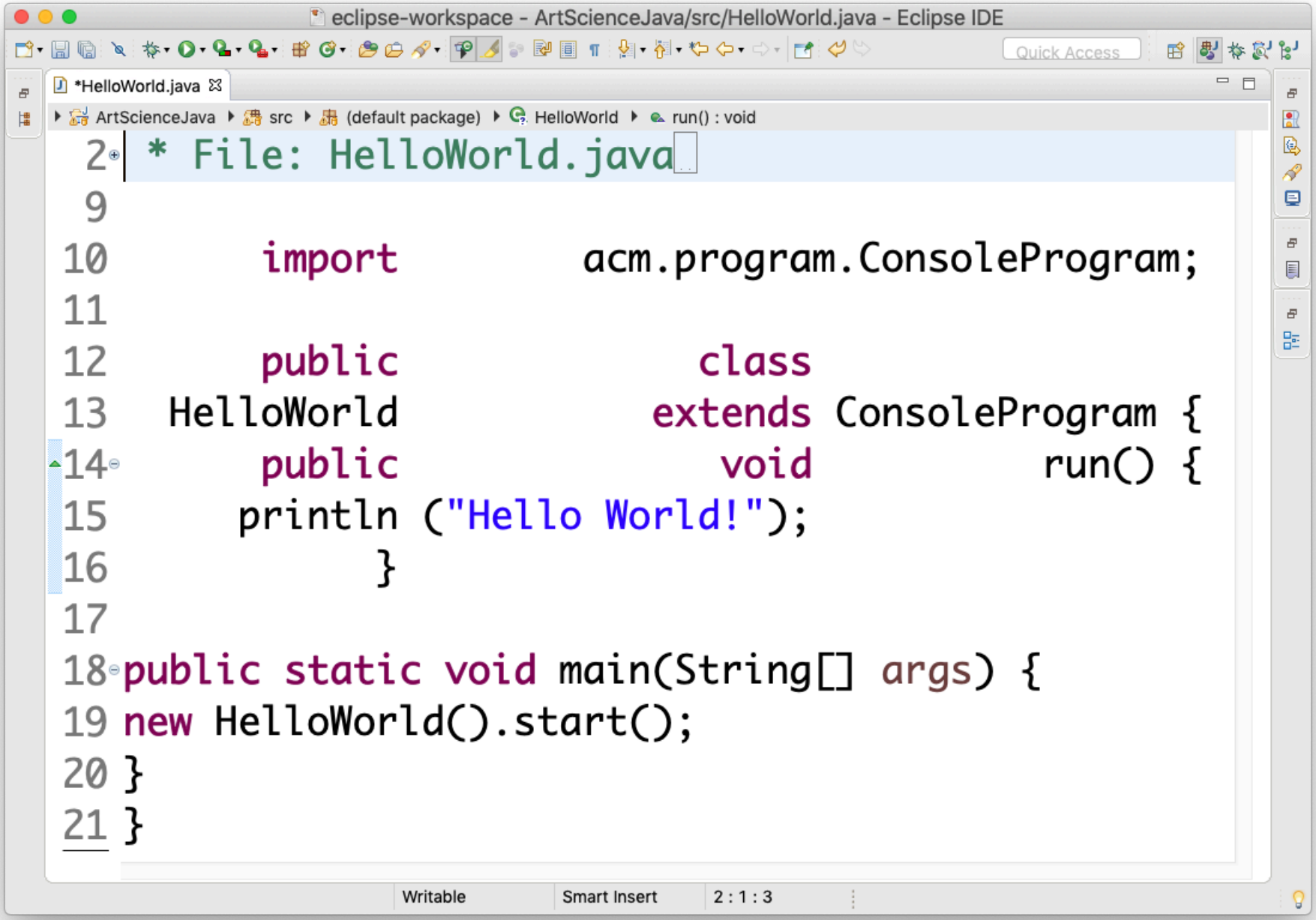
Hello World – "ACM Java"

```
import acm.program.ConsoleProgram;

public class HelloWorld extends ConsoleProgram {
  public void run() {
    println("Hello World!");
  }

  public static void main(String[] args) {
    new HelloWorld().start();
  }
}
```

