LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

Fakultät für Betriebswirtschaft
Munich School of Management

Basics in Programming for MMT

Session 6 – Passing Variables

MMT

www.mmt.bwl.lmu.de

Scope of the Session

1. Repetition

- Class

2. Theory

- Functions
- Pass by Value/Reference
- Objects
- Where are Variables valid?

3. Next

- Session 7

4. Project

Repetition

Repetition

Class (1/2)

- A `class` allows us to define data structures.
- The class (`Ball`) is the abstract description and the objects (`b`) are instances of that class.
- Each object contains its own set of variables defined in the class as fields.

```
Ball b;

void setup () {
  size(600,600);
  b = new Ball (235,237,52);
}

void draw () {...}

class Ball {
  float x;
  float y;
  float d;

  Ball (float x, float y, float d) {
    this.x = x;
    this.y = y;
    this.d = d;
    }
}
```

Repetition

Class (2/2)

Keyword `class` + class name

```
class Ball {
```

Fields

```
  float x;
  float y;
  float d;
```

Constructor: class name + arguments

```
  public Ball (int x, int y, int d) {
    this.x = x;
    this.y = y;
    this.d = d;
  }
```

Methods

```
  void move () {
    x = x+1;
    y = y+1;
  }
```

End of class

```
}
```

Theory

Theory

Functions (1/2)

- We already discussed the example of a function summing up two values.
- We can either pass values or variables to a function.
- Can we also pass variables that we can manipulate in functions?

```
int add (int a, int b) {
  return a+b;
}

void setup () {
  size(600,600);

  int x = add(10,20);
  println(x); // -> 30
}
```

Theory

Functions (2/2)

- If we try to pass a variable to a function and manipulate it inside, does it work?
- We can see in this example that `x` is unchanged …
- Why is that?

```
void add (int t, int a, int b) {
  t = a+b;
}

void setup () {
  size(600,600);

  int x = 0;
  add(x,10,20);
  println(x); // -> 0 ???
}
```

Theory

Pass by Value/Reference

- In Java simple datatypes (int) are passed as values.
- Arguments take values and variables but just pass the value to the inside of the function.
- If we try to change an argument in a function it does not affect the variable used as the argument.

```
void add (int t, int a, int b) {
  t = a+b;
}

void setup () {
  size(600,600);

  int x = 0;
  add(x,10,20); // -> x is passed as the value, changes do not affect x
  println(x);
}
```

Theory

Objects

- Objects are passed as references.
- Changes we perform on passed objects do affect the source/original version.
- We have to be careful if we want to change the source or not.

```
void move (Ball b) {
  b.x = b.x + 1;
}

void setup () {
  size(600,600);

  Ball b = new Ball (100,200,50);
  println(b.x); // -> 100
  move(b);
  println(b.x); // -> 101
}
```

Theory

Where are Variables valid?

- A variable we declare outside (`x`) of `setup` and `draw` can be used in our whole sketch.
- A variable that is declared inside a function (`y`) can just be used inside this function (`{ }`).
- A variable that is declared in the head of a loop (`i`) can just be used inside the loop (`{ }`).

```
int x;

void setup () {
  int y = 0;

  for (int i = 0; i<100; i = i+1) {
  }
}

void draw () {...}
```

Next

Next

Session 7

- Extends
- Implements

Project

Tutorial

1. Try to write a class for the two bats.
2. Write methods for the functionality: move, draw
3. How can we implement the border detection with methods?