Lecture Informatik III

Methods in Java

Differences to Eiffel

- **Overloading:** methods with *different* signatures can share the *same* name
- **Overriding:** methods of the *same* signature can *redefine* inherited methods
- **Class Methods:** a *static* method belongs to the class an not to an instance
- **Hiding:** class methods can *hide* instance methods with the *same* name

globis

Signatures in Java

- In Java the signature consists of
  - name of the method
  - sequence of parameter types
- Not part of the signature is
  - return type
  - parameter names
- **Note:** This implies that two methods with different return type are considered the same!

globis

What is going on?

```
public class Point {
  void setLocation(Point p) {...}
  void setLocation(int a, int b) {...}
}
```

Overloading


```
public class RealPoint extends Point {
  void setLocation(RealPoint p) {...}
  void setLocation(float a, float b) {...}
  void setLocation(int a, int b) {...}
}
```

Overloading

Overriding

globis

What is going on?

- How many methods **`setLocation`** are there in class **`RealPoint`**?
  - **`setLocation(Point)`**
  - **`setLocation(RealPoint)`**
  - **`setLocation(int, int)`**
  - **`setLocation(float, float)`**
- This set of methods is called the Visible Methods in class **`RealPoint`**.
- What happens during a call with two arguments?
- Will the float or the integer version be executed?

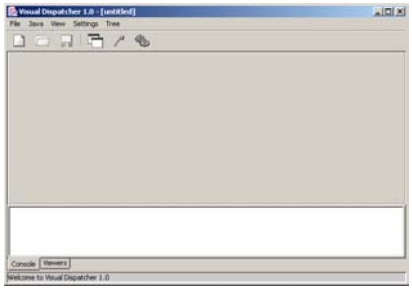
globis

Method Dispatching

- The algorithm to select the method to be executed is called *method dispatching*.
- In Java the algorithm consists of the following steps
  - Compute all Visible Methods
  - Compute all Applicable Methods
  - Compute the Most Specific Method
  - If unique, execute the Most Specific Method
- **Note:** It can happen that there is no most specific method!

globis

Demo


Visual Dispatcher 1.0 - [untitled]
File Java View Settings Tree
Console Viewers
Welcome to Visual Dispatcher 1.0

globis