OPI
Class Invariants

Carsten Schürmann

March 7, 2007

Three Legs

Representation

- Base types. `int`, `double`, `bool`
- Fields of objects.
- Class hierarchy models taxonomy. `Horse` $<$ `Animal`

Functionality

- Base commands. `for`, `<`, `new`, `int x`
- Methods.

Code organization

- Class hierarchy models software engineering process.
- Interfaces.

Example

```
class Ball {
        private int x,y;
        private String color;
        public Ball(String color){
                this.color = color;
                x=0; y=0;
        }
        public void move(int dx, dy){
                x+=dx;
                y+=dy;
        }
        public String toString(){
                return color +  ball at ( + x + , + y + );
        }
}
```

Discussion

Good

- Ease of object descriptions.
- Clear understanding of base types.
- Clear view of object behavior.

But

- What is the relationship between the real world objects that we would like to represent and the ones that represent them?

Process of compilation

Source

- .java file.
- Lexing, Parsing.

Abstract Syntax

- Syntactic analysis, type checking.
- Code generation.

Compiled code

- .class file.
- Run it, if it contains

```
static void main (String[] args) {}
```

Representation Invariants

Observation Consider `int i;`. Use it for

- All integer numbers $\ldots, -3, -2, -1, 0, 1, 2, 3 \ldots$.
- All positive numbers $1, 2, 3, \ldots$.
- All negative numbers $\ldots, 3, 2, 1$.
- All even numbers $\ldots, -4, -2, 0, 2, 4 \ldots$.

But

- What is the relationship between the real world objects that we would like to represent and the ones that represent them?