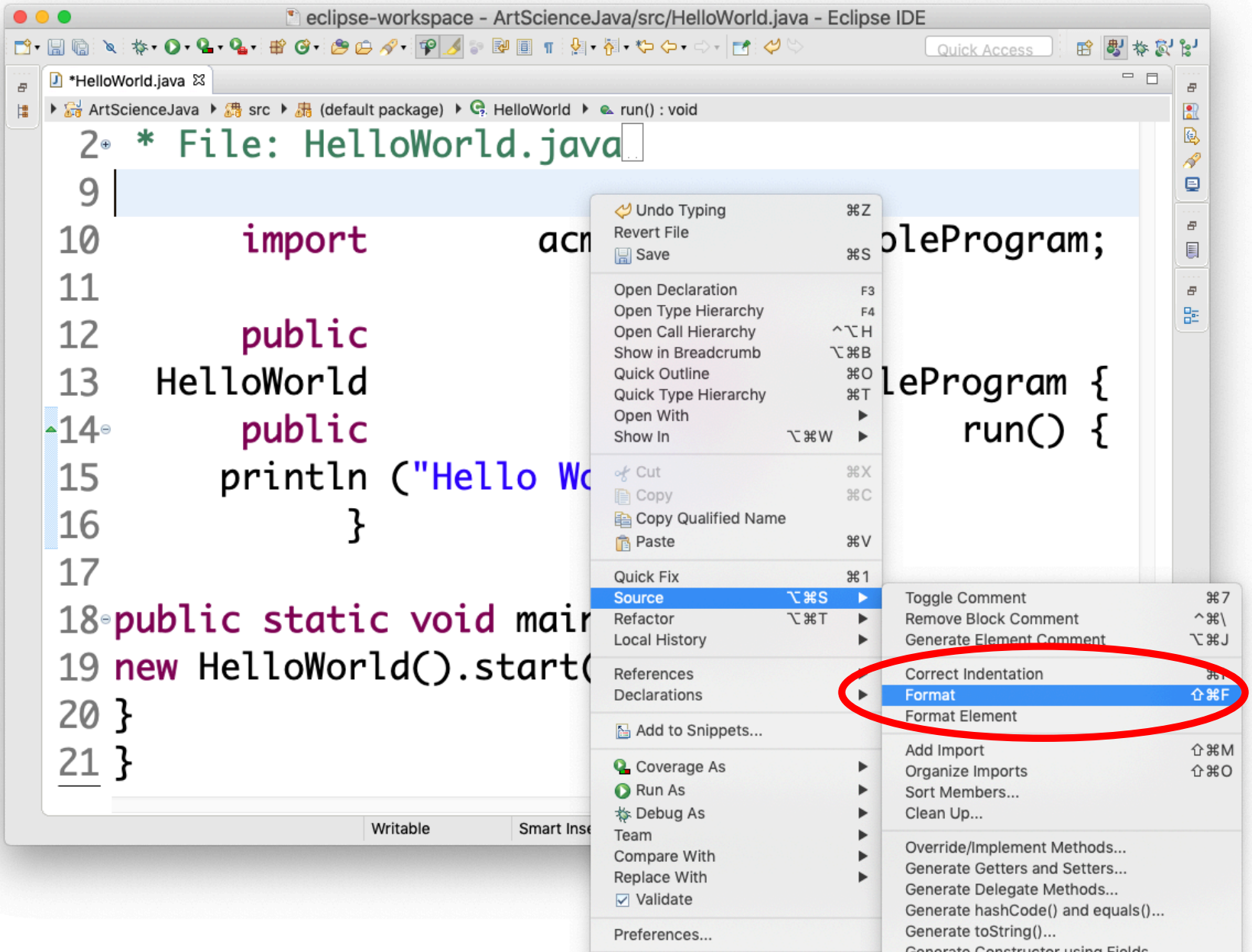
Rationale for ACM Java

- We use ACM Java for didactic purposes
- OO approach to graphics simplifies graphical programming
- I/O model supports automatic testing
- Examples are compatible with text book (except for necessary main() method)
- For further details, see the home page of the ACM Java Task Force: http://cs.stanford.edu/people/eroberts/jtf/

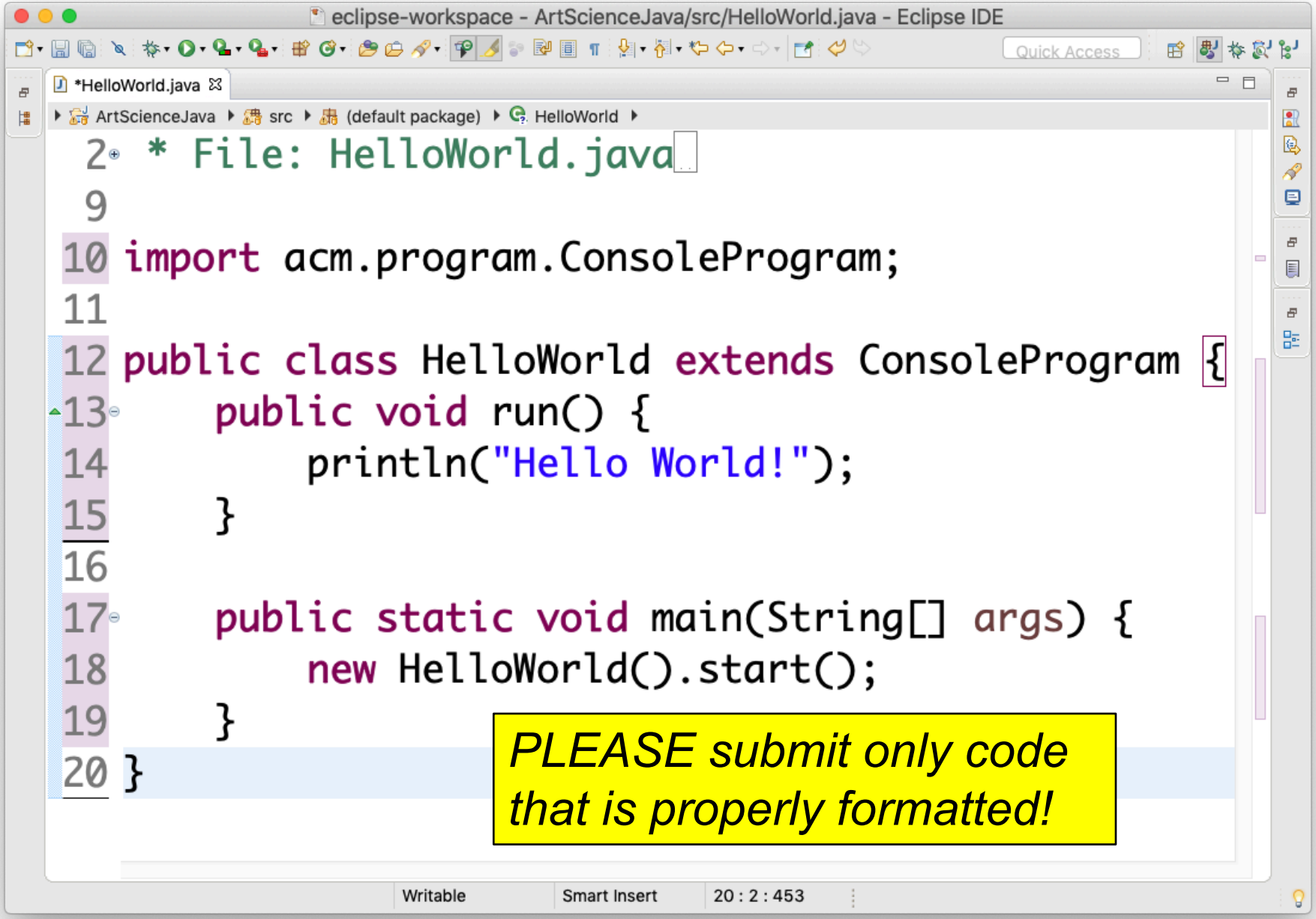
Hello World, on a bad day ...

