



Programming Basics – WiSe21/22

Object orientation

Prof. Dr. Silke Lechner-Greite

Table of contents – planned topics

1. Introduction
2. Fundamental language concepts
3. Control structures
4. Methods
5. Arrays
6. **Object orientation**
7. Classes
8. Packages
9. Characters and Strings
10. Unit Testing
11. Exceptions
12. I/O

Chapter 6: Object orientation

6.1 Comparison of selected programming paradigms

6.2 Core idea of object orientation

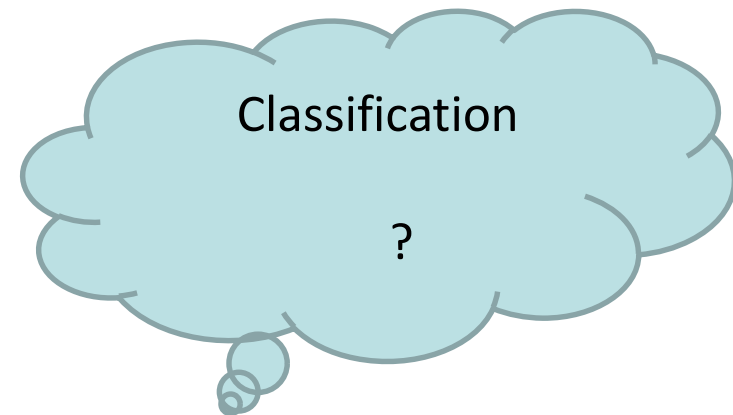
6.3 Definition of terms and characteristic features of objects

Problem description

- Around 680 programming languages are listed in Wikipedia in the list of programming languages (as of September 2021)

https://de.wikipedia.org/wiki/Liste_von_Programmiersprachen

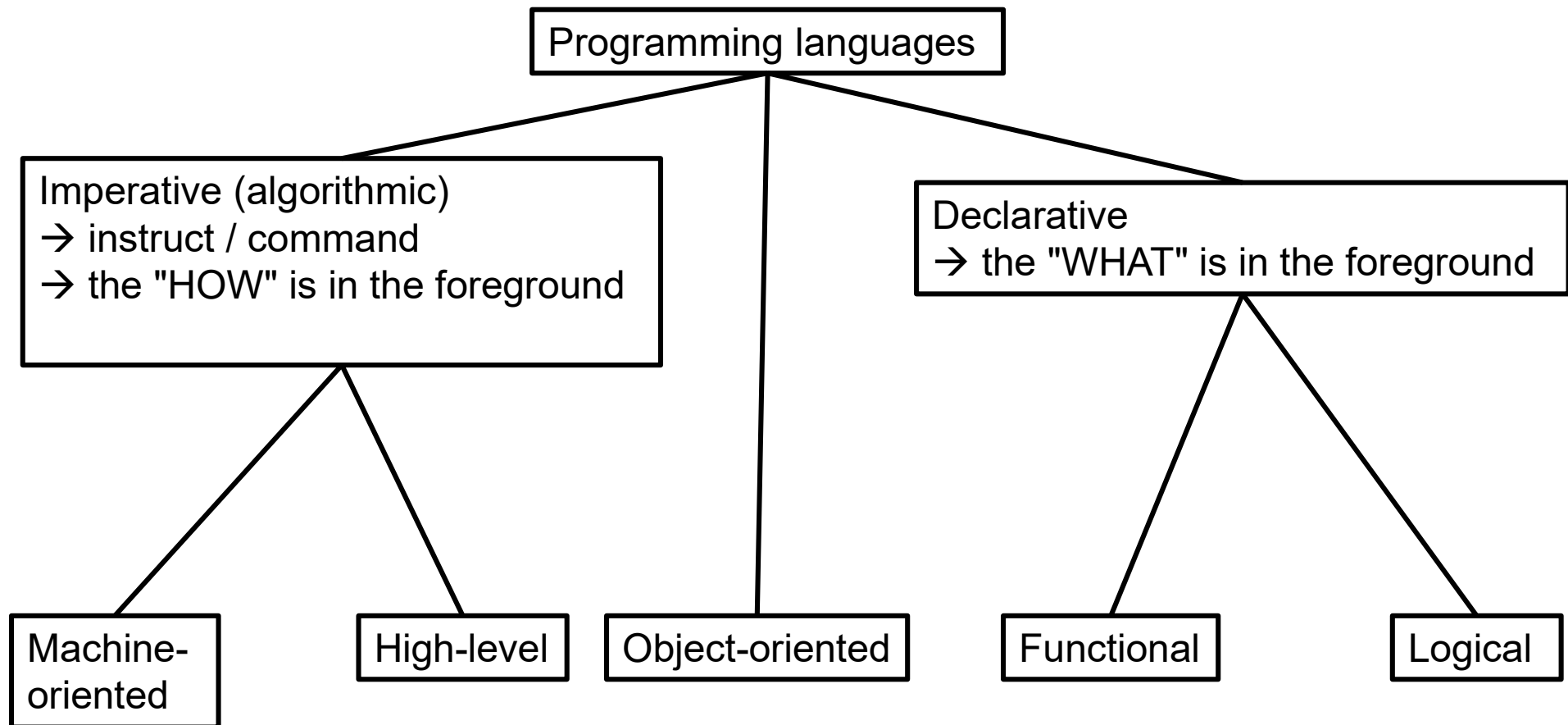
- This raises many questions:
 - ⊞ Which programming language is the right one?
 - ⊞ Which is widely used?
 - ⊞ Which is right for my problem?
 - ⊞ Which is modern?
 - ⊞ How many of them should I know?



Classification by

- Field of application
 - History
 - Programming language generations
 - Programming paradigms
 - Extent of use
 - Core concepts
 - Type of representation
-
- → Here too, there are different approaches

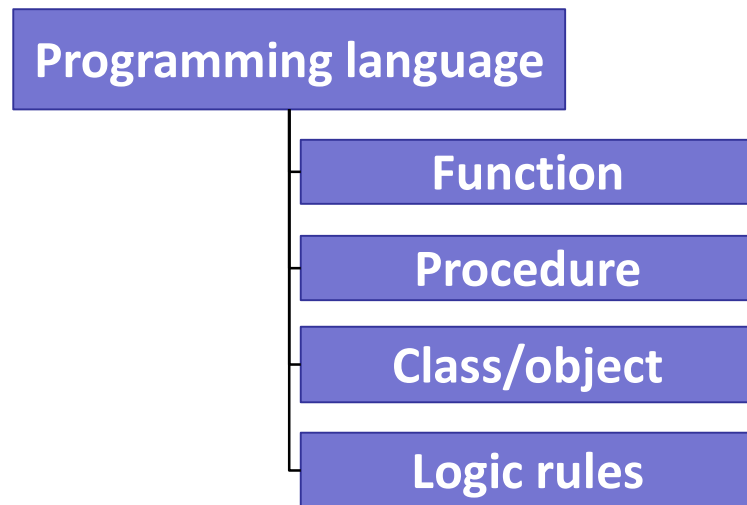
Classification according to programming paradigms



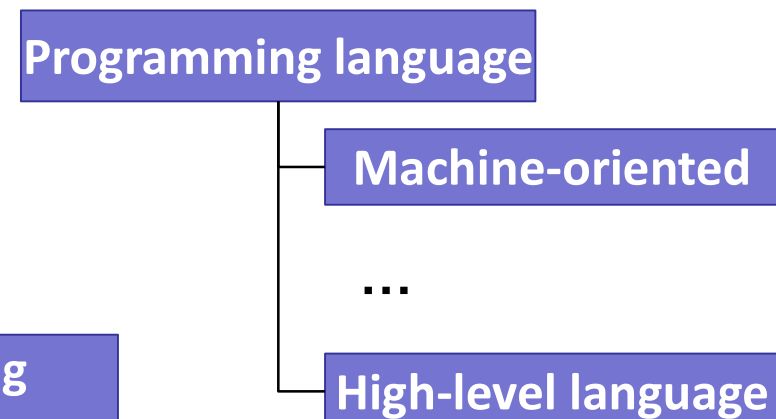
Be careful: there are also mixed forms, e.g. Java – object-oriented & imperative

Example classification approaches for programming languages

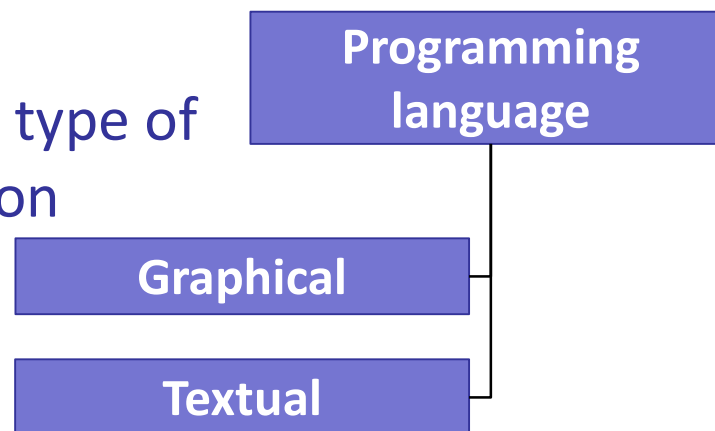
According to core concepts



According to degree of abstraction



According to type of representation



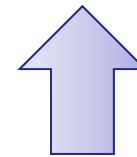
Development of programming paradigms

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

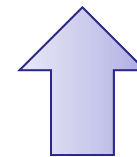
```
Identification Division
Program-ID. HALLOPGM.
Procedure Division.
    Display "Hallo Wiki"
    STOP RUN.
```

```
mov ax, DATA ;
mov ds, ax ;
mov dx, offset
mov ah, 09h ;
int 21h ; - der
```

Object-oriented



Procedural



Machine-oriented

What is a programme?

- A (computer) programme is ...
 - ⊞ a collection of data and control instructions,
 - ⊞ which can be run from a device with a processor.

- So what does that mean?
 - ⊞ A programme manages some kind of data.
 - ⊞ It does something (executable), mostly with/to this data.

- Conclusion: it's all about
 - ⊞ Data
 - ⊞ Behaviour



Current situation (or similar)

- ✦ Within the scope of the "Business Information Systems" degree programme, there is a compulsory course in "Programming Basics". In the current semester, this is being given by Prof. Lechner-Greite. The lectures take place in R 0.02, which has 100 seats.
- ✦ The students Lena Müller (matriculation no. 123456) and Peter Meier attend this course. Lena achieved a grade of 1.0 for a subject during previous studies.
- ✦ At the end of the semester, the lecturer prepares an examination. The students sit the examination, which is then marked by the lecturer.



Split into **data** and **behaviour**

- Within the scope of the "Business Information Systems" degree programme, there is a compulsory course in "Programming Basics". In the current semester, this is **being given** by Prof. Lechner-Greite. The lectures take place in R 0.02, which has 100 seats.
- The students Lena Müller (matriculation no. 123456) and Peter Meier **attend this course**. Lena achieved a grade of 1.0 for a subject during previous studies.
- At the end of the semester, the lecturer **prepares an examination**. The students **sit the examination**, which is then **marked** by the lecturer.

