Objektorientierte Programmierung

Studiengang Medieninformatik

Hans-Werner Lang

Hochschule Flensburg

Vorlesung 6
19.04.2017

Was bisher geschah...

Hochschule Flensburg
University of Applied Sciences

Objektorientierte Programmierung

- Klassen und Objekte, Attribute und Methoden
- Klassen ableiten, Vererbung von Attributen und Methoden
- Methoden überschreiben und überladen, Polymorphie
- Abstrakte Klasse, abstrakte Methoden, Interfaces
- Klassenattribute, Klassenkonstanten, Klassenmethoden

Grafische Benutzungsoberfläche (GUI)

- GUI-Elemente: Frames, Panels, Labels, Buttons, Textfelder
- Reaktion auf Ereignisse mit Listenern

Heute:

- Refactoring
- Testen

Hochschule Flensburg
University of Applied Sciences

Refactoring

Hochschule Flensburg
University of Applied Sciences

Wir haben inzwischen eine Menge Programmcode geschrieben. Zeit, einmal innezuhalten und unseren Programmcode eines zweiten Blickes zu würdigen.

Refactoring bedeutet, sich ein bereits funktionierendes Programm vorzunehmen und es hinsichtlich Struktur und Lesbarkeit zu verbessern.

"Wieso? Läuft doch!"

"If it works, don't fix it."

?

Hochschule Flensburg
University of Applied Sciences

Programme sollen sein...

einfach

knapp

lesbar

elegant

wohlstrukturiert

---

korrekt

Einfachheit, Knappheit

"Ich bin ein Berliner."

(John F. Kennedy)

"I have a dream."

(Martin Luther King)

"Der Ball ist rund."

(Sepp Herberger)

"Wir sind Papst."

(Bild-Zeitung)

Hochschule Flensburg
University of Applied Sciences

Lesbarkeit

Hochschule Flensburg
University of Applied Sciences

Software ist Literatur. Software wird von Menschen geschrieben und von Menschen gelesen.

"Für mich ist Literatur eine Form der Freude.
Wenn wir etwas mit Mühe lesen, so ist der Autor gescheitert."

Jorge Luis Borges (argentinischer Schriftsteller)

Lesbarkeit

Hochschule Flensburg
University of Applied Sciences

Software ist Literatur. Software wird von Menschen geschrieben und von Menschen gelesen.

"Für mich ist Software eine Form der Freude.
Wenn wir etwas mit Mühe lesen, so ist der Autor gescheitert."

Hans Werner Lang (deutscher Informatiker;-)

Lesbarkeit

Hochschule Flensburg
University of Applied Sciences

- Variablen, Funktionen, Klassen treffend benennen
- Programm übersichtlich formatieren
- Sinnvolle Kommentare hinzufügen

Variablen, Funktionen, Klassen treffend benennen

Hochschule Flensburg
University of Applied Sciences

Variablen: `i, p, pos, lastmove, startfields` (alle Buchstaben klein)

Funktionen: (Camel Case, erster Buchstabe klein)

Rückgabetyp	
`boolean`:	`isValid(), hasNext(), comesDigit()`
`void`:	`findMove(), setPosition()`
andere:	`getPosition(), evaluationOf(), bestMove()`

Klassen: `Position, Placement, CircleGrid`
(Camel Case, erster Buchstabe groß)

Konstanten: `G, MAX_SIZE, ALPHANUMERIC` (alle Buchstaben groß)

Eclipse unterstützt Umbenennung: Menü `Refactor | Rename`

Variablen geeignet benennen

Hochschule Flensburg
University of Applied Sciences

Lokale Variablen und Parameter: kurz

```
if (array_to_be_sorted[currentIndex1-1]>array_to_be_sorted[currentIndex1])
{
    int temporaryValue=array_to_be_sorted[currentIndex1-1];
    array_to_be_sorted[currentIndex1-1]=array_to_be_sorted[currentIndex1];
    array_to_be_sorted[currentIndex1]=temporaryValue;
}
```

Besser so:

```
if (a[i-1]>a[i])
{
    int t=a[i-1];
    a[i-1]=a[i];
    a[i]=t;
}
```

Globale Variablen, Attribute in einer Klasse: länger

`lastmove, startfields usw.`

Übersichtliche Formatierung

Hochschule Flensburg
University of Applied Sciences

Nicht so:

```
while (b!=0) {
    if (a>b) {
        a=a-b;
    } else {
        b=b-a;
    }
}
```

Sondern so:

```
while (b!=0)
{
    if (a>b)
    {
        a=a-b;
    }
    else
    {
        b=b-a;
    }
}
```

Oder so:

```
while (b!=0)
    if (a>b)
        a=a-b;
    else
        b=b-a;
```

Zugehörige öffnende und schließende geschweifte Klammern übereinander;

Einrückung 4 Zeichen

Eclipse: Formatierungsregeln importieren:
`http://www.inf.fh-flensburg.de/lang/formatjava.xml`

Sinnvolle Kommentare (1)

Hochschule Flensburg
University of Applied Sciences

Vor Klassendefinitionen:

```
/**
 * repräsentiert einen Bruch bestehend aus Zähler und Nenner
 * (ganzzahlig); stellt Methoden der Bruchrechnung bereit
 */
public class Bruch
{
 ...
```

Der Kommentar ist hier als Javadoc-Kommentar ausgeführt. Mithilfe des Javadoc-Tools lassen sich automatisch HTML-Dokumentationsdateien erzeugen.

