Computer Programming

Exceptions

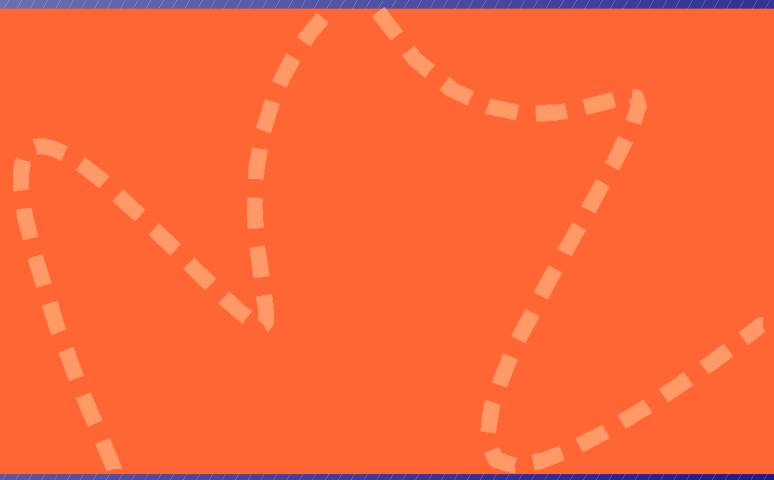
Threads

Willy Picard
Department of Information Technology
The Poznan University of Economics
<*picard@kti.ae.poznan.pl*>

Agenda

- Lecture Goal(s)
- Refreshments
- Exceptions
- Threads
- Conclusion

Lecture Goal(s)



Lectures Overview

Fundamental Concepts

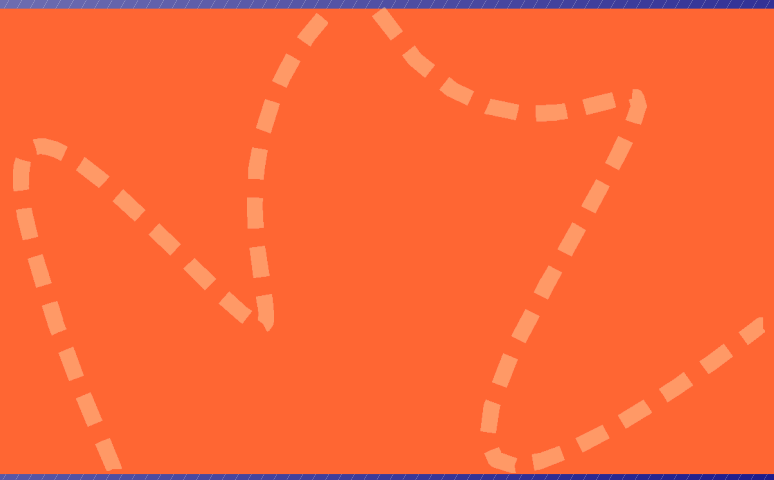
- 1: Introduction
- 2: Basic data structures & Statements
- 3: Object-oriented programming I
- 4: Object-oriented programming II
- 5: Object-oriented programming III
- 6: Complex data structures
- 7: Threads and Exception handling