Data

Lechner-Greite	123456
R 0.02	Business Information Systems
Lena Müller	Programming Basics
	1.0
Peter Meier	100 seats

Behaviour

Giving lectures	Marking exams
	Attending lectures
Solving tasks	Asking questions
	Preparing exams
Taking exams	Answering questions

Machine-oriented approach

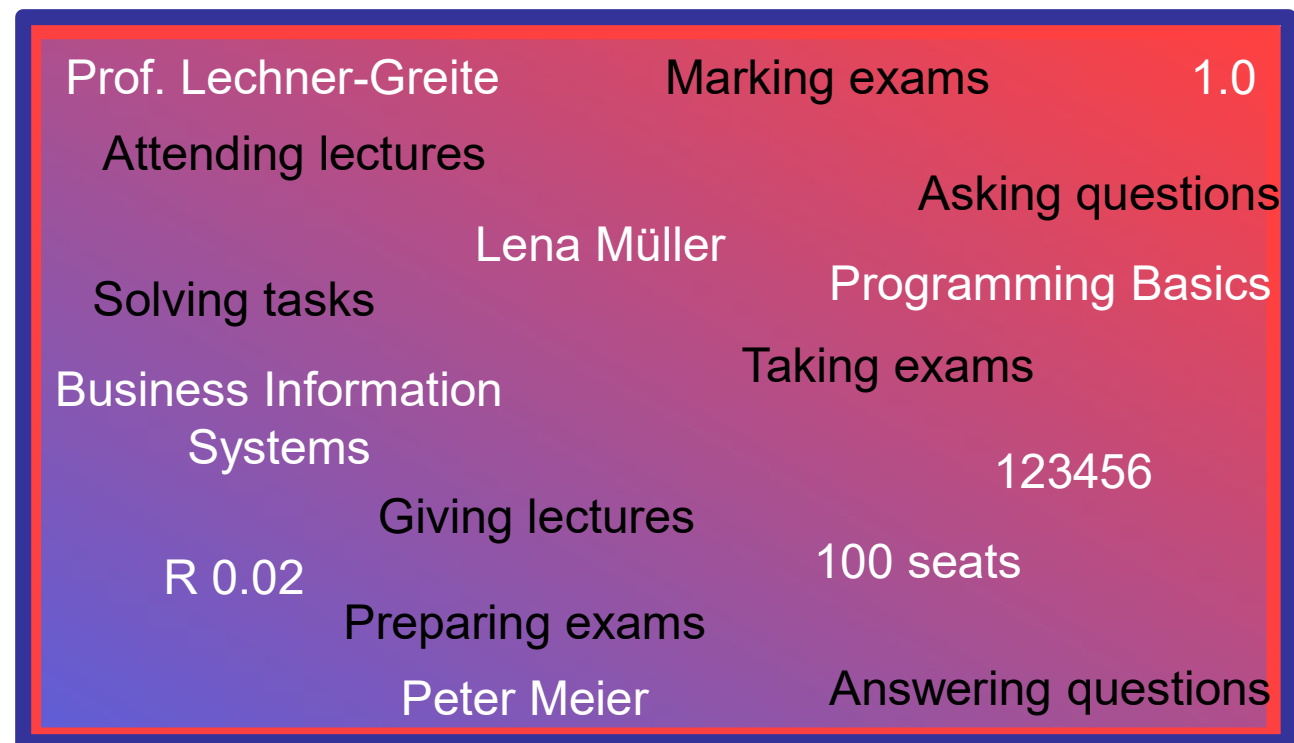
➤ Monolithic

- ⊞ Data space and behavioural space extend across the entire system
- ⊞ Any function can change any data element (!)

Data

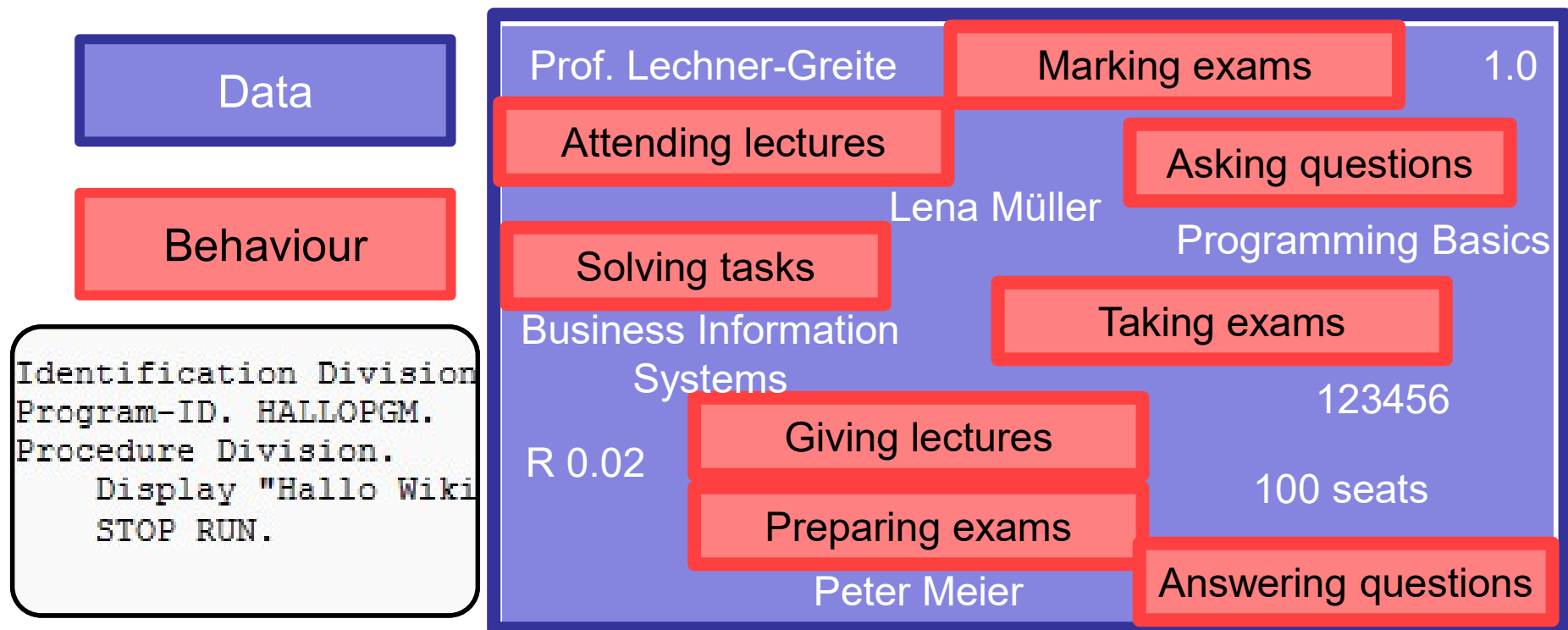
Behaviour

```
mov ax, DATA ;
mov ds, ax ;
mov dx, offset
mov ah, 09h ;
int 21h ; - der
```



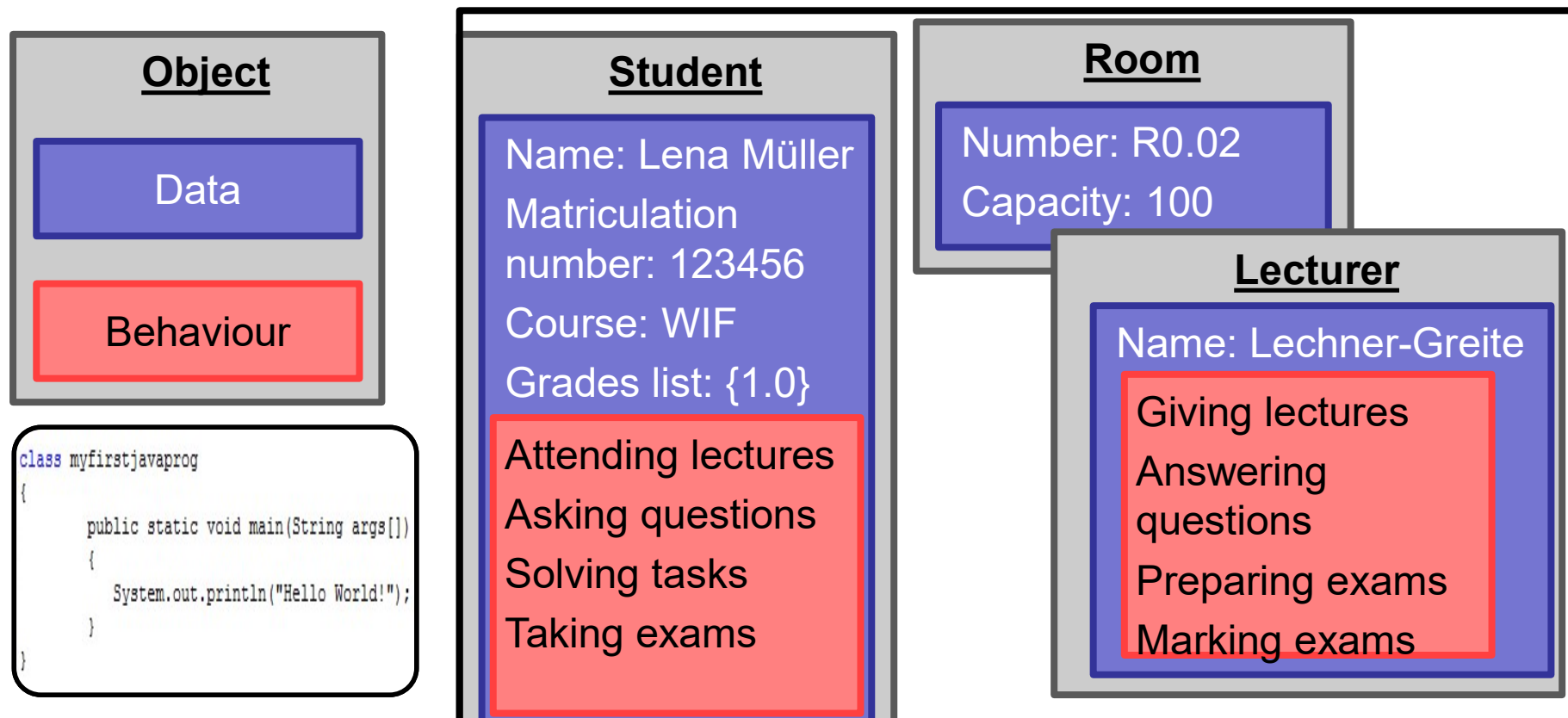
Procedural approach

- Structure monolithic, behavioural space modular
 - ⊞ Data space extends across the entire system
 - ⊞ Behavioural space structured, one entry point, each function/method



Object-oriented approach

- Modular structure of data AND behaviour
 - ⊞ Data space and behavioural space modular
 - ⊞ Module forms a unit of data and behaviour