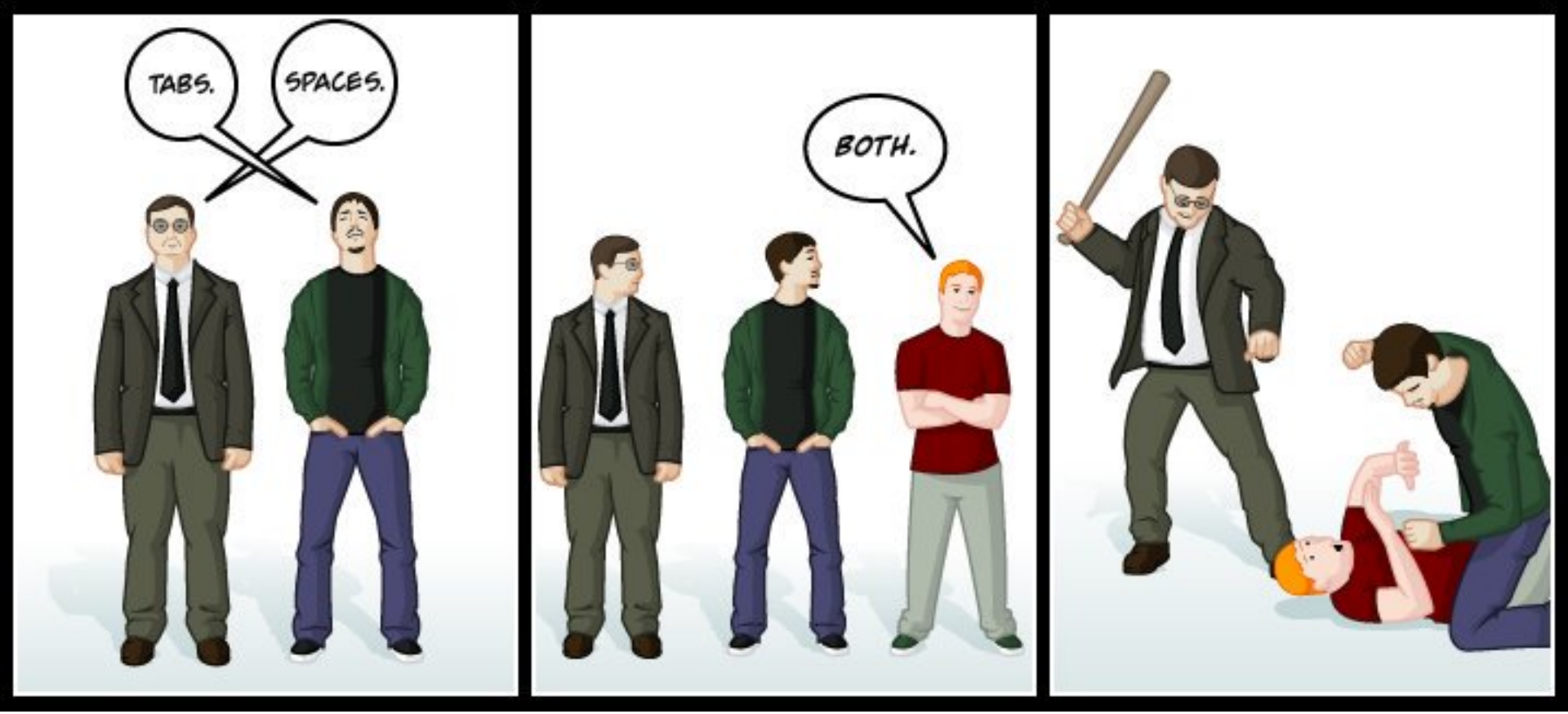


The Magic of Ctrl-Shift-F (Cmd-Shift-F)



Much better ...


eclipse-workspace - ArtScienceJava/src/HelloWorld.java - Eclipse IDE
Quick Access
*HelloWorld.java
ArtScienceJava
src
(default package)
HelloWorld
2 * File: HelloWorld.java
9
10 import acm.program.ConsoleProgram;
11
12 public class HelloWorld extends ConsoleProgram {
13 public void run() {
14 println("Hello World!");
15 }
16
17 public static void main(String[] args) {
18 new HelloWorld().start();
19 }
20 }
PLEASE submit only code that is properly formatted!
Writable
Smart Insert
20 : 2 : 453



www.emacswiki.org, GPL

Beginning of line: indentation (*Einrückung*), use tabs

Within line: alignment (*Ausrichtung*), use spaces

Eclipse automatically does the right thing

See also https://mickaelistria.wordpress.com/2016/09/12/indentation-and-alignment-tabs-and-spaces/