LECTURE GOALS

Today's Goal

To provide programming knowledge about exceptions and threads

Refreshments

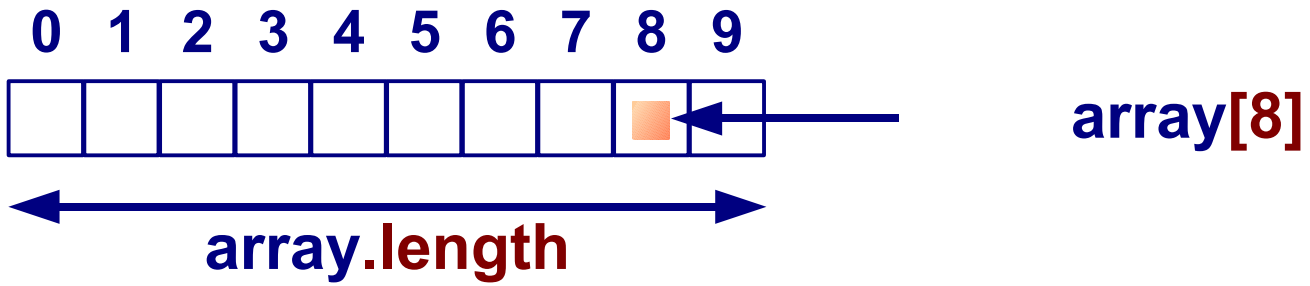


The API Specification

- Documentation
  - Packages
  - Interfaces
  - Classes
  - Inheritance
  - Attributes
  - Methods