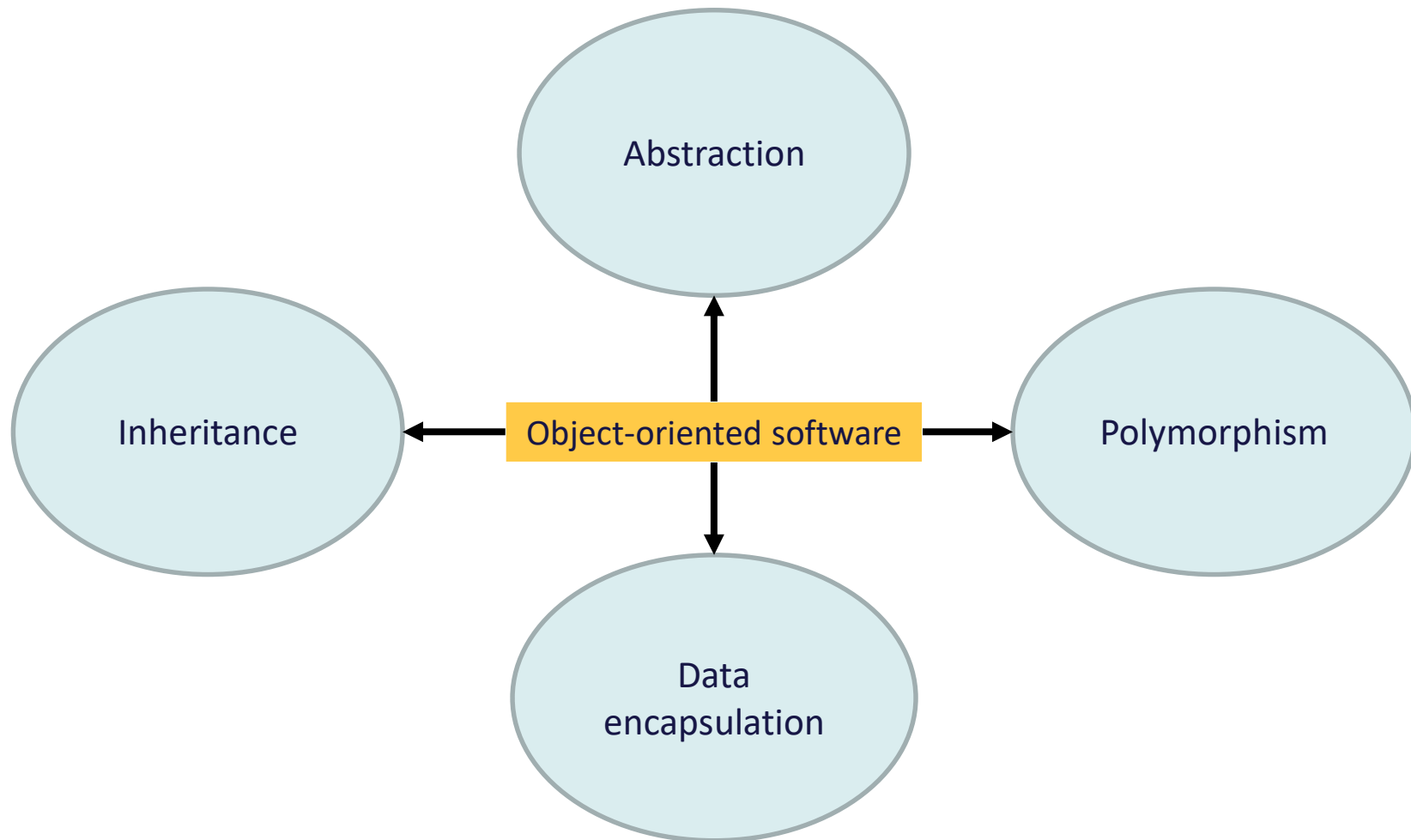
Chapter 6: Object orientation

6.1 Comparison of selected programming paradigms

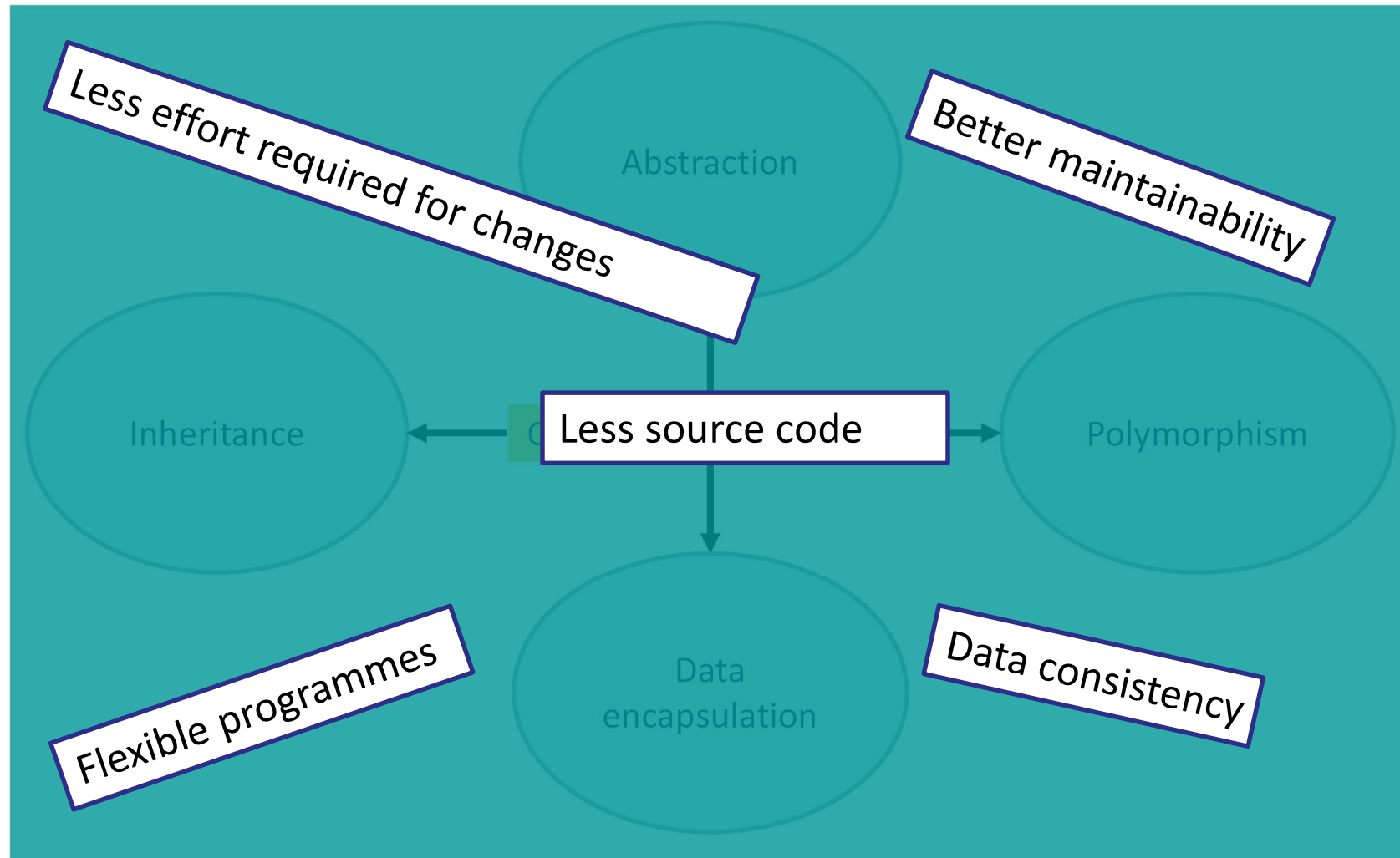
6.2 Core idea of object orientation

6.3 Definition of terms and characteristic features of objects

4 elements of object-oriented software



Benefits



Core idea of object orientation

➤ Object orientation

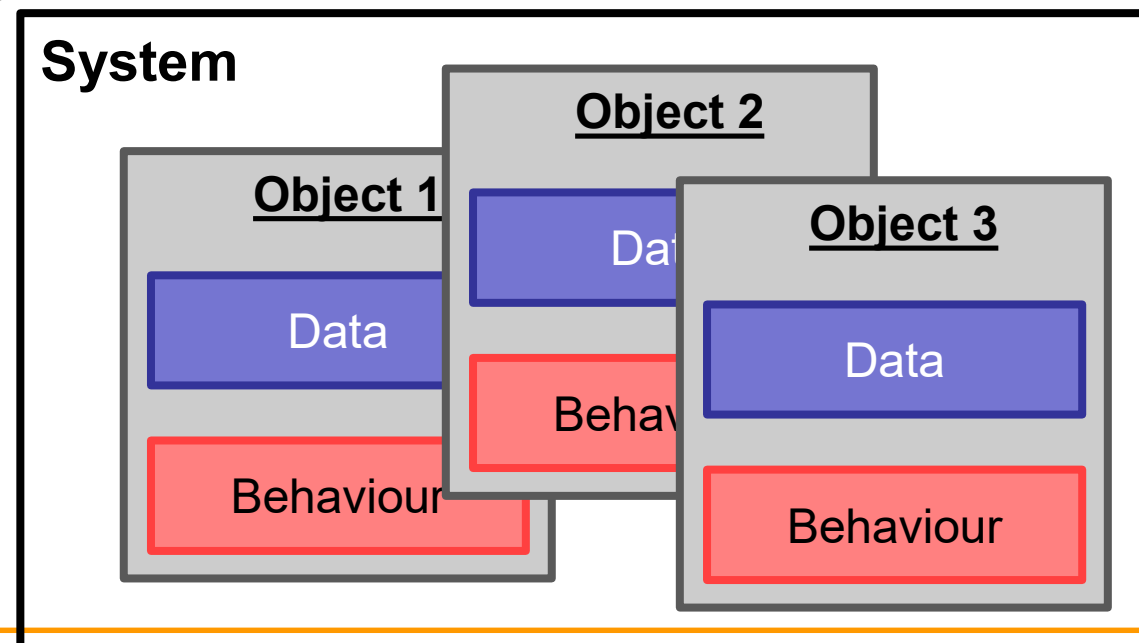
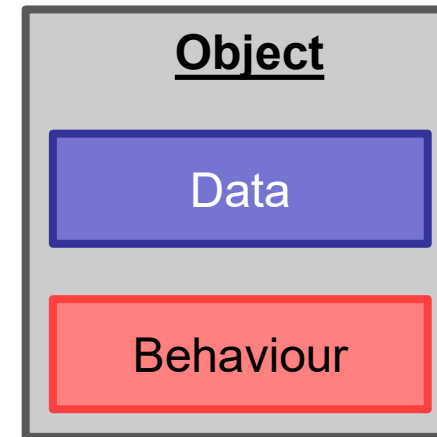
⊞ Data encapsulation:

- ⊞ Object / class encapsulates data and behaviour into a unit
- ⊞ Only the **own** behaviour accesses data of the object (!!!)

⊞ Abstraction of details of the real world

➤ System

- ⊞ modular design
- ⊞ structured by objects / classes
- ⊞ Classes have relationships



Chapter 6: Object orientation

6.1 Comparison of selected programming paradigms

6.2 Core idea of object orientation

6.3 Definition of terms and characteristic features of objects

What is actually an object?