Variables

Variable: placeholder for a piece of data;

"a box to put something into"

- Name (Identifier) – “what the box is called”
- Type – specifies what type of data a variable may hold, “what can be in the box”
- Value – “what’s actually in the box right now”

Variable Declarations

```
int x = 42;
```

A declaration of a variable (with name) x of type int with initial value 42

Syntax template for declaration:

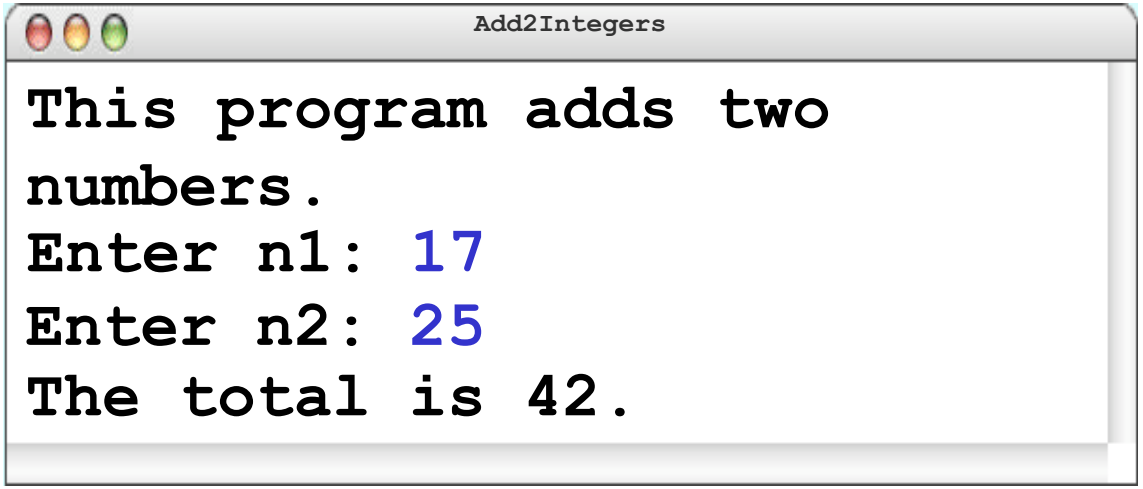
type identifier [**=** *expression*] **;**

where square brackets [] denote optional part, here the optional *initialization* of a variable

Variables

```
class Add2Integers extends ConsoleProgram {
   public void run() {
      println("This program adds two numbers.");
      int n1 = readInt("Enter n1: ");
      int n2 = readInt("Enter n2: ");
      int total = n1 + n2;
      println("The total is " + total + ".");
   }
}
```

n1	n2	total
17	25	42



Java Packages

Typically a collection of useful classes

A.k.a. "library packages" or "libraries"

Syntax: **`import`** *package*`.`*class*`;`

Java Packages

If several classes from same package are needed, *may* abbreviate:
`import` *package* **`.*;`**

Name of package may also include "`.`", but cannot use `*` to include several packages

Advice:

- Use `*` sparingly
- Let the IDE Content Assist add import statements for you (Eclipse: Ctrl-Space)

Coding Advice – Comments

```
// A one-line comment
// Another one-line comment

/*
 * A longer comment.
 * Every program deserves a comment -
 * except for those programs
 * that only live on slides.
 * You may write comments in German,
 * but we encourage you to try English.
 */
```

Writing Javadoc Comments

```
/**
 * Returns the next random integer between 0 and
 * {@code n}-1, inclusive.
 *
 * @param n The number of integers in the range
 * @return A random integer between 0 and {@code n}-1
 */
public int nextInt(int n)
```

`public int nextInt(int n)`

Returns the next random integer between 0 and `n`-1, inclusive.

Parameter: `n` The number of integers in the range
Returns: A random integer between 0 and `n`-1

Note: Eclipse can automatically generate Javadoc templates ("Generate Element Comment").
These must of course still be filled with content!

Programming Idioms / Patterns

- Holistic (top-down) vs. reductionistic (bottom-up)
- Holistic pattern “read integer from user”:
  `int` *variable* **`= readInt("`***prompt***`");`**

Core Object-Oriented (*OO*) Terminology

- *Class*: template for individual objects
- *Object*: an instance of a particular class
- *Method*: named sequence of program steps (statements)
- *Constructor*: special method that creates new objects

Syntax: **`new`** *constructor* (*arguments*) **`;`**

oop - Can I say a Constructor i
Sicher | https://stackoverflow.com/questions/3646461/can-i-say-a-c...
Apps | Erste Schritte | Maps | WEB.DE Services | Mopped | Importiert | Andere Lesezeichen

Can I say a Constructor is a Method? [closed]

Get personalized job matches now
stackoverflow JOBS
Get started

I wonder if can I say that a **constructor** is a special case of a **method**?

10

oop methods constructor terminology

share improve this question

2

edited Oct 7 '10 at 16:04
skaffman
309k • 74 • 677 • 689

asked Sep 5 '10 at 14:53
Tom Brito
9,252 • 43 • 129 • 220

closed as primarily opinion-based by bmargulies, Suever, KittMedia, Will, Ryan Bemrose Jul 9 '16 at 19:34

Many good questions generate some degree of opinion based on expert experience, but answers to this question will tend to be almost entirely based on opinions, rather than facts, references, or specific expertise.

If this question can be reworded to fit the rules in the help center, please edit the question.

8 Answers

active oldest votes

13

You can say anything. Whether anyone will disagree with you depends on the context. Some language communities and standards define things that way.