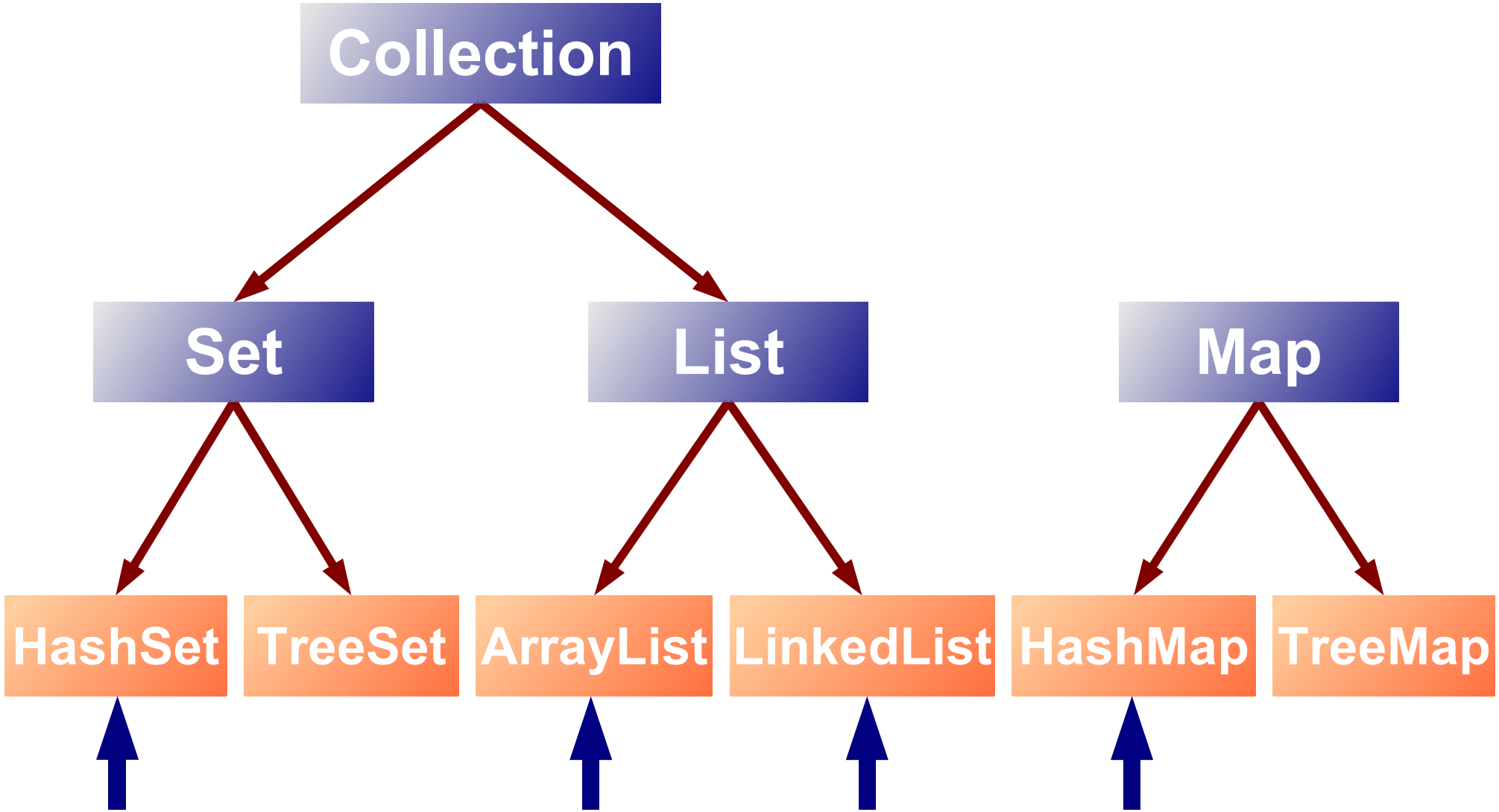
REFRESHMENTS

Arrays in Java



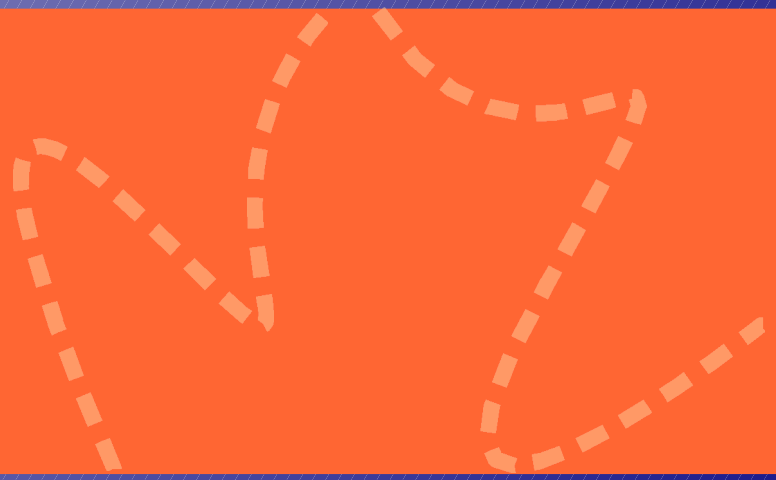
REFRESHMENTS

Implementations



REFRESHMENTS

Exceptions



Ariane 5

- June 4, 1996, maiden flight
  - 40 seconds
  - 500$ million lost
- Cause
  - A software error
  - An exceptional event not handled

Exception Definition

An exception is an event during program execution that prevents the program from continuing normally

Example of Exception in Java

```
List myCats = new ArrayList();
myCats.add(new Cat("Felix", 1200));
Cat myThirdCat = myCats.get(3);
```

- `IndexOutOfBoundsException`
- Check the Java API !!!

Exception Handler Definition

An exception handler is a block of code that reacts to a specific type of exception

Try-catch-finally in Java

- Syntax

```
try {
   <some statements>
} catch (ExceptionType type){
   <some statements>
} catch (ExceptionType2 type2){
   <some statements>
} finally {
   <some statements>
}
```

Example of Try-catch-finally

```
List myCats = new ArrayList();
myCats.add(new Cat("Felix", 1200));
Cat myThirdCat;
try {
   myThirdCat = myCats.get(3);
} catch (IndexOutOfBoundsException e){
   System.out.println("Only "+
            myCats.size()+" cats known");
   System.err.println(e.getMessage());
}
```