- **Object = representation of a specific unit of the real world**
 - ✦ Independent, self-contained
 - ✦ Clearly delimitable from other objects
- Examples that are easy to grasp:



James' car



Luthers' book




Sissi's carriage

- More abstract examples:
 - ✦ Marriage between Tina and Peter
 - ✦ Course "Programming Basics"

Characteristics of an object

➤ Specific object

- ✦ Has various properties that determine its current **state**




- Tina
- 9 years old
- born on 23.04.2012




- M-JD 1955
- Built in 1973
- 150 297 km

- ✦ Can perform dynamic **behaviour**

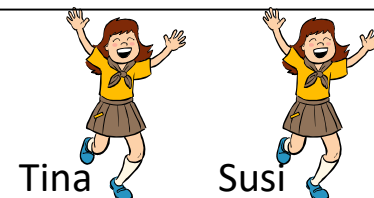


- answers
- jumps
- gets older



- driving
- breaking
- locking doors

- ✦ Is uniquely identifiable (**identity**)



Characteristics of an object – schema



Concept

Identity

Identifier refers uniquely to the object

Status == Data

Relevant values that describe the object (properties)

Behaviour

Operations that the object can perform

Object

Abstraction

Real world



State of an object

➤ State

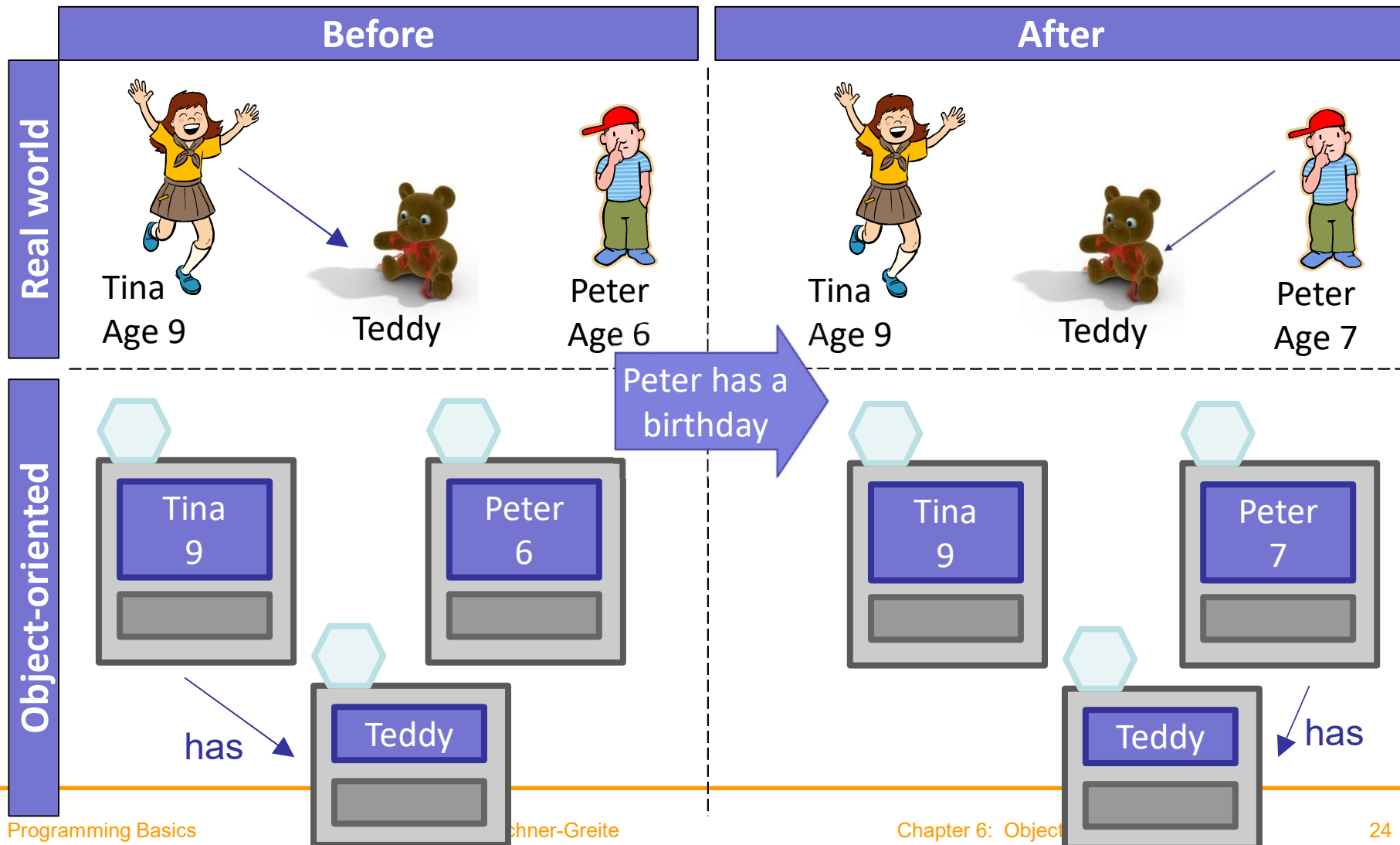
- ⊞ Relevant properties of the object (= **attributes**)
- ⊞ Relationships of the object to other objects
(relationships are later also represented by attributes.)
- ⊞ Attributes form the possible **data space** of the object
- ⊞ The state is determined by the current value assignment of the attributes

➤ Example:



State of an object – example

- State can be changed over the lifetime of the object



Representation of objects

- So far
 - ⊞ Informal sketching of objects
 - ⊞ Images are not uniform => difficult to understand
- Wanted: **standardised graphical representation for objects**
- **UML = Unified Modeling Language**
 - ⊞ Notation building blocks for object-oriented modelling
 - ⊞ Modelling of structure
 - ⊞ Modelling of behaviour
 - ⊞ Standardisation by the Object Management Group
 - ⊞ <http://www.uml.org/>



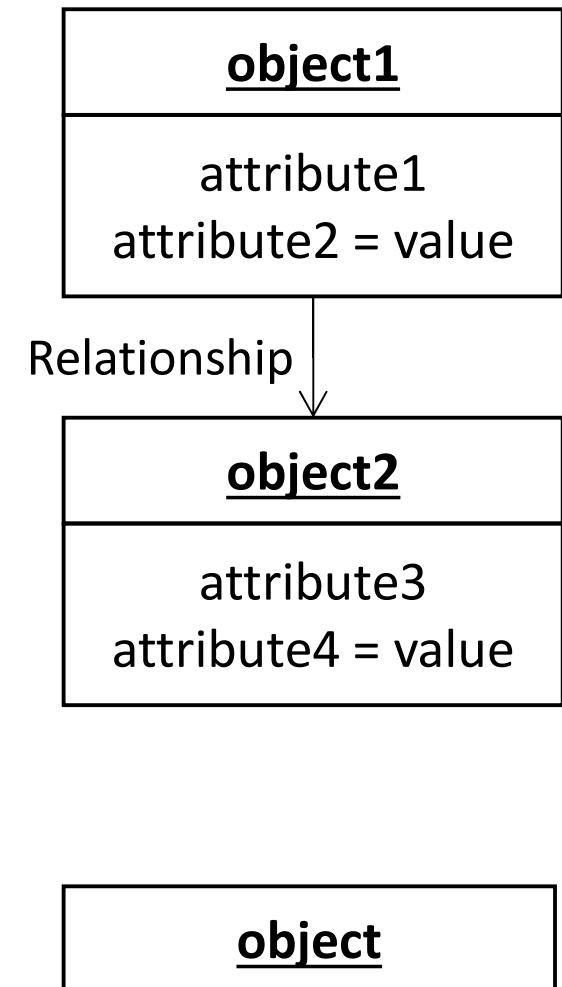
Object diagram

➤ Representation of objects in UML

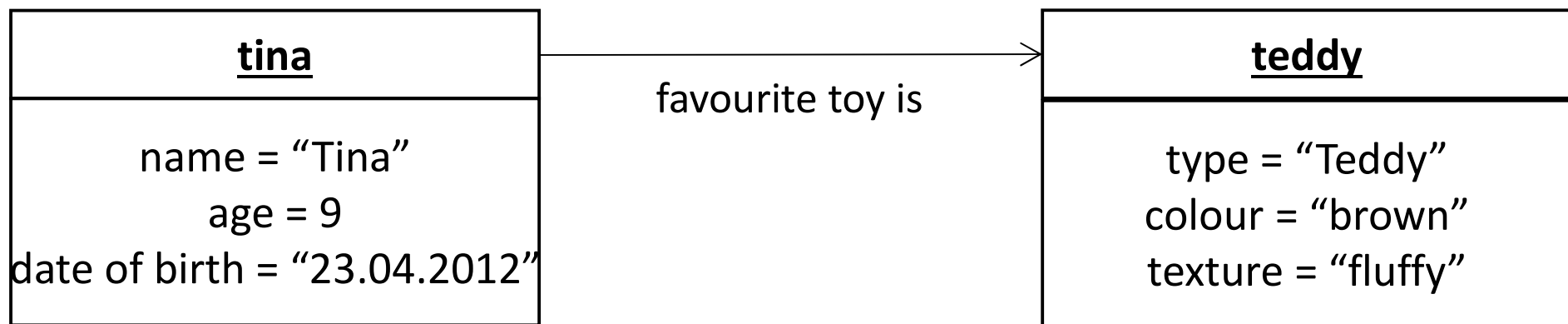
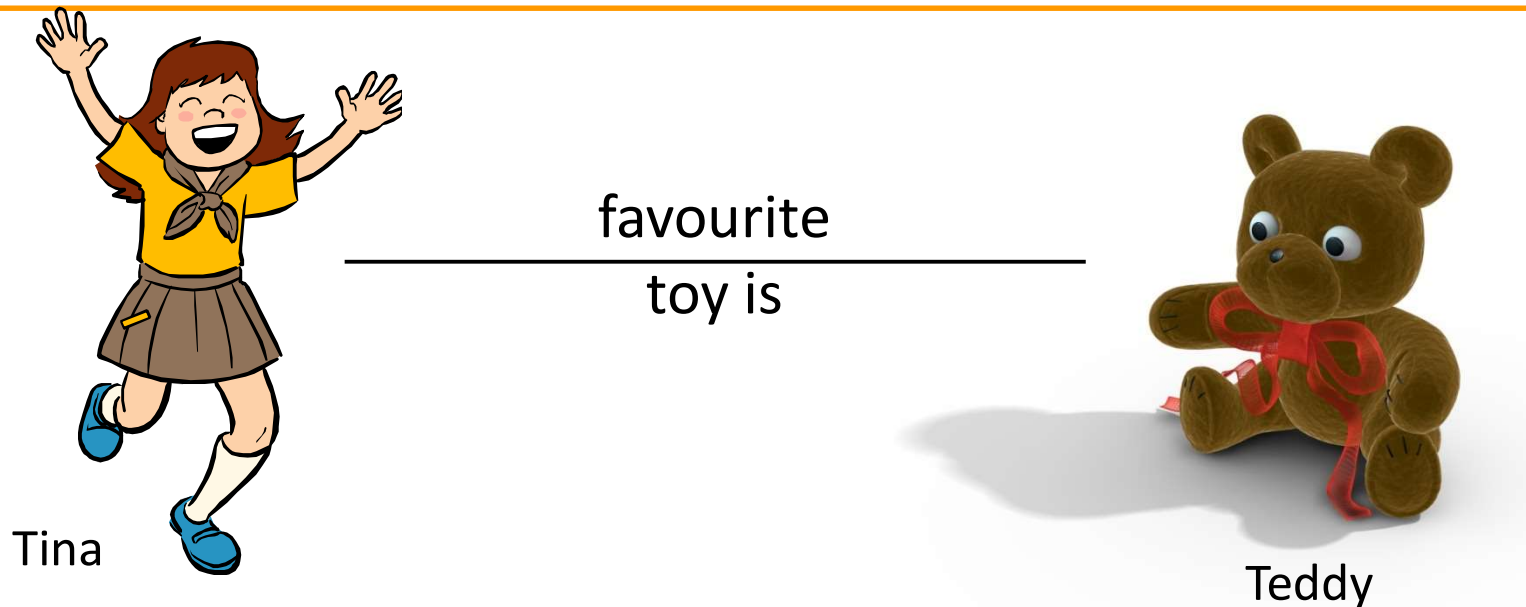
- ⊞ Rectangle usually with only two areas
 - ⊞ Object identifier (underlined)
 - ⊞ Attributes, possibly with specific value assignments (type specification possible)
- ⊞ Relationships to other objects possible
- ⊞ Attributes and relationships within the scope of the specifications through class diagram

➤ Variant


- ⊞ Attributes not explicitly displayed
- ⊞ Object symbol then has only one field



Object diagram – example 1



Example: finding objects and attributes

- You've probably already picked up ice cream in a cone from the ice cream parlour.
 - What questions were you typically asked?
 - ⊞ How many scoops of ice cream?
 - ⊞ In a cone or cup?
 - ⊞ What kind of ice cream
 - My standard response
 - ⊞ Two scoops
 - ⊞ In a cone
 - ⊞ Blueberry and pistachio
- 
- How can this be represented with object-oriented modelling?

