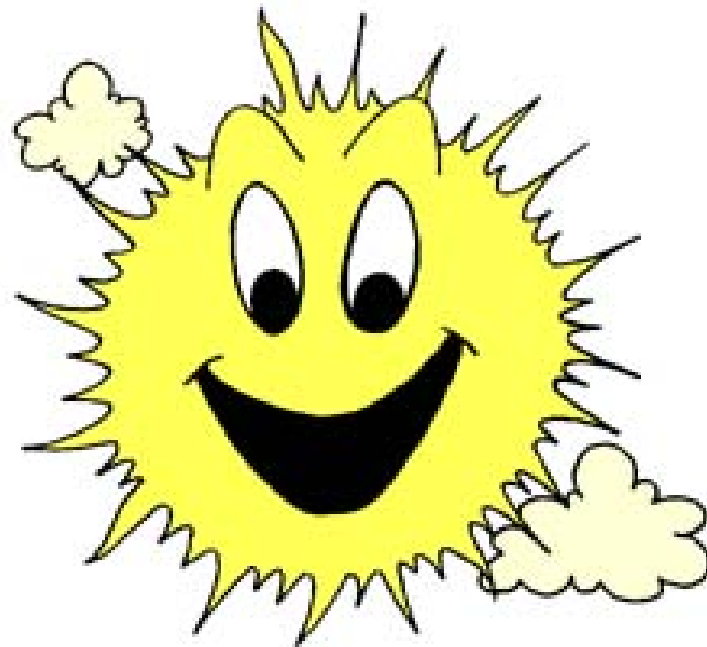


Organisation: Praktikumsbelegung

- Mi5x (B) 3
- Mi5y (C) 16
- Mi6x (B) 5
- Mi6y (B) 10
- Do3x (B) 10
- Do4x (C) 8
- Do6x (C) 16

Wer von Mi5y (C), Do6x (C) möchte in eine weniger belegte Gruppe wechseln?

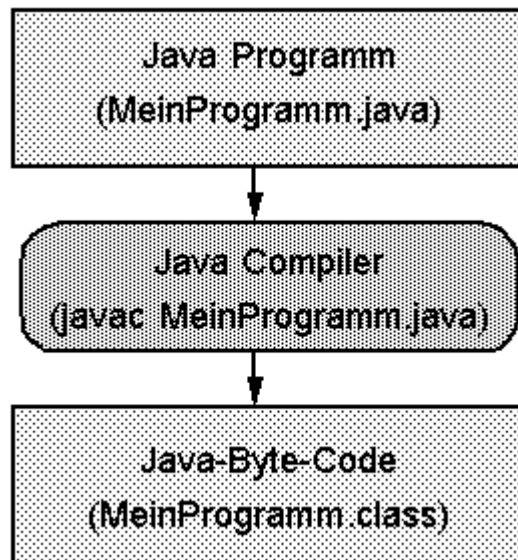
Wiederholung 1. Vorlesung



1.3.1.4 Hausaufgabe – Vorbereitung n. Vorl.

- Machen Sie sich mit Java-Dokumentation und Literatur vertraut.
- Installieren Sie ein JDK
- Schreiben Sie einfache Programme auf Basis von Beispielen (Applets sind nicht so wichtig)
- Ziel:
 - Programmstruktur
 - Datentypen String, int, double, boolean, ...
 - Klassen und Vererbung
 - Struktur der Bibliotheken kennen lernen und verstehen.

1.3.1.4 unser erstes Programm ...



ausführen

```
public class Hello
{
    public static void main( string[] argv )
    {
        system.out.println( "HELLO" );
    }
}
```

```
C:\WINDOWS>cd C:\1_Hello
C:\1_Hello>javac Hello.java
C:\1_Hello>
```

```
C:\1_Hello>javac Hello.java
C:\1_Hello>java Hello
HELLO
C:\1_Hello>
```

1.3.1.4 Java

- Quelldatei muss wie jeweilige Klasse heißen
- Eine Datei pro Klasse
- Keine Trennung von Definition und Deklaration wie z.B. in C++
- Mindestens eine Klasse muss eine main-Funktion haben:
`public static void main(String[] argv)`
`{ ... }`
- Befehl zum **Compilieren**:
> `javac MeineKlasse.java`
- Befehl zum **Ausführen** (Aufruf der VM):
> `java MeineKlasse`