More elaborately, it depends on what you mean by a 'method.' In C++, for example, one way to analyze the creation process is to say that it consists of a call to an operator new (perhaps just placement) followed by a call to a constructor *method*. From an implementation standpoint, a constructor looks, walks, and quacks like a method. In some compilers, you can even invoke one explicitly.

From a more theoretical viewpoint, someone might claim that constructors are some distinctive species. However, there is no single, true, privileged conceptual model of methods, constructors, or purple unicorns.

Gosh this is all subjective.

share improve this answer

edited Sep 6 '10 at 11:56

answered Sep 5 '10 at 14:58

bmargulies

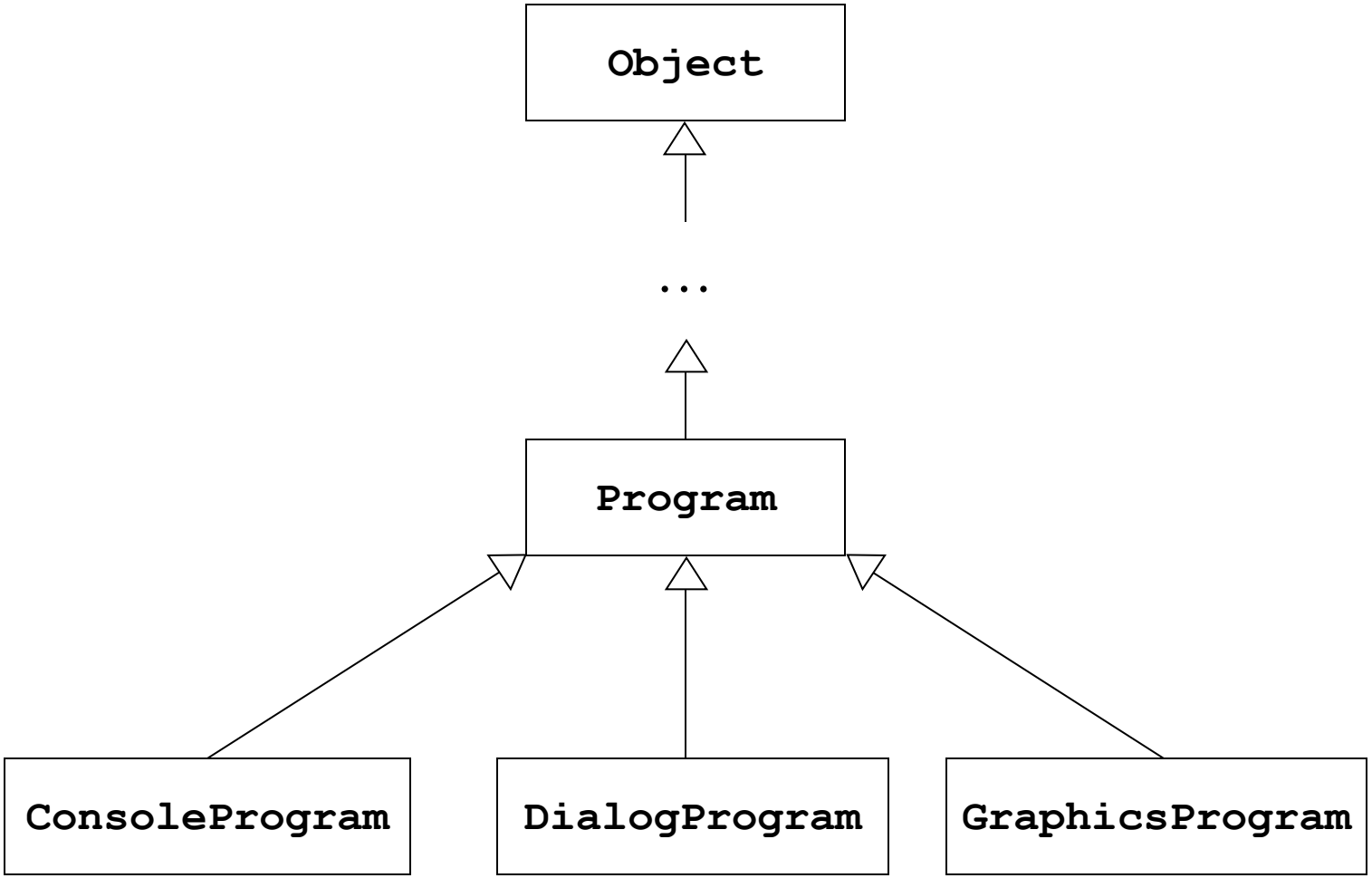
71.4k • 29 • 131 • 247

Core OO Terminology

- *Object variable*: a variable whose type is a class, and which holds a *reference* to an object that is an instance of that class
- **Note:** (object) variables have names, but objects don't
- Classes form hierarchies, a *subclass* *extends* (*inherits* from) a *superclass*

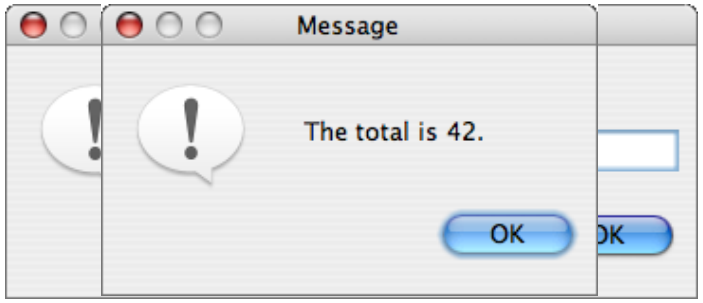
Syntax: **`class`** *subclass* **`extends`** *superclass*