Example – key questions for OO modelling

- Key questions for finding objects and attributes
 1. What are the things we are talking about here?
 2. What are the relevant properties of the things being considered?
 3. Are these things somehow related? If so, how?

- Task
 - ⊞ Answer the questions for the ice cream parlour situation described!



http://bilder.t-online.de/b/54/65/70/60/id_54657060/tid_da/kinder-im-hoersaal.jpg

Exercise – Object diagram (1)

➤ Task

- ✦ Please use pen and paper.
- ✦ Based on the results of the previous task, create an object diagram for the ice cream parlour situation.
- ✦ You have 5 minutes.



Example – key question 1

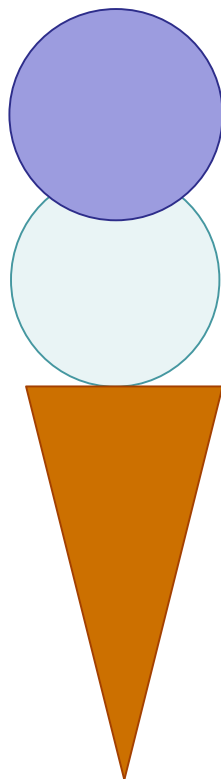
- What are the things we are talking about here?

- My standard ice cream
 - ⊞ Two scoops
 - ⊞ In a cone
 - ⊞ Blueberry and pistachio

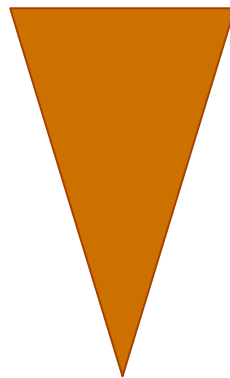


Example – possible solution (1)

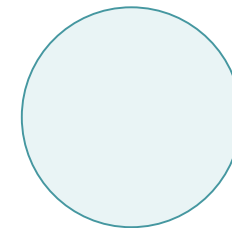
1. What are the **things** we are talking about here?



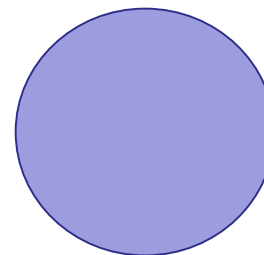
My standard
ice cream



Cone



A scoop of
pistachio flavour
ice cream



A scoop of
blueberry flavour
ice cream

Example – possible solution (1 – OO)

- Things that we are talking about here – represented in an object-oriented manner

my ice cream

My standard ice cream

scoop of pistachio

A scoop of pistachio flavour ice cream
which goes in my ice cream

scoop of blueberry

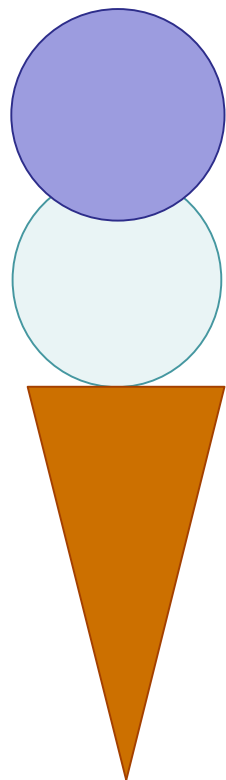
A scoop of blueberry flavour ice cream
which goes in my ice cream

cone

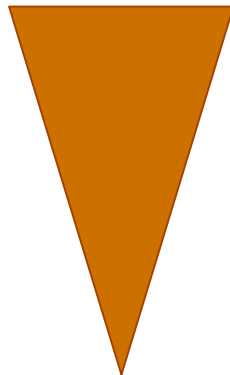
The ice cream cone my
ice cream goes into

Example – key question 2

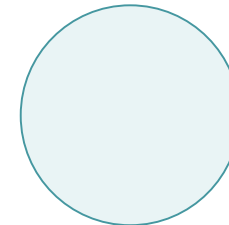
- What are the relevant **properties** of the things being considered?



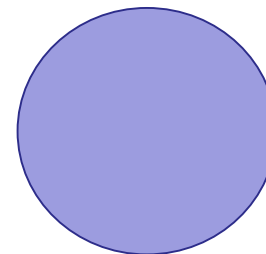
My standard
ice cream



Cone



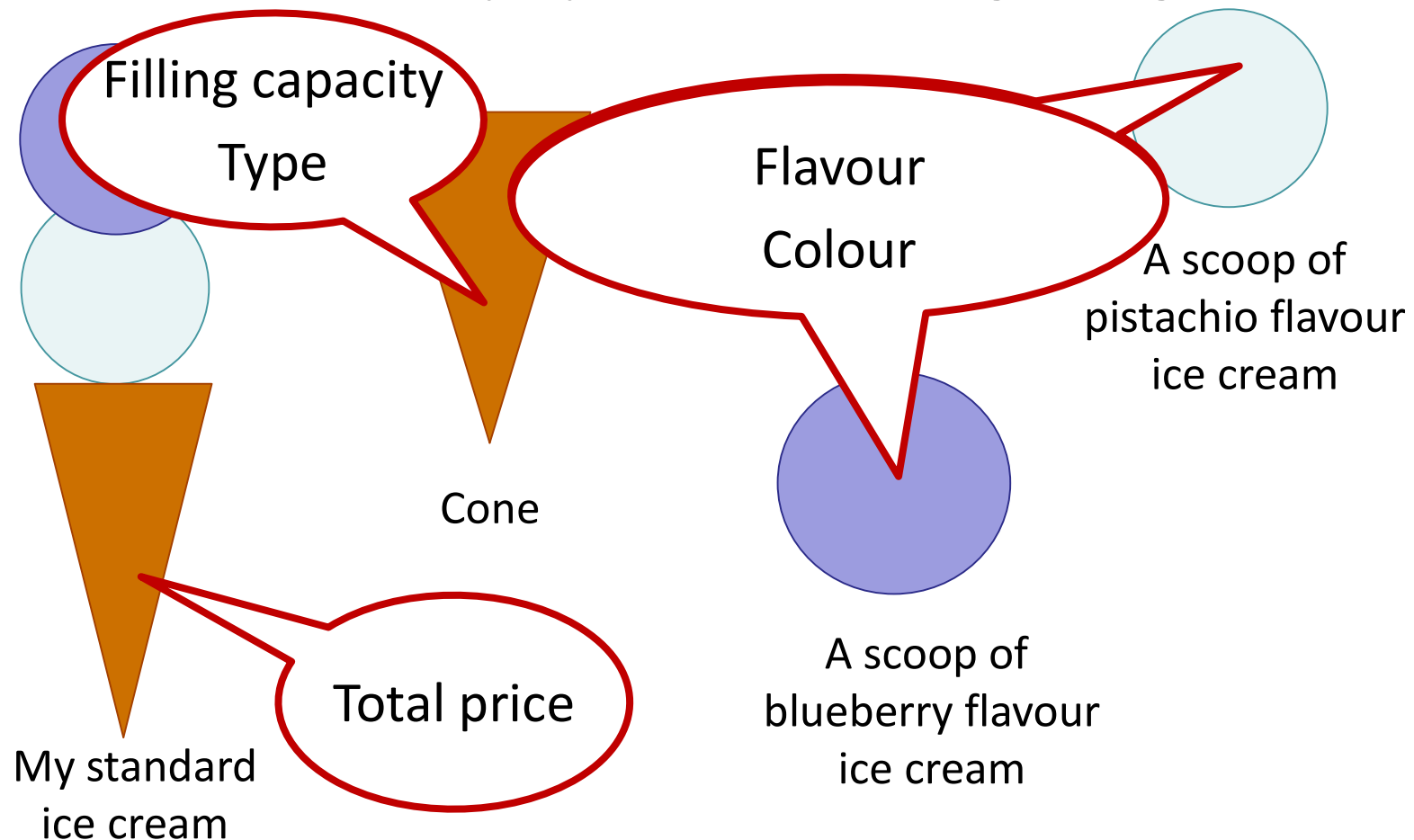
A scoop of
pistachio flavour
ice cream



A scoop of
blueberry flavour
ice cream

Example – possible solution (2)

- What are the relevant properties of the things being considered?



Example – possible solution (2 – 00)

- Relevant properties of the things – represented in an object-oriented manner

<u>my ice cream</u>
total price = 2 Euros

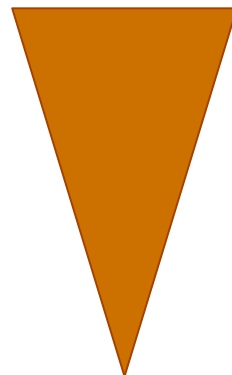
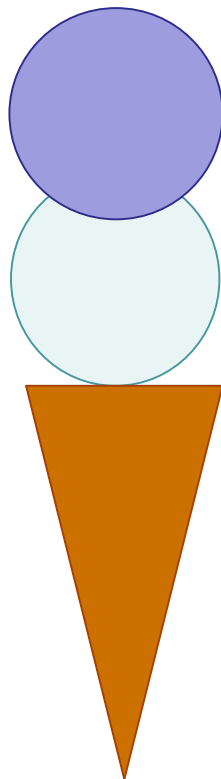
<u>scoop of pistachio</u>
flavour = pistachio colour = green

<u>cone</u>
capacity = 2 type = cone

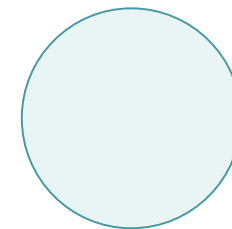
<u>scoop of blueberry</u>
flavour = blueberry colour = purple

Example – key question 3

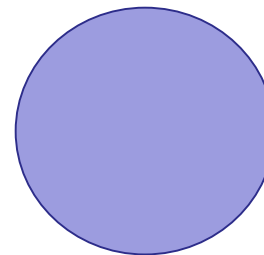
- Are these things somehow **related**? If so, how?