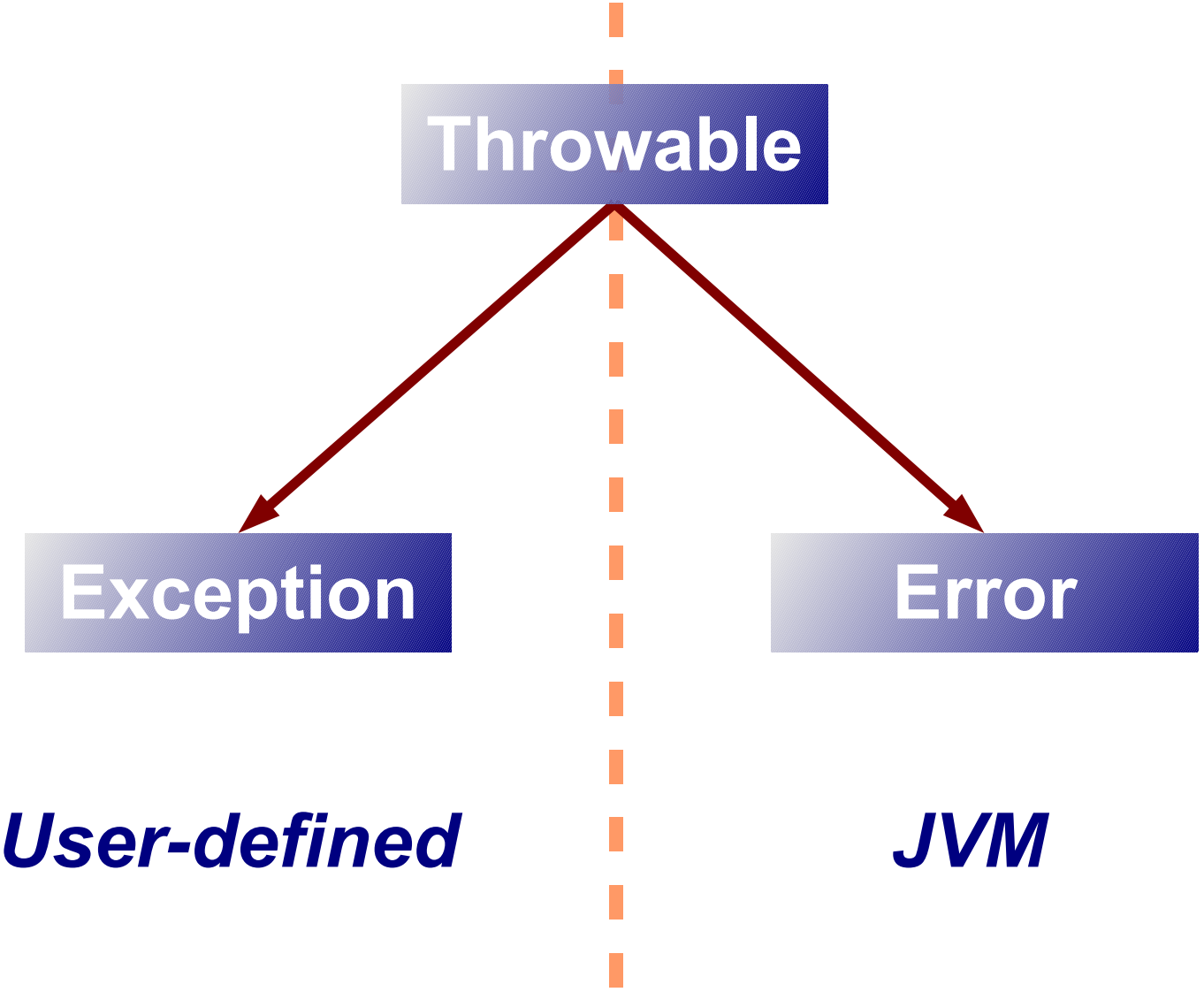
Methods and Exceptions

- Syntax
  - <method definition> throws <exceptionType>{
    ...
    }
- Example
  - public void eat(int foodAmount)
    throws Exception {
    ...
    }

Throwing an Exception

- Syntax
  - <method definition> throws <exceptionType>{
    ...
    throw new <exceptionType>();
    }
- Example
  - public void eat(int foodAmount)
    throws Exception {
    if (foodAmount <= 0){
    thrown new Exception("Only positive"+
    "food amounts for feeding!");
    }
    ...
    }

Exception vs Error



Defining an Exception

- Syntax
  - Class <exceptionName> extends Exception{

```
    ...
    throw new <exceptionType>();
    ...
}
```

