Object and Classes

juampi

Concept:

- Java objects models objects from a problem domain.

Concept:

- Objects are created from **classes**. The class describes the kind of objects; the **object** represent individual instantiations of the class.

Concept:

- Objects are created from **classes**. The class describes the kind of objects; the **object** represent individual instantiations of the class.

Calling methods

- We can communicate with objects by **invoking methods** (message send) on them. Objects usually do something if we invoke a method.

Parameters

- Methods can have **parameters** to provide additional information for a task.

Concept:

- The header of the method is called its **signature**, it provides information needed to invoke that method.

Data types

- Parameters have **types.** The type defines what kinds of values a parameter can take.

Multiple instances

- Many similar objects can be created from a single class.

State

- Objects have state. The **state** is represented by storing values in fields.

Return values

- Methods may return information about an object via return value.

Object as parameters

- Methods may return information about an object via return value.

Exercises

- What are the types of the following values?
  - 0
  - “hello”
  - 101
  - true
  - -1
  - 3.1415

The clock example

11:03

Abstraction

- **Abstraction** is the ability to ignore details of parts, to focus attention on a higher level of a problem.

Modularization

- **Modularization** is the process of dividing a whole into well-defined parts that can be built and examined separately and that interact in well-defined ways.

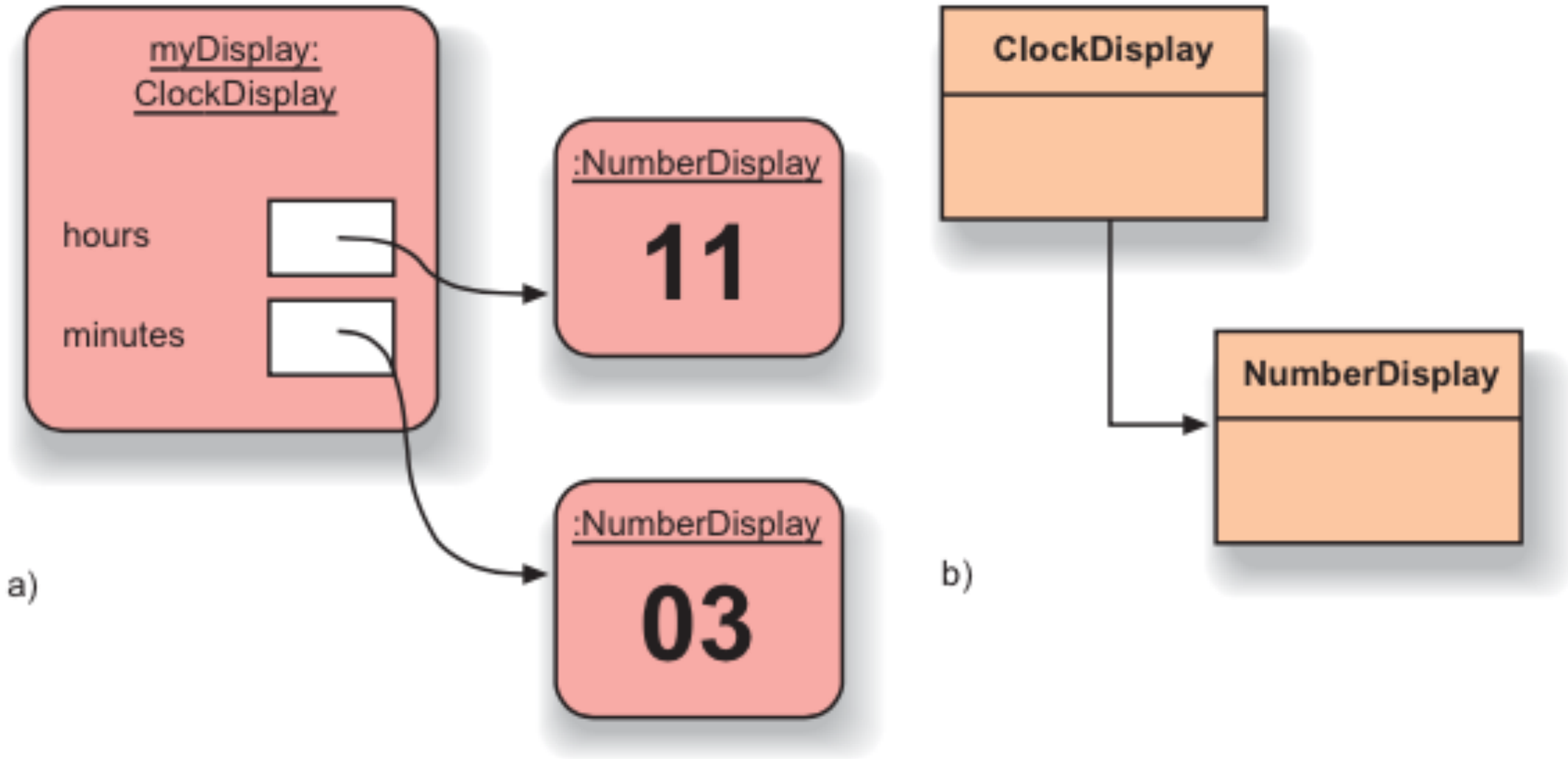
Two digit number

03

Classes define types

- A class name can be used as the type for a variable. Variables that have a class as their type can store objects of that class.

Class diagrams vs object diagrams



Primitives types and object types

- The primitive types in JAVA are the non-objects types. Types such as int, boolean, char, double, and long are the most common primitive types. Primitive types have no methods.

Object creating objects

- Objects can create other objects, using the **new** operator.

Multiple constructors

- A class may contain more than one constructor.

Overloading

- A class may contain more than one method of the same name, as long as each has a distinctive set of parameter types.

Internal method calls

- Methods can call other methods of the same class as part of their implementation. This is called an **internal method call.**

External method calls

- Methods can call methods of other objects using dot notation. This is called an **external method calls.**