Cone



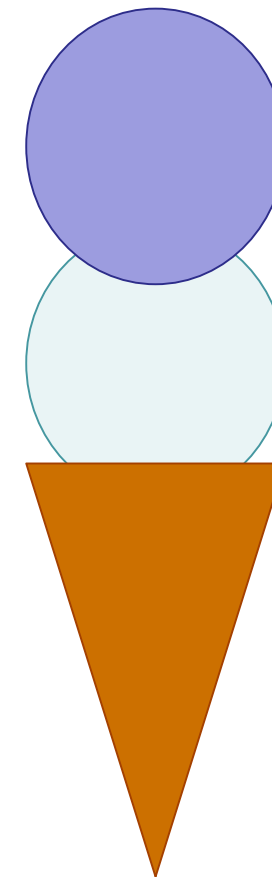
A scoop of
pistachio flavour
ice cream



A scoop of
blueberry flavour
ice cream

Example – possible solution (3)

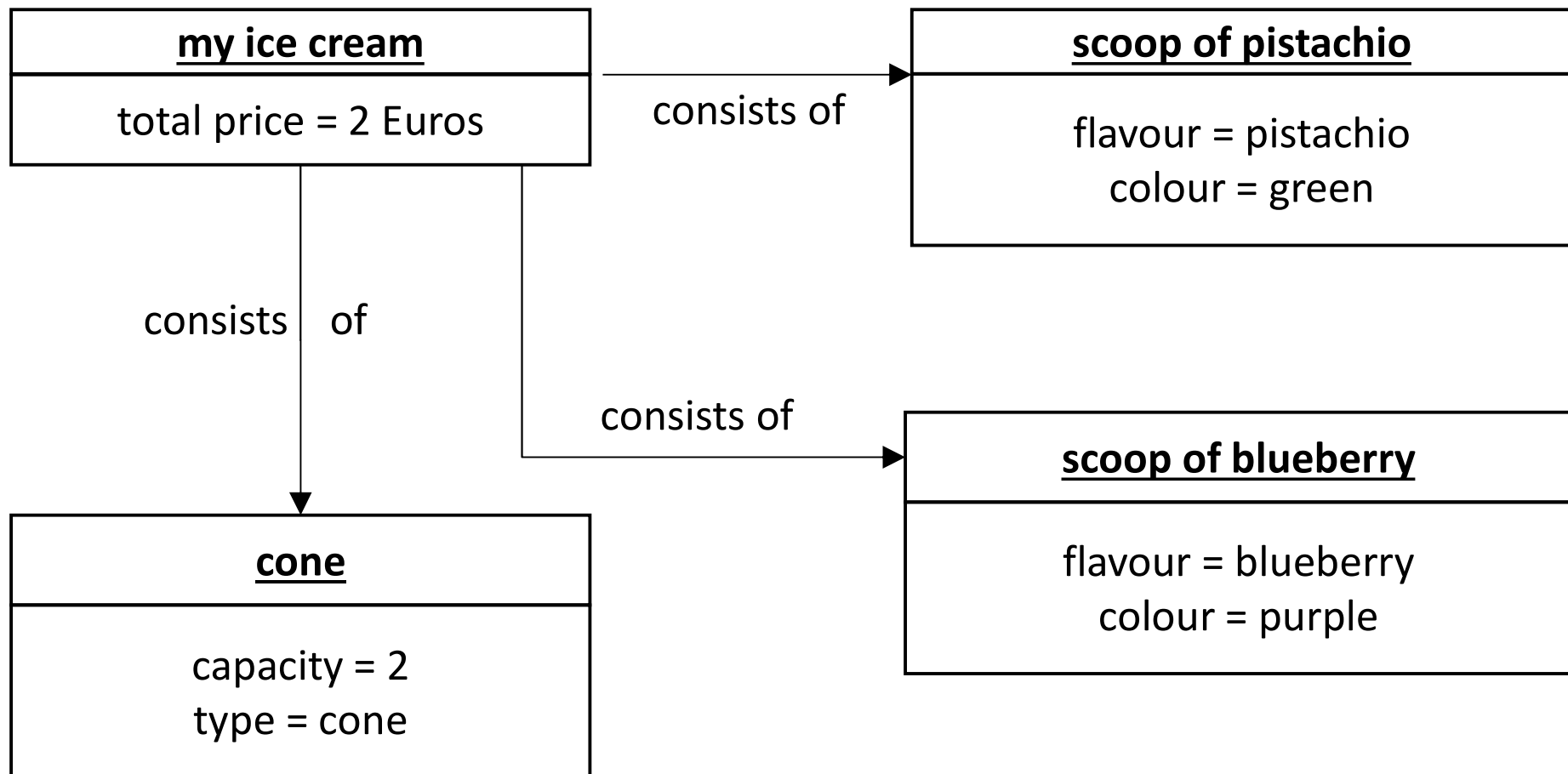
- Are these things somehow related?
If so, how?
- ✚ My standard ice cream consists of...
 - ✚ a cone
 - ✚ a scoop of pistachio flavoured ice cream and
 - ✚ a scoop of blueberry flavoured ice cream.



My standard
ice cream

Example – possible solution (3 – 00)

- Relationships between the things – represented in an object-oriented manner



Exercise – Object diagram (2)

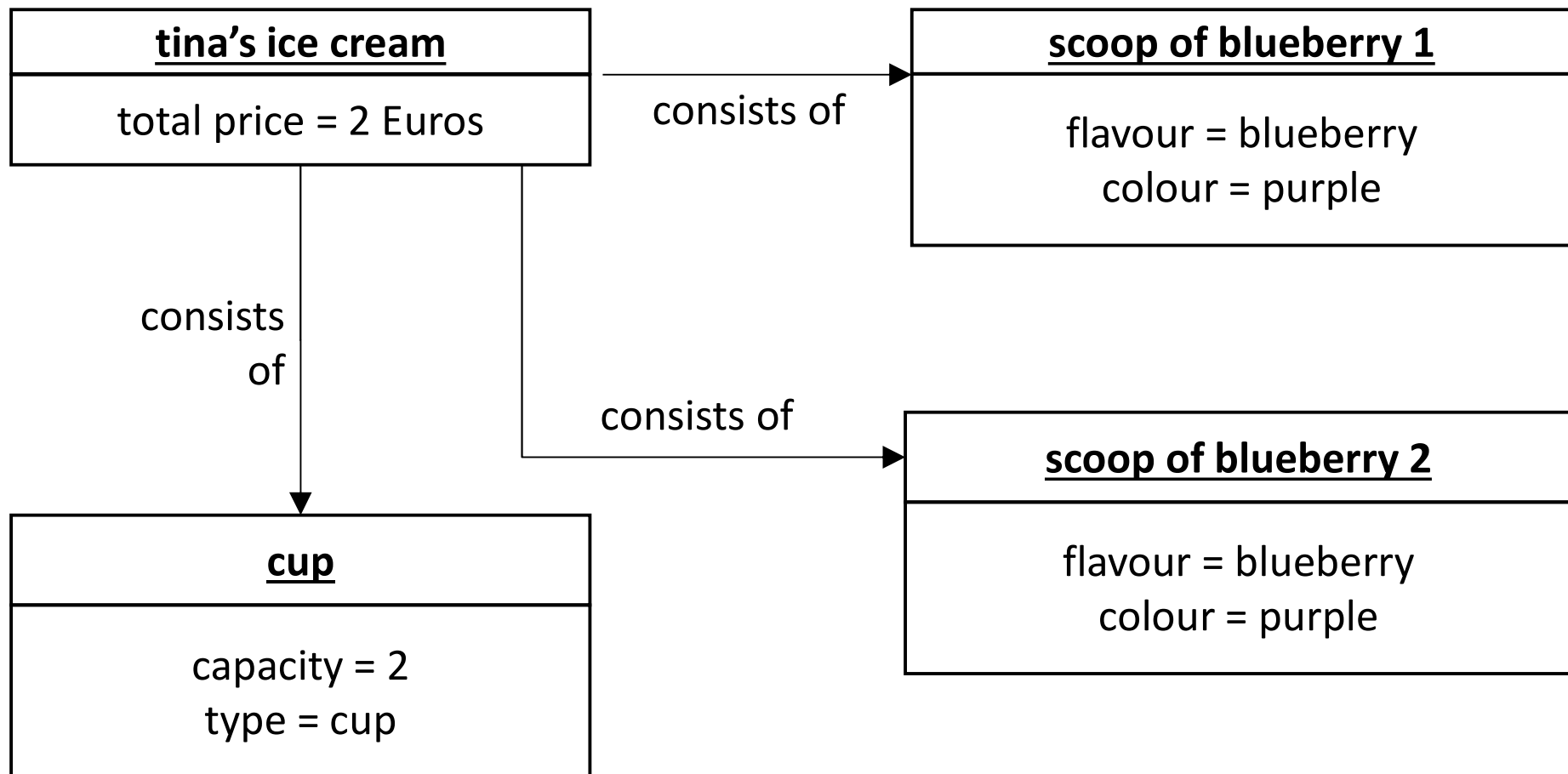
➤ Task

- ✦ Create a second object diagram by modelling Tina's ice cream.
- ✦ Tina wants two scoops of blueberry ice cream in a cup.
- ✦ You have 5 minutes.



Exercise – Possible solution (OO)

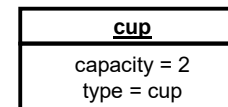
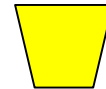
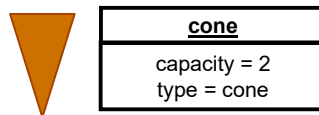
- Tina's ice cream – represented in an object-oriented manner



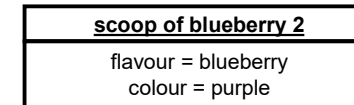
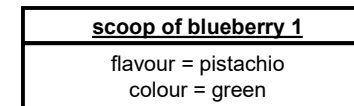
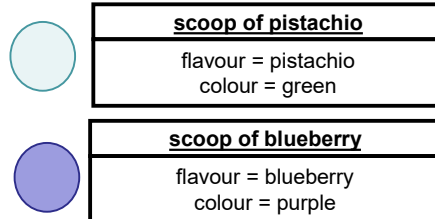
Exercise – Questions about the result (1)

➤ What have we now used overall?

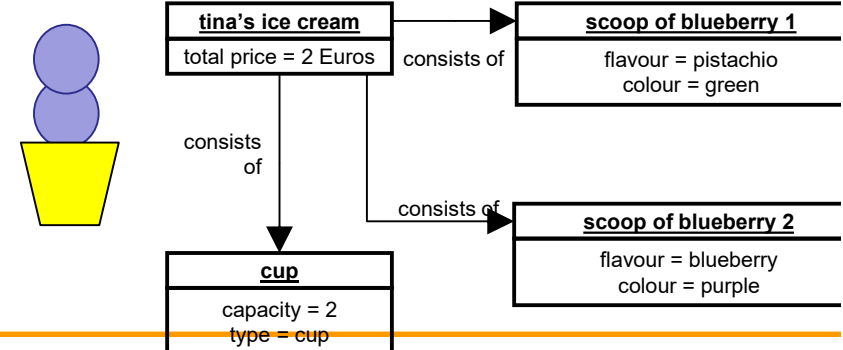
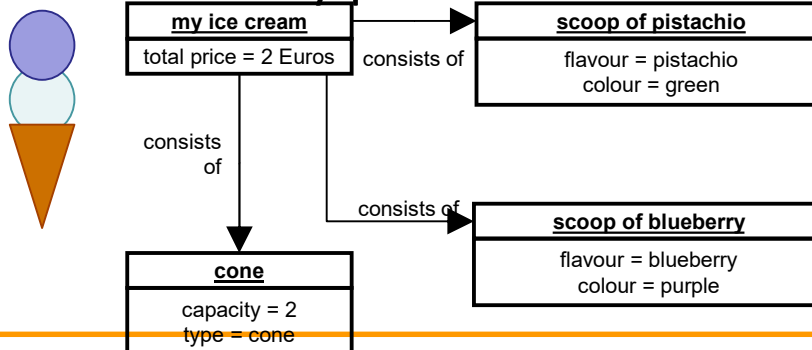
⊞ How many containers (cups/cones)?



