

SE 350 - Object-Oriented Software Development

Exam Review

Instructor: Stefan Mitsch

Programming Paradigms

```
public int f(int x, int y) {  
    if (x < y) return x;  
    else return f(x-y, y);  
}
```

What is the result of the functional program above?

- `f(5,2) == 1` ?
- `f(5,3) == 2` ?
- `f(3,1) == 0` ?
- `f(7,4) == 2` ?

Object-Oriented Programming

True or false?

- Classes are objects?
- Every class must have at least 1 field and 1 method?
- An object is an instance of a class?
- An object has no state?
- A class cannot have a name, only objects can have names?
- A class can extend itself?

Object-Oriented Programming

```
public class C {  
    private int i = 1;  
    public void f() { i = i+2; }  
    public void g() { i = 1; }  
}
```

How many times to call `f()` until `i > 6` ?

- `i` never becomes larger than `6` ?
- 1 time?
- 2 times?
- 3 times?
- 4 times?

How many times to call the sequence `f(); g();` until `i > 6` ?

Method Overloading

```
public class A {  
    private String f() { return "f()"; }  
    private String f(String s) { return "f(String)"; }  
    private String f(String s, String t) { return "f(String,String)"; }  
    private String f(String s, int t) { return "f(String,int)"; }  
}
```

True or false result?

- `f() == "f()"` ?
- `f("x") == "f(String)"` ?
- `f("1") == "f(String,int)"` ?
- `f("x",1) == "f(String,int)"` ?
- `f("x","1") == "f(String,int)"` ?

Inheritance

```
public class A {  
    public static String f() { return "A"; }  
    public String g() { return A.f(); }  
    public String h() { return f(); }  
}  
  
public class B extends A {  
    public static String f() { return "B"; }  
    public String g() { return "B"; }  
    public String h() { return f(); }  
}
```

What is the result when using the following reference `A x = new B();` :

- `x.f() == "A" ?`
- `x.f() == "B" ?`
- `x.g() == "A" ?`
- `x.g() == "B" ?`
- `x.h() == "A" ?`
- `x.h() == "B" ?`
- `A.f() == "A" ?`

Inheritance

```
public final class C {  
    public final String s = "C";  
    public static String t = "C";  
    public static int f() { return 1; }  
    public final int g() { return 2; }  
}
```

- Class `C` can be extended?
- The value of `s` can be changed?
- The value of `t` can be changed?
- The method `f` can be overridden?
- The method `g` can be overridden?

Inheritance

```
public class A {}  
public class B extends A {}
```

Which casts are safe, which ones always fail, and which ones must be checked?

- `A x1 = new A() ?`
- `B x2 = new B() ?`
- `A x3 = new B() ?`
- `B x4 = new A() ?`
- `B x5 = x3 ?`
- `x1 = x2 ?`
- `x2 = x3 ?`

Constructor Chaining

```
public class A {  
    public A() { this(0, 0); }  
    public A(int x) { this(x, 0); }  
    public A(int x, int y) {}  
}  
public class B extends A {  
    public B(int x) { super(x); }  
}  
public class C extends B {  
    public C() { super(1); }  
}
```

What is the constructor sequence when instantiating a `new C()` ?

Java Classes

```
public abstract final class a {  
    protected String empty() { return ""; }  
}
```

True or false?

- The class name violates Java coding conventions?
- The class compiles?
- The method `empty` must be abstract?
- The class declares no fields?

Java Instantiation

```
public class A {  
    public A(int a) {}  
    public A(String a) {}  
}  
  
public class B {}
```

Which of the following instantiation statements compile?

- `A a = new A() ?`
- `A a = new A(1) ?`
- `A a = new A("1") ?`
- `A a = new A(1.0) ?`
- `B b = new B() ?`
- `B b = new B(1) ?`

Visibility

```
public class A {  
    private String a = "a";  
    protected String b = "b";  
    public String c = "c";  
}  
public class B extends A {  
    public String getA() { return a; }  
    public String getB() { return b; }  
    private String getC() { return c; }  
}
```

True or false?

- Field `a` is only visible within class `A` ?
- Method `getA()` compiles
- Method `getC()` must be `public` because `c` is `public` ?
- Field `b` is not visible to subclasses of class `A` ?

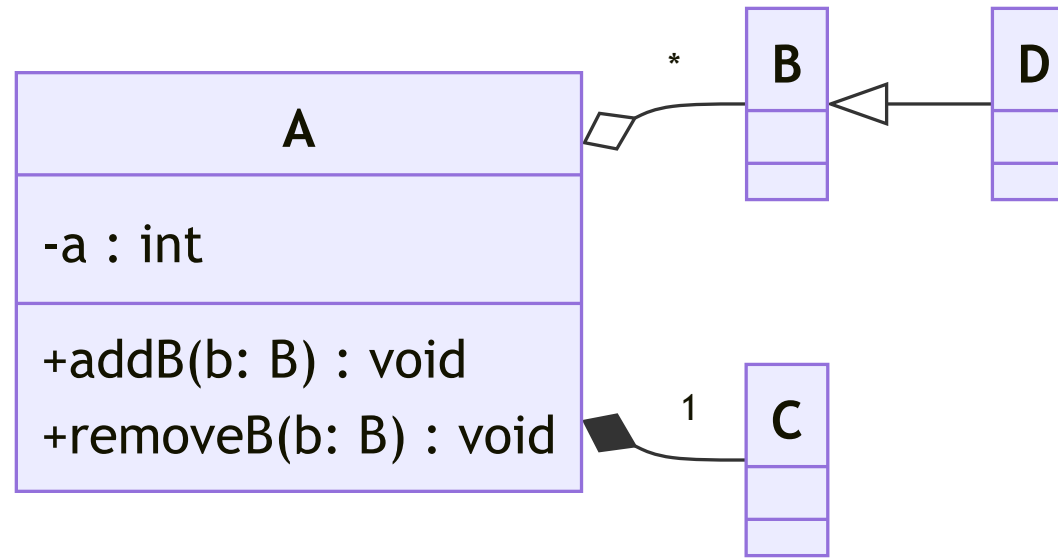
Test-Driven Development

- Which test passes?
- How to fix the code of the failing test?

```
public class A {  
    public String a = "A";  
    public A(String a) { this.a = a; }  
}
```

```
public class ATests {  
    @Test public void test1() { assertEquals("A", new A().a); }  
    @Test public void test2() { assertEquals("A", new A("a").a); }  
}
```

UML Class Diagrams



True or false?

- When an **A** is destroyed, the **C** remains alive by itself?
- A **D** is an **A** ?
- An **A** can have any number of **B** s?
- An **A** has a **public** field **a** ?
- **B** s cannot be added to or removed from their **A** ?