Defining an Exception

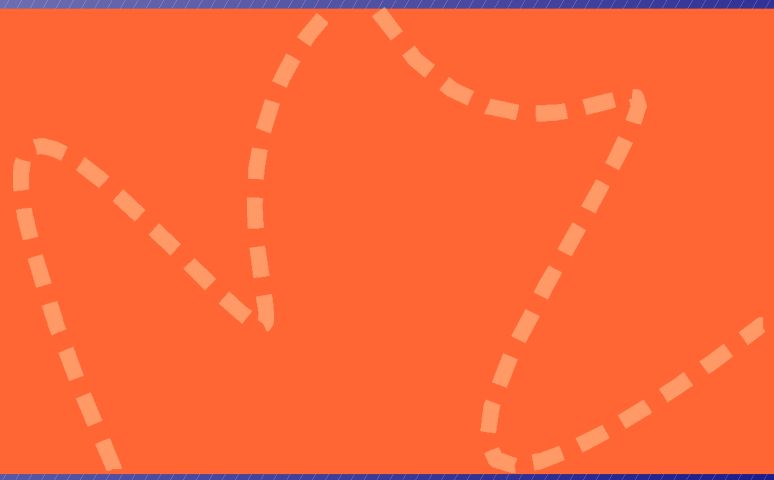
- Example

```
public class InvalidFoodAmountException
   extends Exception{
 private int _amount;

 public InvalidFoodAmountException(int amount){
   super("Food amount "+amount+
      " is not valid");
   _amount = amount;
 }

 public int getAmount(){
  return _amount;
 }
}
```

Exception Example

Exceptional Animals

Threads



Why Multi-threading?

