Object interaction

Creating cooperating objects

A digital clock

11:03
Abstraction and modularization

- **Abstraction** is the ability to ignore details of parts to focus attention on a higher level of a problem.
- **Modularization** is the process of dividing a whole into well-defined parts, which can be built and examined separately, and which interact in well-defined ways.

Example: Car Design (I)

- A design who designs the body shape of a car can abstract from the technical details of the wheels and the engine.
- However, a design who designs the wheels and engine must consider their technical details.
Example: Car Design (II)

- Why are cars built successfully?
  - A car is divided into independent modules; separate people work on these modules independently
  - Abstraction is used to put smaller modules together to create bigger modules

Modularizing the clock display

One four-digit display?

Or two two-digit displays?
Quiz

• How do compare the two design, a single four-digit display or two two-digit displays?

Implementation - NumberDisplay

```java
public class NumberDisplay {
    private int limit;
    private int value;

    Constructor and methods omitted.
}
```
public class ClockDisplay
{
    private NumberDisplay hours;
    private NumberDisplay minutes;

    Constructor and methods omitted.
}

Class diagram (I)

- Class diagram shows the classes of an application and the relationships between them.
- Presents a static view of a program.
Object-oriented Programming in Java

Class diagram (II)

Object diagram (I)

- Shows the objects and their relationship during execution.
- Presents the dynamic view of a program.
Primitive types vs object types (I)

- **Primitive type**: No methods, predefined in Java. Common ones include int, boolean, char, double, and long.
- **Object type**: Defined by classes, includes data and methods that operate on the data.
Primitive types vs. object types (II)

SomeObject obj;  

int i;  

Quiz: What is the output?

• int a;
  int b;
  a = 32;
  b = a;
  a = a + 1;
  System.out.println(b);

• Person a;
  Person b;
  a = new Person("Everett");
  b = a;
  a.changeName("Delmar");
  System.out.println( b.getName());
Object-Oriented Programming in Java

Primitive types vs. object types (III)

```
ObjectType a;

ObjectType b;

___________ b = a; ____________

int a;       int b;

32           32
```

Source code: NumberDisplay

```
public NumberDisplay(int rollOverLimit) {
    limit = rollOverLimit;
    value = 0;
}

public void increment() {
    value = (value + 1) % limit;
}
```
The modulo operator

- The 'division' operator (/), when applied to int operands, returns the result of an integer division.
- The 'modulo' operator (%) returns the remainder of an integer division.
- E.g., generally:
  \[ 17 \div 5 = \text{result 3, remainder 2} \]
- In Java:
  \[ 17 \div 5 = 3 \]
  \[ 17 \mod 5 = 2 \]

Quiz

- What is the result of the expression \( (8 \mod 3) \)?
- What are all possible results of the expression \( (n \mod 5) \)?
Source code: NumberDisplay

```java
public String getDisplayValue()
{
    if(value < 10) {
        return "0" + value;
    } else {
        return "" + value;
    }
}
```

Concepts

- abstraction
- modularisation
- classes define types
- class diagram
- object diagram
- object references
- primitive types
- object types
Creating objects (I)

```java
public class ClockDisplay {
    private NumberDisplay hours;
    private NumberDisplay minutes;
    private String displayString;

    public ClockDisplay() {
        hours = new NumberDisplay(24);
        minutes = new NumberDisplay(60);
        updateDisplay();
    }
}
```

Creating Objects (II)

- The “new” operation does two things:
  - Creates a new object of the named class
  - Executes the constructor of that class
- A class may have more than one constructor, as long as they have different signatures.
Method Overloading

- In general, a class may have more than one method with the same name, as long as they have different signatures.

ClockDisplay object diagram
Creating objects

in class NumberDisplay:

public NumberDisplay(int rollOverLimit);

formal parameter

in class ClockDisplay:

hours = new NumberDisplay(24);

actual parameter

Method calling

public void timeTick()
{
    minutes.increment();
    if(minutes.getValue() == 0) {
        // it just rolled over!
        hours.increment();
    }
    updateDisplay();
}
Internal method

/**
 * Update the internal string that
 * represents the display.
 */
private void updateDisplay()
{
    displayString =
        hours.getDisplayValue() + ":" +
        minutes.getDisplayValue();
}

Method calls

• internal method calls
    updateDisplay();
    ...
    private void updateDisplay()

• external method calls
    minutes.increment();
Method calls (2)

`object . methodName ( parameter-list )`

Concepts

- object creation
- overloading
- internal/external method call
Example: Mail system

- Three classes: MailServer, MailClient, and MailItem
- A single mail server object must be created.
- Several mail client objects can be created. Each is identified with a user name.
- Mail items can be sent and received between clients, via the mail server.

The this keyword (I)

```
public class mailItem() {
    private String from;
    private String to;
    private String message;
    public MailItem(String from, String to, String Message) {
        this.from = from;
        this.to = to;
        this.message = message;
    }
    ...
}
```
The this keyword (II)

• By default, the definition in the closest enclosing block is used.
• The this keyword is used to indicate that the variable is a field, instead of a local variable.

Quiz

• Why don’t we simply use a different name?
Debugger

• A software tool that helps in examining how an application executes.

Debugger commands

• Breakpoint: Stop the execution at a particular line of source code
• Step: Execute a single line of code, and then stop again
• Step Into: Step into the execution of a method, and stop at the first line inside the method.
Acknowledgement

The original authors of these slides are the authors of the textbook. The instructor made necessary modifications, with permissions from the authors.