

## Übungen zu Einführung in die Informatik

Hinweis: Freiwillige Abgabe bis 21.01.08, 12.00.

### Aufgabe 12-1

### Defensives Programmieren

- Definieren Sie eine Klasse `NonPositiveIntException` extends `Exception` und modifizieren Sie die Methode `gcd` in der Klasse `Divisor` aus Übungsblatt 3 so, dass ein solches Ausnahmeobjekt geworfen wird, wenn die Parameter `x` und `y` nicht beide größer als 0 sind. Testen Sie die modifizierte Methode. Modifizieren Sie wenn notwendig auch die Klasse `Fraction` aus Übungsblatt 5, so dass diese fehlerfrei kompiliert werden kann.
- Definieren Sie eine Klasse `NoElephantException` und modifizieren Sie die Klasse `Hunting`, die Sie von unserer Web-Seite herunterladen können (Übungsblatt 3, Lösungsvorschlag), so dass die Methode `findAnElephant` ein solches Ausnahmeobjekte wirft, wenn sich kein Elephant in `africa` befindet.
- Laden Sie die Klassen `Game` und `Heap` von unserer Web-Seite herunter (Übungsblatt 7, Lösungsvorschlag). Führen Sie das Programm aus und finden Sie heraus, welche Exceptions vom der Zeile `int n = Integer.parseInt(args[0])` gleich am Anfang der `main`-Methode geworfen werden können. Machen Sie das Programm durch geeignete Exception-Behandlung robust.

### Aufgabe 12-2

### Exceptions

Gegeben seien die folgenden (teilweise fehlerhaften) Klassen:

```
public class E1 extends Exception {}
public class E2 extends E1 {}
public class E3 extends E1 {}
public class E4 extends Error {}
public class E5 extends RuntimeException {}
public class X
{
    public void oops() throws E1 {}
    public X swap() { return this; }
}
public class X1 extends X
{
    public void oops() throws E2
    {
        throw new E2();
    }
    public X swap()
    {
        return new X2();
    }
}
public class X2 extends X
{
    public void oops() throws E1
    {
        throw new E3();
    }
    public X swap()
    {
        return new X3();
    }
}
public class X3 extends X
{
    public void oops() {}
    public X swap() throws X1
    {
        return new X1();
    }
}

public class A
{
    private X xs = new X1();
    private X x = new X2();
    protected void swap()
    {
        x = x.swap();
    }
    protected void swaps()
    {
        xs = xs.swap();
    }
    public void f() throws E1
    {
        try
        {
            x.oops();
            xs.oops();
        }
        catch (E2 e)
        {
            swaps(); // (1)
        }
        finally
        {
            swap(); // (2)
        }
    }
    public static void main(String[] args)
    {
        A a = new A();
        for (int i = 0; i < 10; i++)
        {
            a.f();
        }
    }
}
```

- Welche Fehler enthalten die oben angegebenen Klassen?
- Welche Möglichkeiten gibt es, diese Fehler zu korrigieren? Geben Sie eine korrigierte Versionen der fehlerhaften Klassen an.
- Welche der Stellen (1) und (2) werden (in Ihrer korrigierten Version) beim Durchlaufen der `for`-Schleife mit den Werten `i = 0`, `i = 1`, `i = 2`, `i = 8` und `i = 9` ausgeführt?
- Welche der oben angegebenen Ausnahmen können im Rumpf einer Methode mit der Signatur `void f() throws E2` ausgelöst werden, ohne in der Methode selbst abgefangen zu werden?