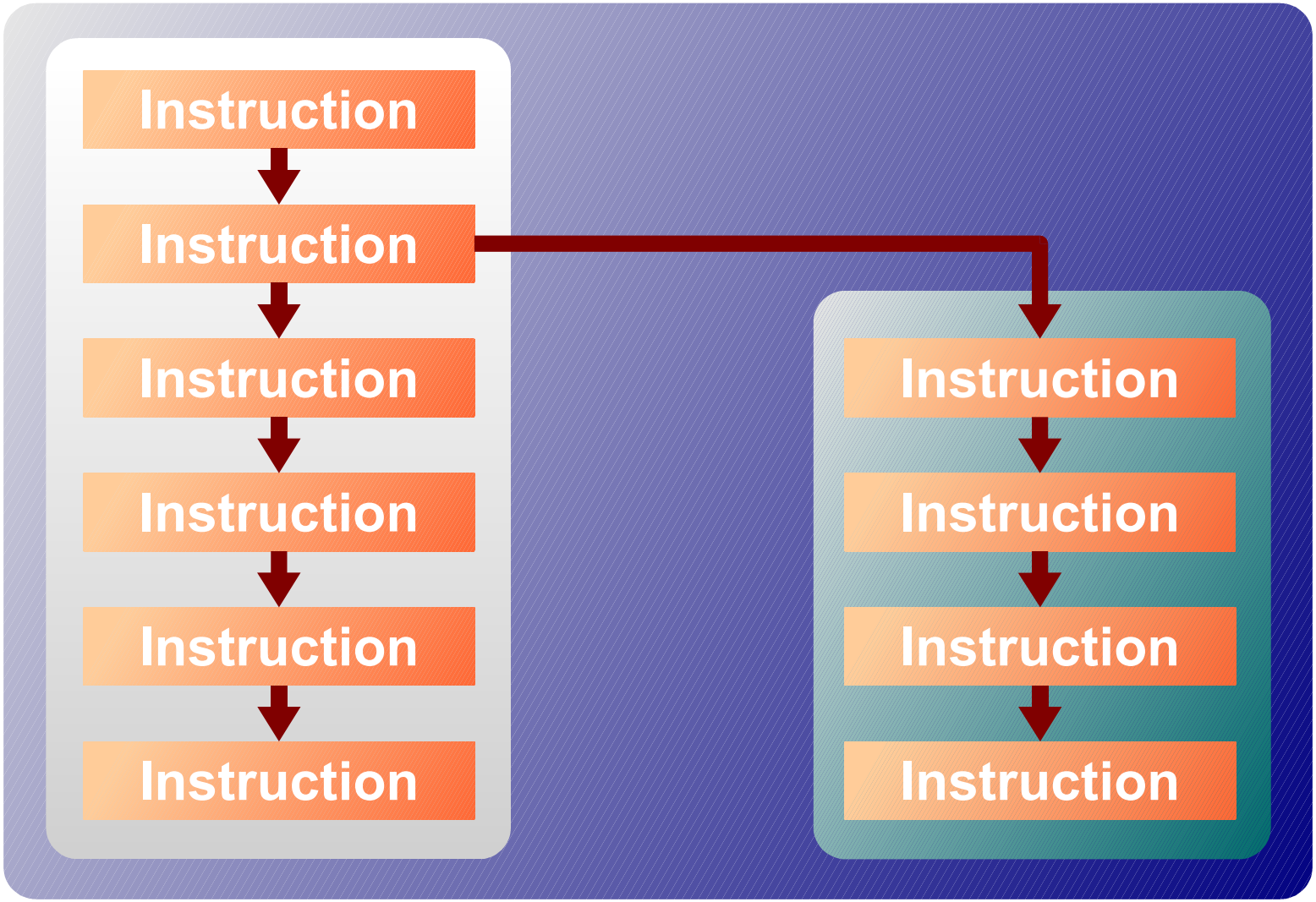
- A program
  - A sequence of task

    OR
  - Many sequences of tasks

Thread Definition

A thread is an a single sequential flow of control within a program

A Multi-Threaded Program



Multi-Threading in Java

- Two techniques
  - Extends `java.lang.Thread`
  - Implements `java.lang.Runnable`

Extending java.lang.Thread

- Example of definition

```
public class MyThread extends Thread{
  public void run(){
  ...
  }
```

- Example of thread start

```
...
Thread myThread = new MyThread();
myThread.start();
...
```