The `acm.program` Hierarchy

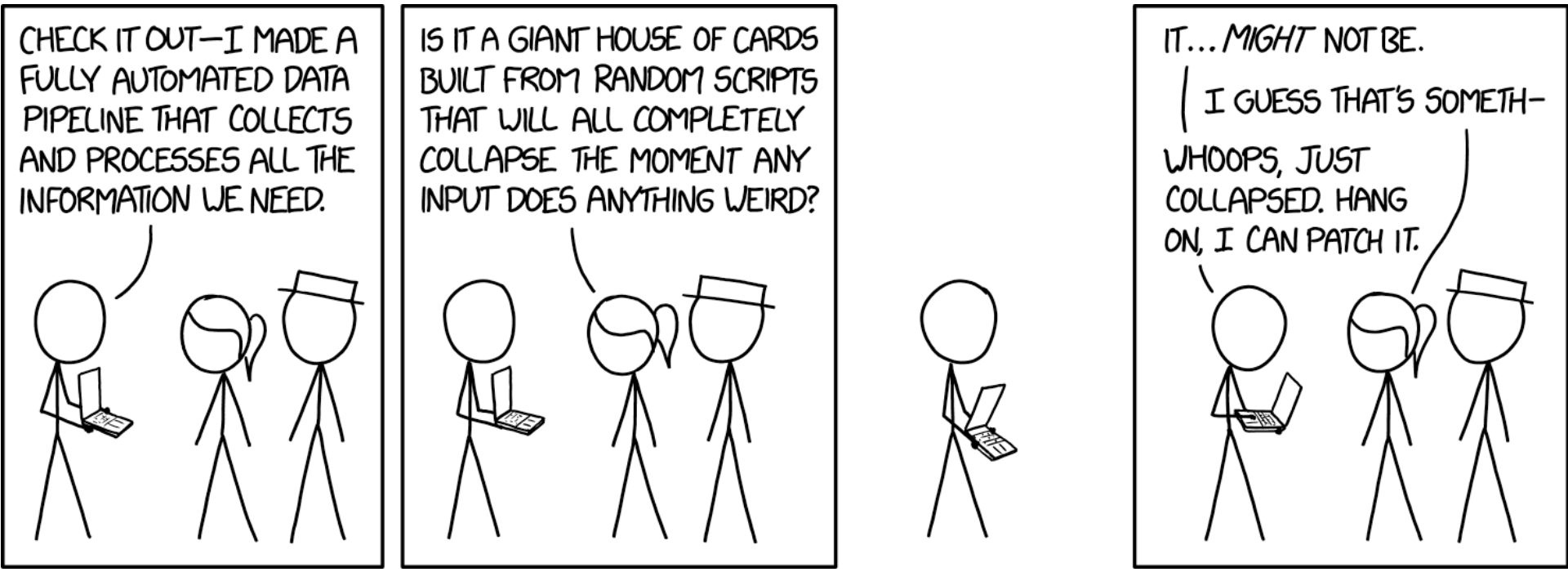


The `DialogProgram` Class

```
public class Add2Integers extends DialogProgram {
   public void run() {
      println("This program adds two numbers.");
      int n1 = readInt("Enter n1: ");
      int n2 = readInt("Enter n2: ");
      int total = n1 + n2;
      println("The total is " + total + ".");
   }
}
```