⊞ How many scoops of ice cream?

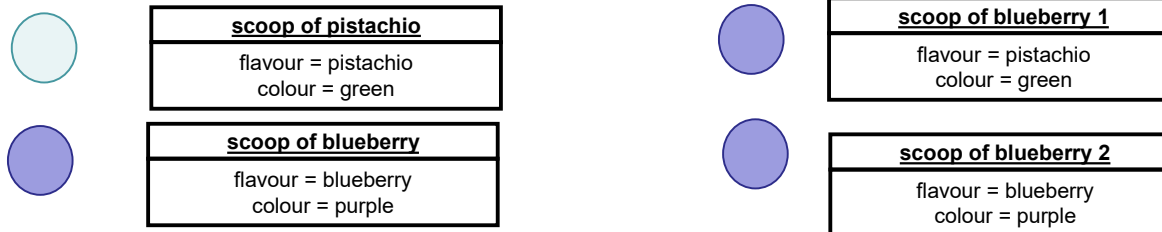


⊞ How many portions of ice cream were served overall?



Exercise – Questions about the result (2)

➤ What does this mean for the individual objects?

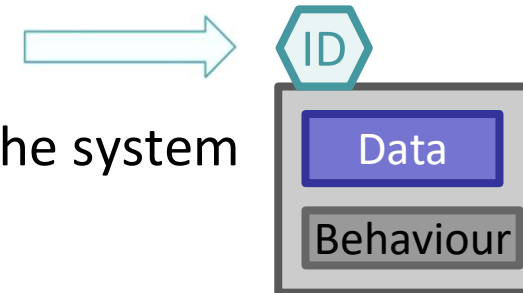


- ✦ Multiple similar objects possible at the same time
 - ✦ Contents (= state, data values) of the objects are independent of each other
 - ✦ States can be different
 - ✦ Or also identical
- Objects with the same state are nevertheless clearly distinguishable from each other!!!

Identity of an object

➤ Identity of an object

- ✚ Object is uniquely identifiable throughout the system
- ✚ Object identifier is set "at birth" of the object
- ✚ Object identity cannot be changed!!!
- ✚ Object identity is independent of the current state of the object
- ✚ For an object in a running program, corresponds to the address at which the object is located in memory



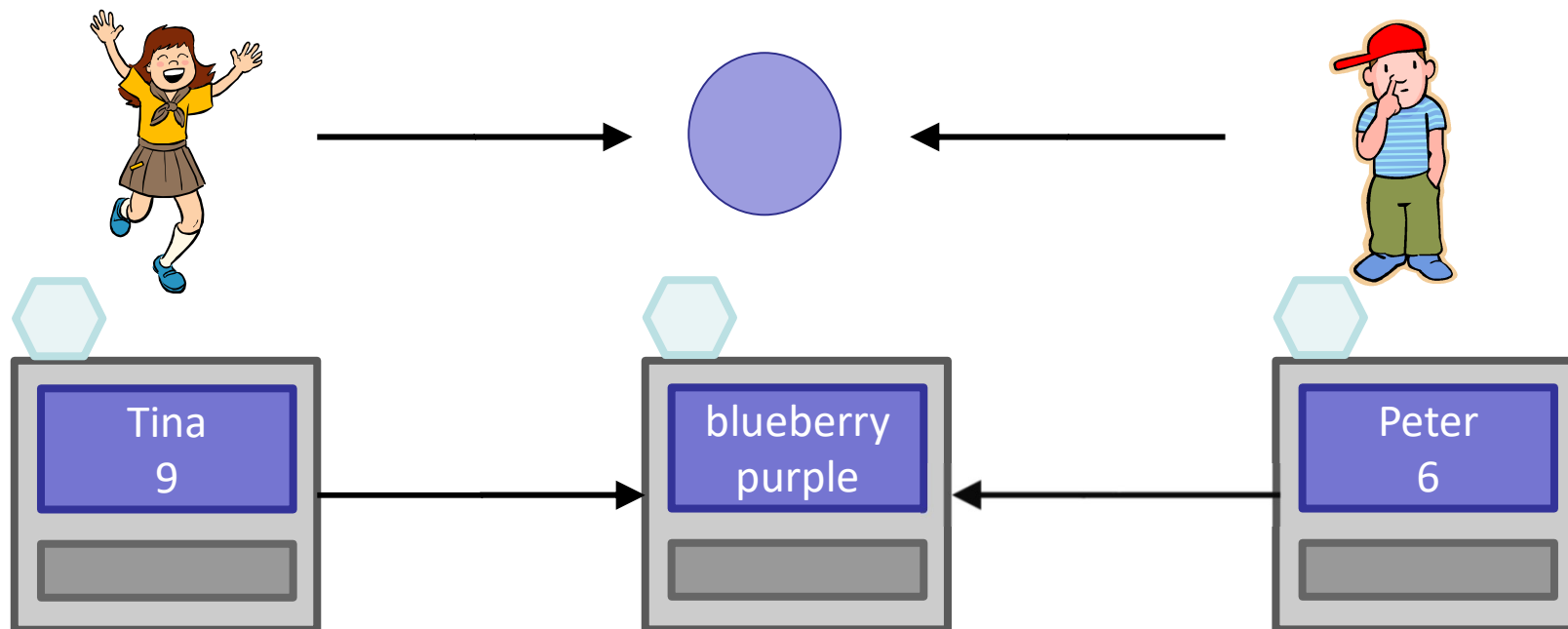
➤ Please note

- ✚ Object identity cannot be changed over the lifetime of the object
- ✚ In contrast, the object state can be changed

Identity of objects – example

➤ Example

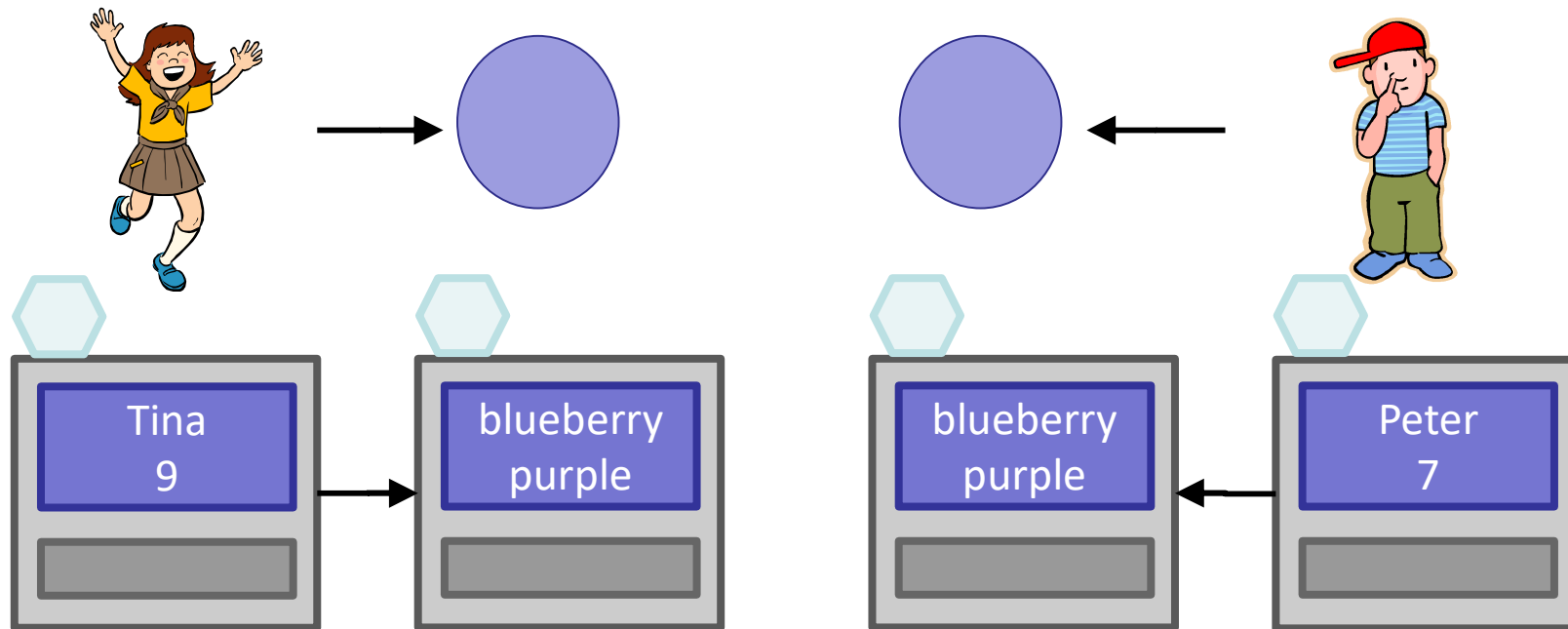
- ✚ Tina and Peter eat *one* scoop of blueberry ice cream
- ✚ Two references to the same blueberry ice cream



Equal contents of objects – example

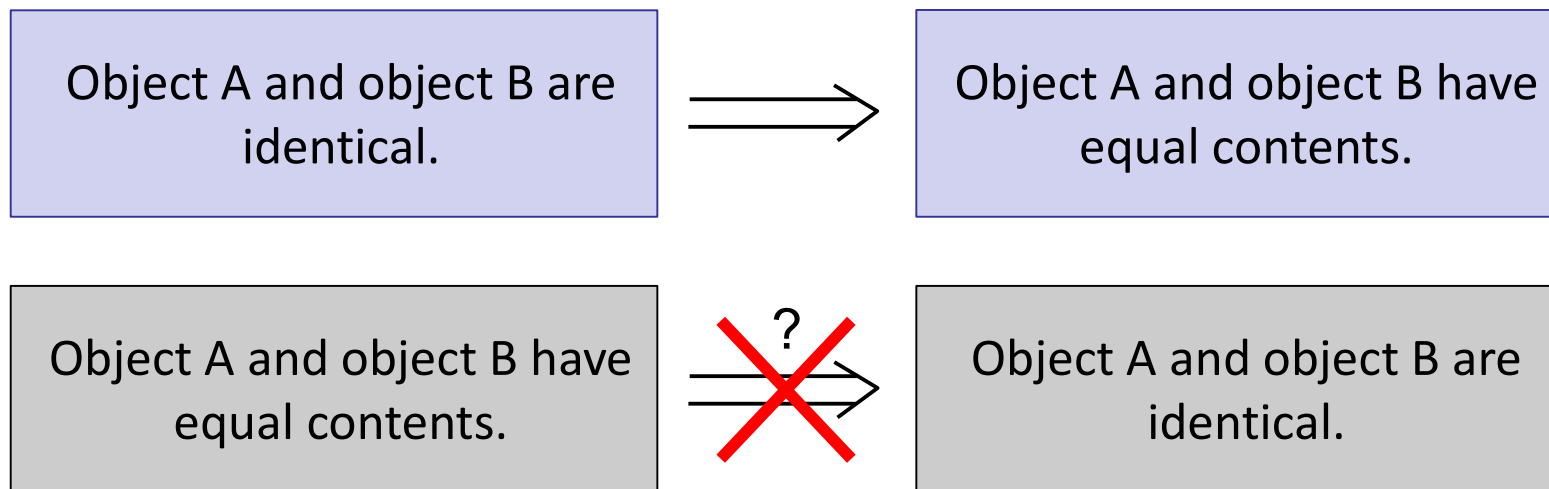
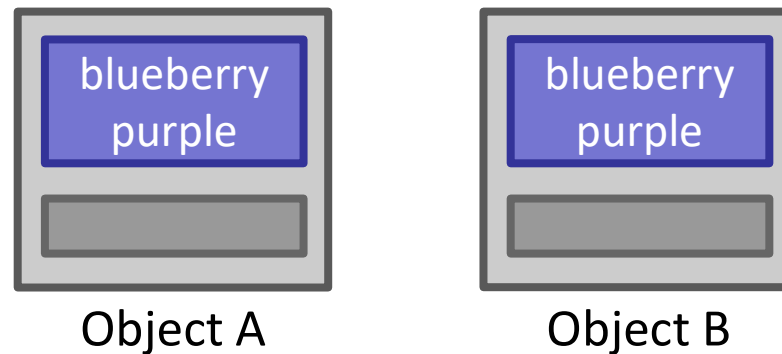
➤ Example