Implementing java.lang.Runnable

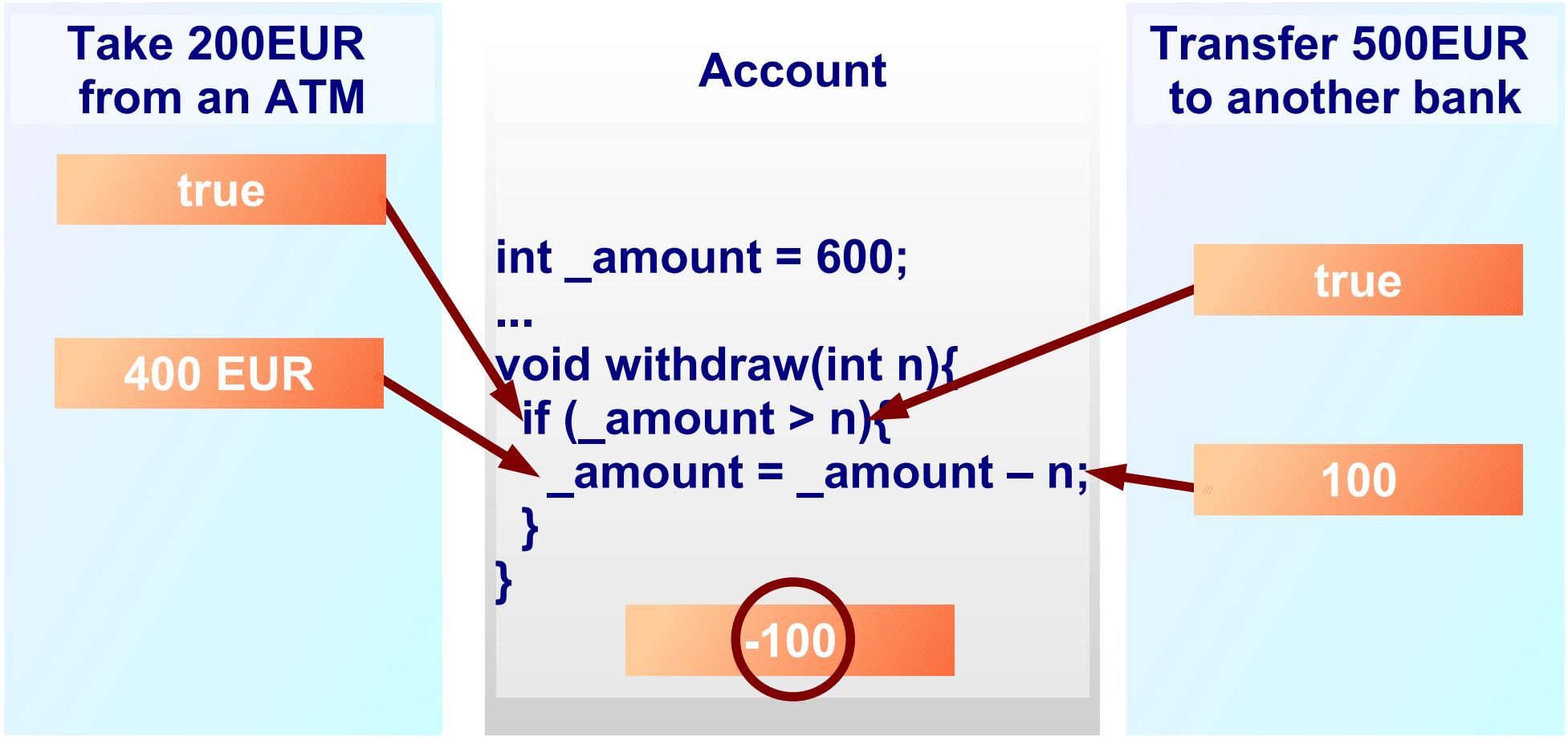
- Example of definition

```
public class MyRunnable implements Runnable{
  public void run(){
  ...
  }
```

- Example of thread start

```
  ...
  Runnable aRunnable = new MyRunnable();
  Thread myThread = new Thread(aRunnable);
  myThread.start();
  ...
```

Concurrent Accesses



The synchronized Keyword

- Locks
  - A method
  - A set of statements
  - A block
- Only one thread owns the lock
- One lock associated with one object