Please visit http://pingo.upb.de/643250



https://xkcd.com/2054

Sending Messages to Objects

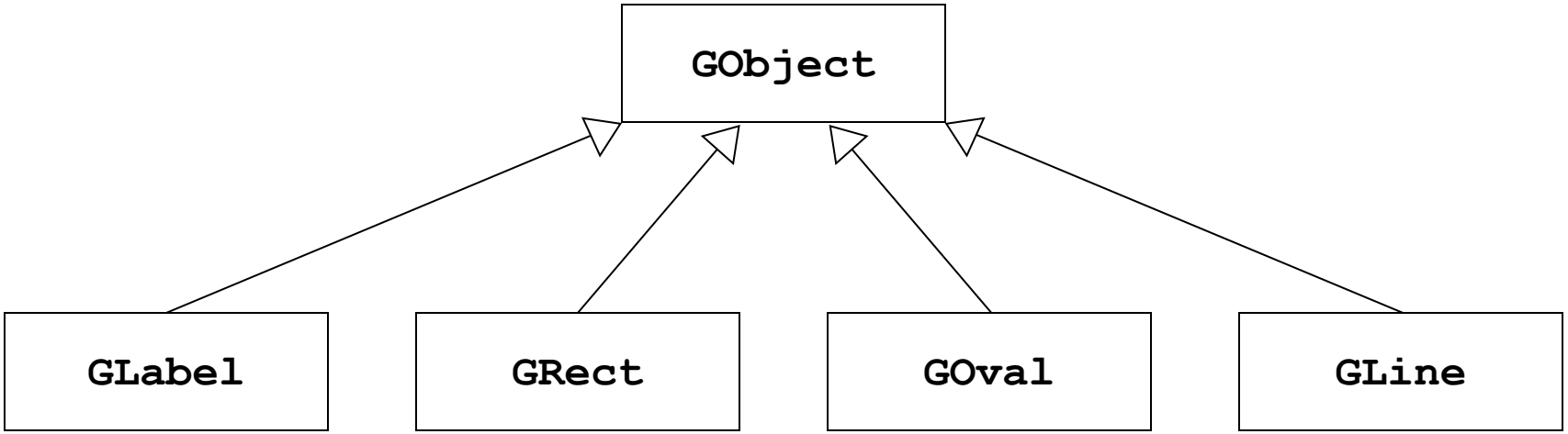
receiver`.`*method*`(`*arguments*`);`

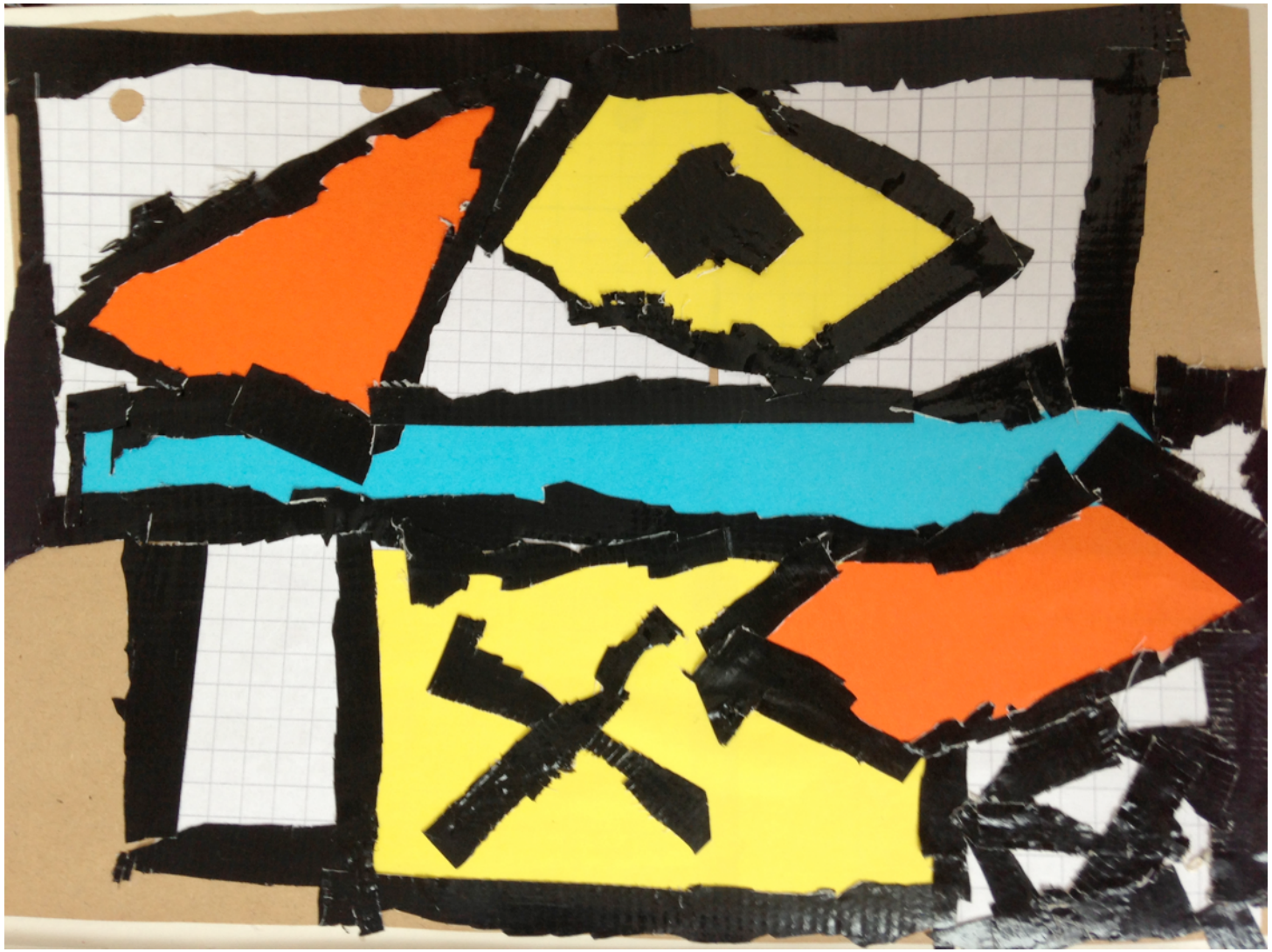
```
public class HelloProgram extends GraphicsProgram {
   public void run() {
      GLabel label = new GLabel("hello, world", 100, 75);
      label.setFont("SansSerif-36");
      label.setColor(Color.RED);
      add(label);
   }
}
```