- ✦ Tina and Peter *each* eat *their own* scoop of blueberry ice cream
- ✦ Two references to two different scoops of blueberry ice cream
- ✦ Objects with the same state, but not identical



Identity vs. equal contents

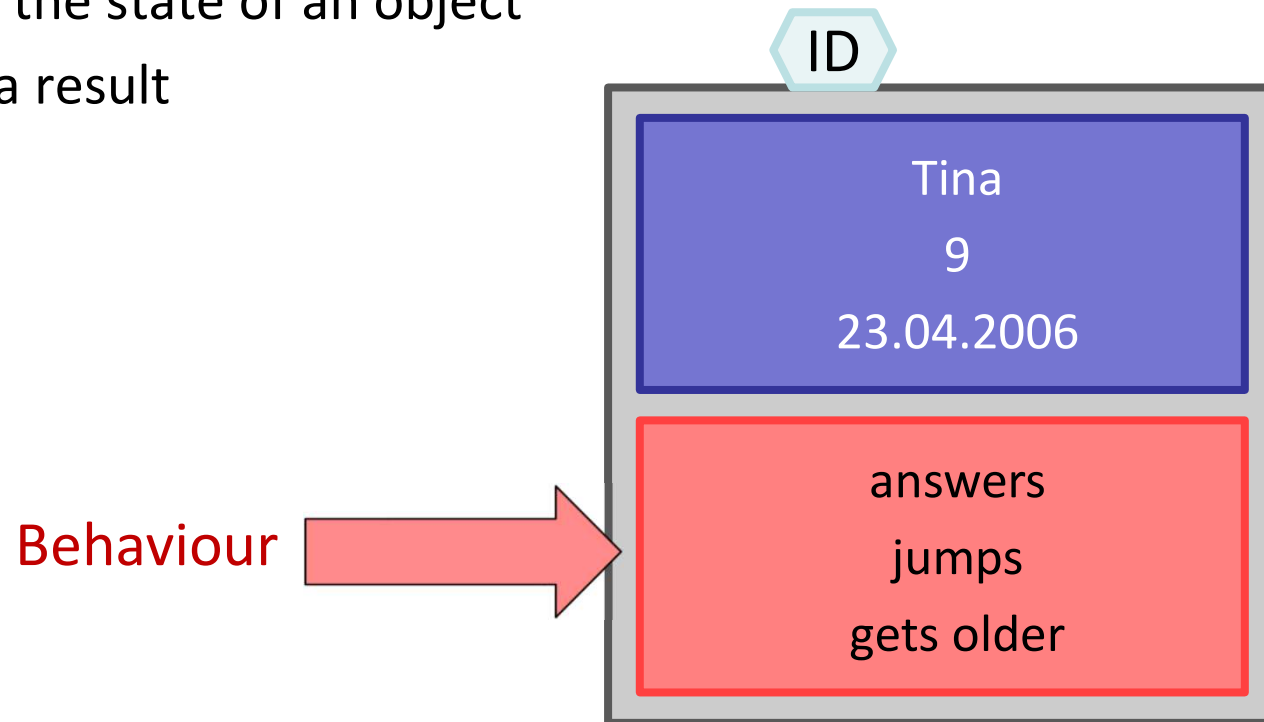
➤ Remember



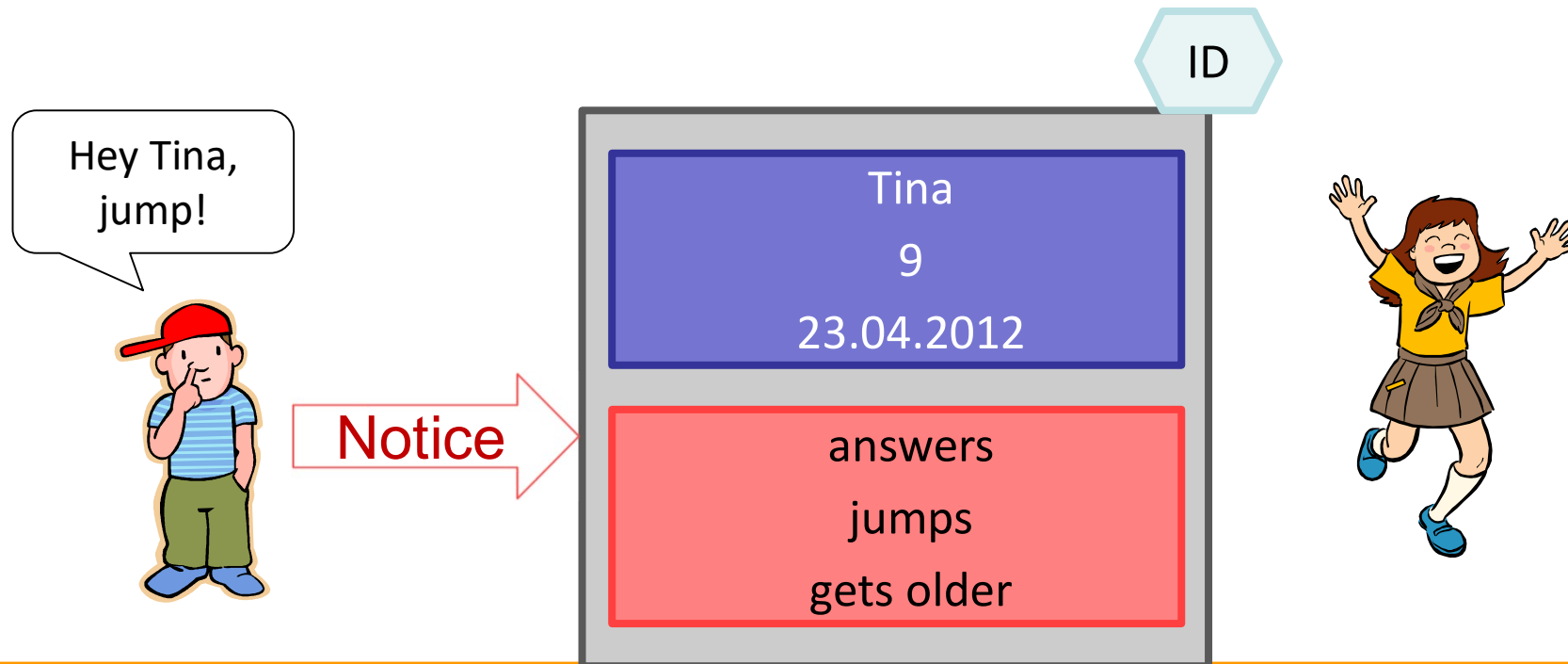
Behaviour of an object

➤ Behaviour

- ✦ Operations (actions) that an object can perform
- ✦ Determines which messages an object responds to
- ✦ Can change the state of an object
- ✦ Can return a result



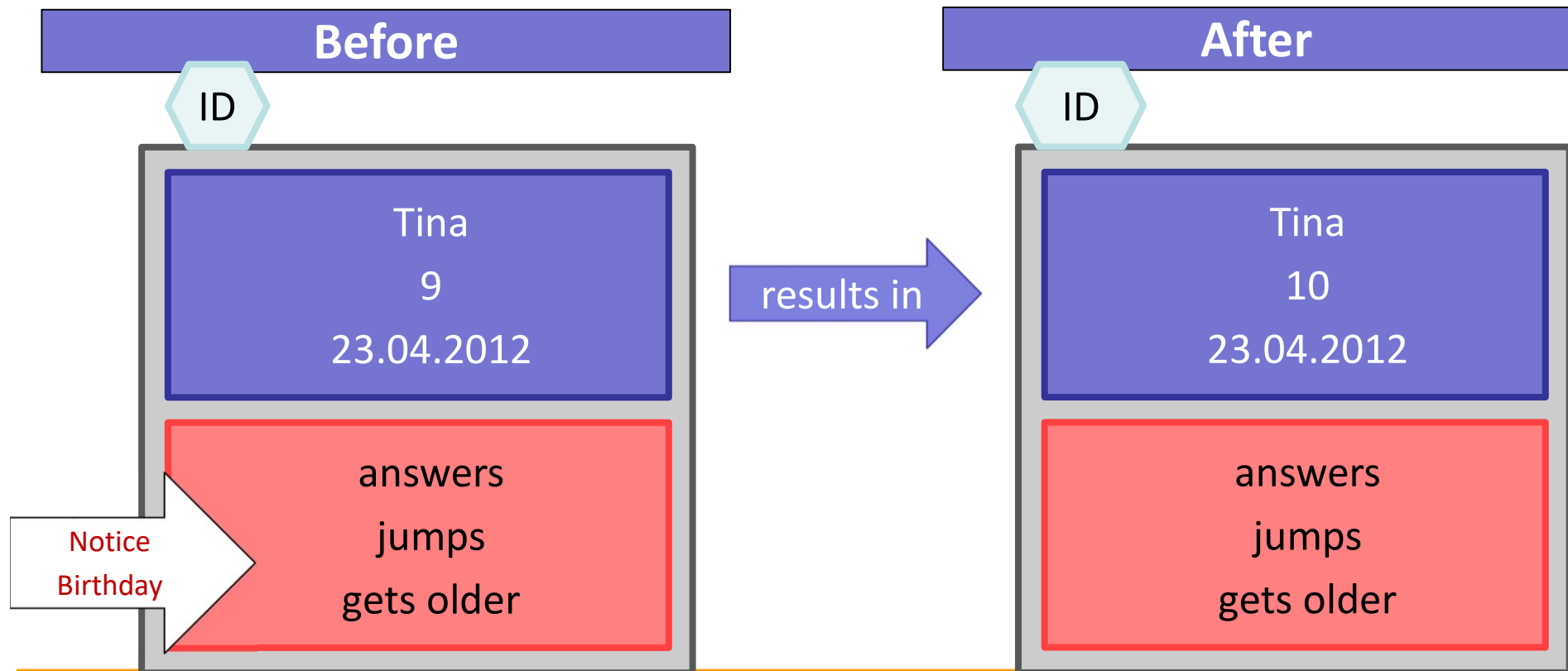
- Operation realises specific behaviour of the object
 - ✦ Is called by anyone
 - ✦ "Anyone" is usually also an object (another or the same)
 - ✦ Call triggers execution of the behaviour



Manipulative operation

- Manipulative operation changes the value of the object's attributes

⊞ Thereby changes the state of the object



Requesting operation



- Requesting operation returns a result to the caller
 - ⊞ Caller can continue working with the result if necessary
 - ⊞ Answer may depend on the state of the object
 - ⊞ Requesting operation is also called **function**