Synchronized Examples

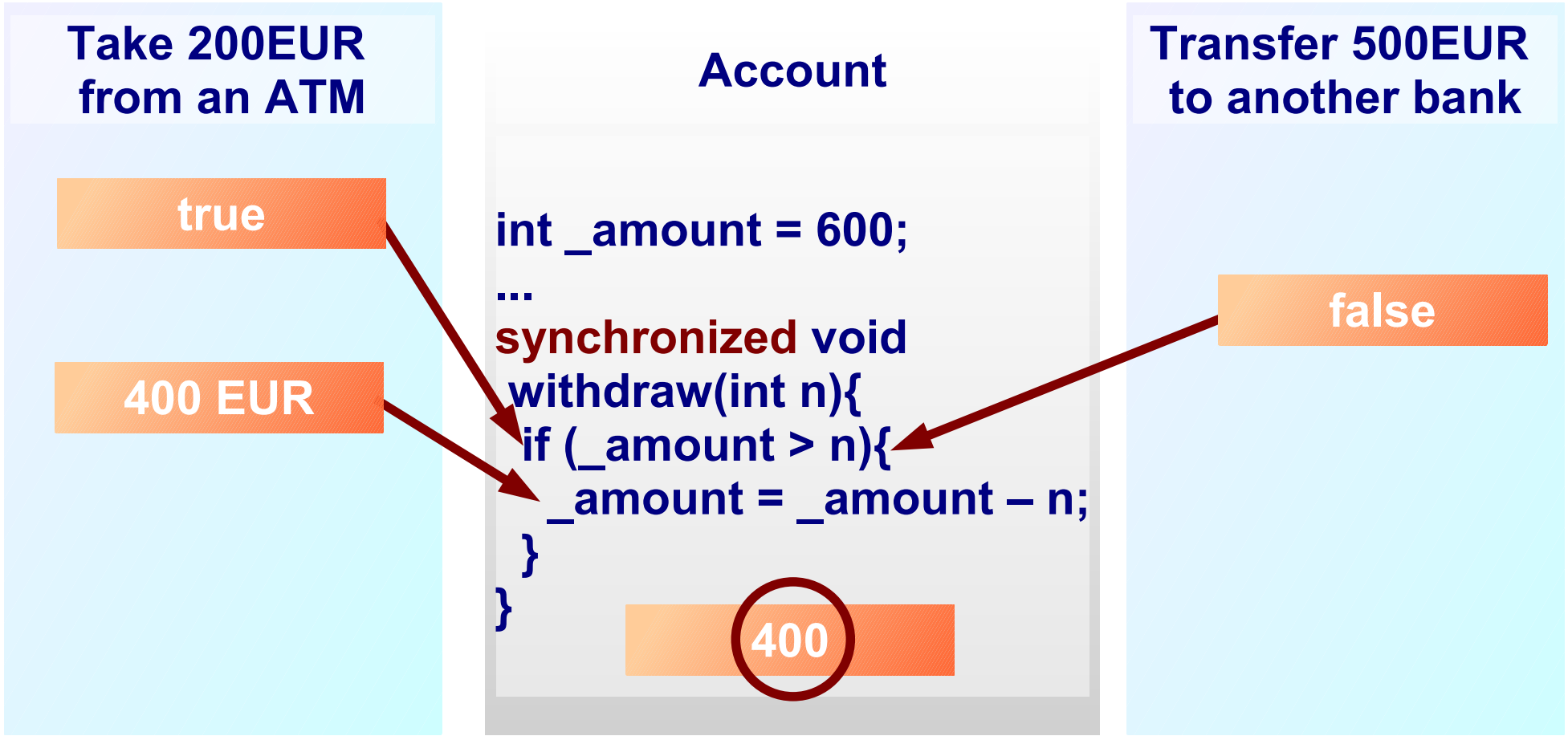
- Example of a synchronized method

```
public synchronized void withdraw(int n){
  ...
  }
```

- Example of synchronized statements

```
public void withdraw(int n){
    ...
    synchronized(aObject){
     ...
    }
  }
```

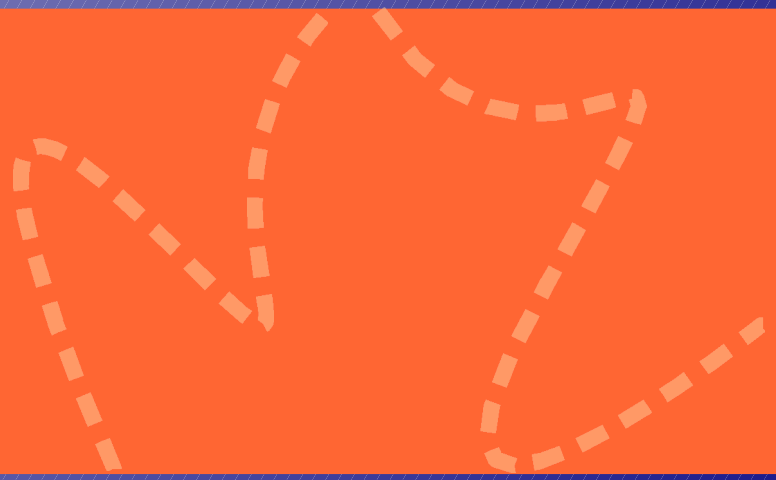
Synchronized Accesses



Thread Example

Slimming Animals

Conclusion



Advanced Programming

- Exceptions
  - More robust programs
- Multi-Threading
  - Better performance
  - More complex programs

See you next week