label

hello, world

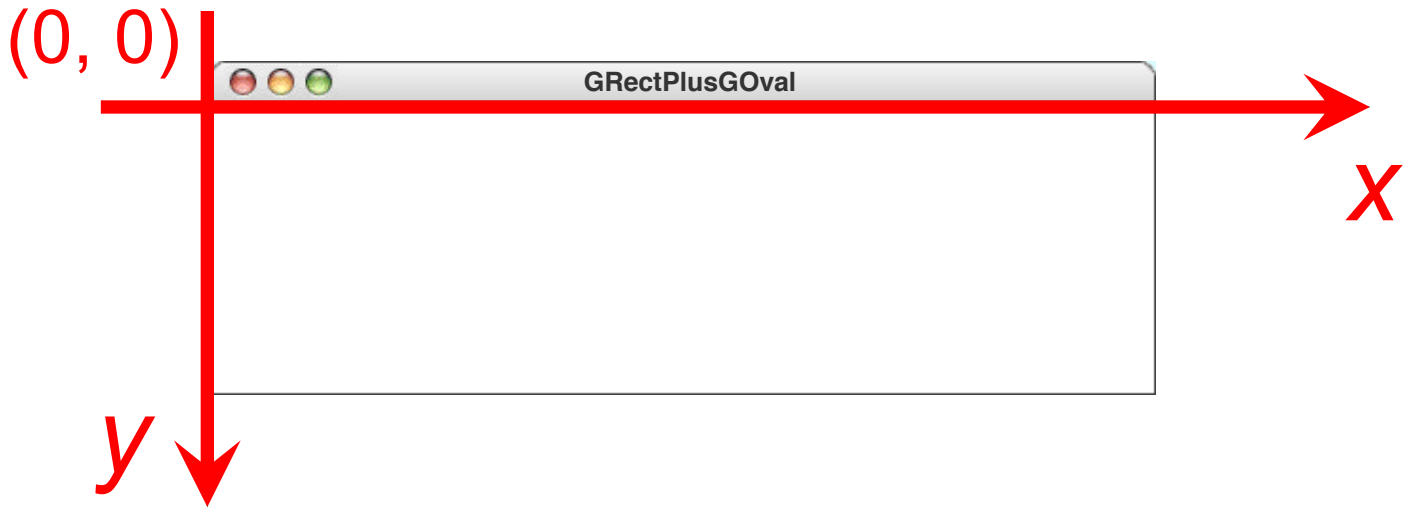
The `GObject` Hierarchy





Wikipedia / marcel4995 / CC

The Java Coordinate System



- *x*-coordinate increases rightwards
- *y*-coordinate increases **downwards**
- Origin (0, 0) in **upper**-left corner of canvas

Operations on the `GObject` Class

object`.setColor(`*color*`)` Sets color of object to specified color constant.
object`.setLocation(`*x*`, `*y*`)` Changes location of object to point (*x*, *y*).
object`.move(`*dx*`, `*dy*`)` Moves object on screen by adding *dx* and *dy* to its current coordinates.

Standard color names, defined in `java.awt` package:

```
Color.BLACK       Color.RED       Color.BLUE
Color.DARK_GRAY   Color.YELLOW    Color.MAGENTA
Color.GRAY        Color.GREEN     Color.ORANGE
Color.LIGHT_GRAY  Color.CYAN      Color.PINK
Color.WHITE
```

Operations on the **`GLabel`** Class

Constructor

`new GLabel(`*text*`, `*x*`, `*y*`)`**
Creates a label containing the specified text that begins at the point (*x*, *y*).

Methods specific to **`GLabel`** class

label`.`**`setFont(`***font*`)`**
Sets the font used to display the label as specified by the font string.

Font typically specified as string

"*family*`-`*style*`-`*size*"

where

family denotes font family
style is either **`PLAIN`**, **`BOLD`**, **`ITALIC`**, or **`BOLDITALIC`**
size is integer indicating point size

Drawing Geometrical Objects

Constructors

`new GRect(` *x*`,` *y*`,` *width*`,` *height*`)`
Creates a rectangle whose upper left corner is at (x, y) of the specified size.

`new GOval(` *x*`,` *y*`,` *width*`,` *height*`)`
Creates an oval that fits inside the rectangle with the same dimensions.

`new GLine(` x_0`,` y_0`,` x_1`,` y_1`)`
Creates a line extending from (x_0, y_0) to (x_1, y_1).

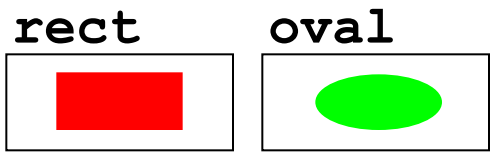
Methods shared by the **`GRect`** and **`GOval`** classes

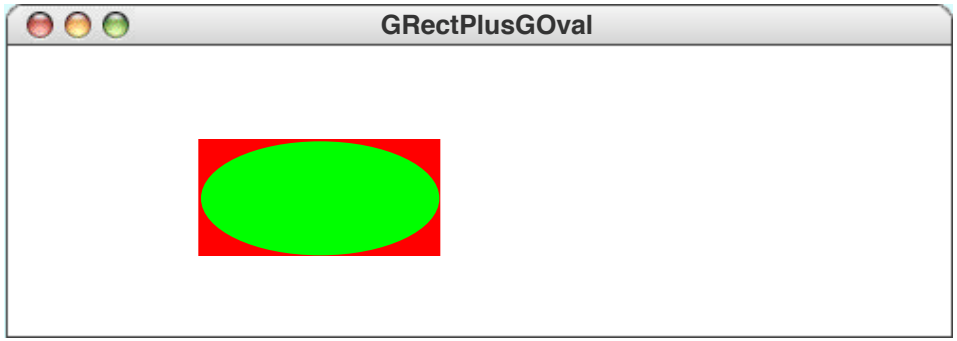
object`.`**`setFilled(`***fill*`)`
If *fill* is **`true`**, fills in the interior of the object; if **`false`**, shows only the outline.

object`.`**`setFillColor(`***color*`)`
Sets the color used to fill the interior, which can be different from the border.

The GRectPlusGOval Program

```
public class GRectPlusGOval extends GraphicsProgram {
   public void run() {
      GRect rect = new GRect(100, 50, 125, 60);
      rect.setFilled(true);
      rect.setColor(Color.RED);
      add(rect);
      GOval oval = new GOval(100, 50, 125, 60);
      oval.setFilled(true);
      oval.setFillColor(Color.GREEN);
      add(oval);
   }
}
```





skip simulation

Summary

- Got started with “hello world”
- **Coding advice:** properly format your code
- *Holistic* vs. *reductionistic*
- *Classes* form *hierarchy*,
  subclass inherits from *superclass*
- *Objects* are *instances* of a class
- Java coordinates
- GObjects