Sinnvolle Kommentare (2)

Hochschule Flensburg
University of Applied Sciences

Vor Methodendefinitionen:

```
// Division: Multiplikation von this mit dem Kehrwert von other
public Bruch div(Bruch other)
{
    return this.mul(other.rec());
}
```

Parameter zitieren, hier: *other*.
Besser als nur: `// Multiplikation mit dem Kehrwert`

Rückgabewert zitieren:

```
// gibt true zurück, wenn this < other ist
public boolean isLess(Bruch other)
{
...
```

Sinnvolle Kommentare (3)

Hochschule Flensburg
University of Applied Sciences

Vor Codeabschnitten:

```
// Vergleichselement x wählen
int m=lo+(hi-lo)/2;
int x=a[m];
```

Hinter Anweisungen:

```
setVisible(true);   // als letzte Anweisung des Konstruktors
```

Überflüssige Kommentare vermeiden:

```
i=i+1;      // i um 1 erhöhen
```

Eleganz

```
// sortiert ein Teilstück des Arrays a
// vom Index lo bis zum Index hi
private void quicksort (int lo, int hi)
{
    int i=lo, j=hi;

    // Vergleichselement x wählen
    int m=lo+(hi-lo)/2;
    int x=a[m];

    //  Aufteilung
    while (i<=j)
    {
        while (a[i]<x) i++;
        while (a[j]>x) j--;
        if (i<=j)
            exchange(i++, j--);
    }

    // Rekursion
    if (lo<j) quicksort(lo, j);
    if (i<hi) quicksort(i, hi);
}
```

Hochschule Flensburg
University of Applied Sciences

Struktur

Hochschule Flensburg
University of Applied Sciences

Lagern Sie zusammengehörige, kompliziertere Codeabschnitte in Funktionen aus.

Alt:

```
// Vergleichselement x wählen
int m=lo+(hi-lo)/2;
int x=a[m];
```

Neu:

```
// Vergleichselement x wählen
int x=getPivotElement(lo, hi);
```

Anderes Beispiel:

Alt:

```
if (z>=0 && z<m && s>=0 && s<n)
    return new Position(z, s);
```

Neu:

```
if (isValidPosition(z, s))
    return new Position(z, s);
```

Literatur

Webseite http://www.inf.fh-flensburg.de/lang/se/lesbarkeit.htm

R.C. Martin: *Clean Code*. Prentice Hall (2009)

M. Fowler: *Refactoring*. Addison-Wesley (1999)

Hochschule Flensburg
University of Applied Sciences

Testen

Hochschule Flensburg
University of Applied Sciences

"Testing can only show the presence of errors, not their absence."
(E.W. Dijkstra)

Aber:

Durch systematisches Testen lässt sich das Vorhandensein von Fehlern minimieren.

Und:

Automatisierte Tests sind wesentlich für ein fortwährendes Refactoring.

Testen

- Komponententest (Unit-Test)
- Integrationstest
- Systemtest
- Akzeptanztest

Hochschule
Flensburg
University of
Applied Sciences

Testgesteuerte Entwicklung (test-driven development)

Hochschule Flensburg
University of Applied Sciences

- zuerst wird der Test geschrieben
- dann das Programm

Beispiel Unit-Test:

```
import junit.framework.TestCase;

public class TestBruch extends TestCase
{
    public static void testComparisons()
    {
        assertTrue(new Bruch(1, 2).isLess(new Bruch(1, 1)));     //  1/2 < 1
        assertTrue(new Bruch(1, 3).isGreater(Bruch.ZERO));       //  1/3 > 0
        assertTrue(new Bruch(-2, 3).isLess(new Bruch(1, -2)));   // -2/3 < -1/2
        ...
```

Variante des *Pair Programming*:

- einer schreibt den Test (möglichst fies)
- der andere schreibt das Programm

und dann wird geschaut, wer gewinnt.

Pair Programming ist eine Arbeitsmethode in der sogenannten agilen Softwareentwicklung: Zwei Programmierer/innen arbeiten zusammen an demselben Programmstück (wie Sie teilweise im Labor).

Zusammenfassung

Refactoring

- Lesbarkeit
- Struktur

Testen

- Unit-Test

Hochschule Flensburg
University of Applied Sciences

Beim nächsten Mal:

- Datenstrukturen: *ArrayList* und andere
- Typ-Parameter
- Iteratoren