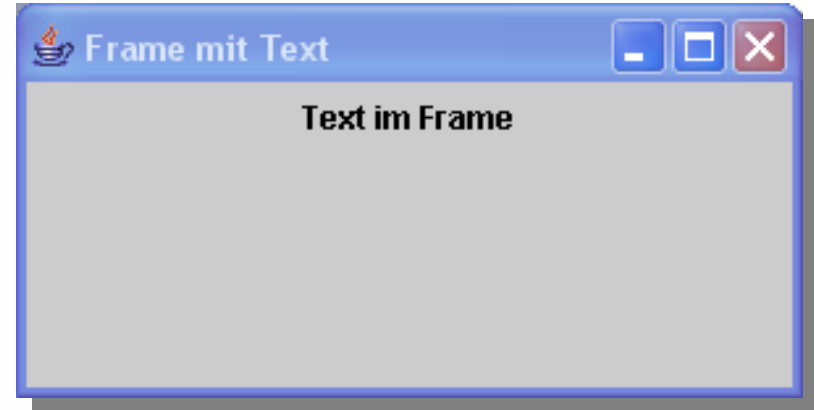
1.3.1.4 unser zweites Programm ...

```
import java.awt.*;
import javax.swing.*;

public class Hello2 extends JFrame
{
    Container c;
    JLabel beschrift;

    public Hello2()
    {
        c = getContentPane();
        c.setLayout( new FlowLayout() );
        beschrift = new JLabel( "Text im Frame" );
        c.add( beschrift );
    }

    public static void main( String[] argv )
    {
        Hello2 fenster = new Hello2();
        fenster.setTitle( "Frame mit Text" );
        fenster.setSize( 300, 150 );
        fenster.setVisible( true );
        fenster.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    }
}
```



1.3.1.4 ... ein Applet ...

```
import java.awt.*;
import javax.swing.*;

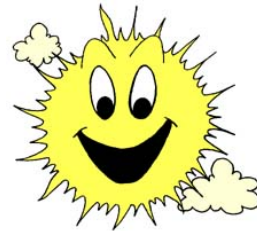
public class HelloApplet extends JApplet
{
    Container c;
    JLabel beschrift;

    public void init()
    {
        c = getContentPane();
        c.setLayout( new FlowLayout() );
        beschrift = new JLabel( "Text im
        c.add( beschrift );
    }
}
```

```
<html>
  <head>
    <title>Applet mit Text</title>
  </head>
  <body>
    <b>Ein Applet mit Text</b>
    <hr></hr>
    <applet code = "HelloApplet.class" width="300" height="150">
      Hier sollte eigentlich ein Applet auftauchen und laufen
    </applet>
  </body>
</html>
```



Ende der Wiederholung



1.3.1.4 Sicherheit bei Applets

- Applets sind meist unbekannter Code
- Bei Ausführung von Applets müssen Sicherheitsbeschränkungen eingehalten werden
- Ein Applet darf
 - nicht auf das lokale Dateisystem zugreifen - es sei denn der Benutzer läßt es ausdrücklich zu
 - keine Verzeichnisse ansehen, Existenz von Dateien prüfen
 - keine Dateiattribute (Größe, Zugriffsrechte) prüfen
 - keine Dateien lesen, schreiben, löschen, umbenennen
 - keine Netzwerkverbindungen zu irgendeinem Computer aufbauen
 - die Ausführung nicht über `exit()` beenden

1.3.1.5 Dokumentation von Java-Code

- Code muss dokumentiert werden!
- Mit javadoc wird HTML-Dokumentation erstellt!

Beispiel:

```
/**
 * Beispiel Hello World
 * @author P. R. O'Grammer
 * @version 17.10.2005
 */
public class Hello {

    /** Test Methode
     * @param in int
     * @return int
     */
    public int test(int in) { return in + 42; }
}
```

1.3.1.5 Dokumentation der Java-API

- Online Dokumentation zu Java
- Auch mit javadoc erstellt!
- Am besten Link auf Desktop erstellen!
[Java-Dokumentation](#) allgemein
- Spezielle die
[Java-API-Dokumentation](#)

Organisation: Fragen zum ersten Praktikum



Häufig gestellte Frage: Parameter in Java

```
public class Test
{
    public static void main( String[] argv ) {
        int argc = argv.length; // Größe eines Feldes (array)
        if ( argc < 1 ) // Prüfen, ob Parameter übergeben wurden
            System.out.println( "Keine Parameter übergeben" );
        else
            System.out.println( argc + " Parameter übergeben: " );

        for ( int i = 0; i < argc; i++ ) { // Parameter anzeigen
            System.out.println( (i+1) + ": " + argv[i] );
        }
        System.out.println(); // Leerzeile
        // Versuchen, Parameter in int zu parsen
        for ( int i = 0; i < argc; i++ ) {
            try { // try-Block
                int zahl = Integer.parseInt( argv[i] );
                System.out.println( "Argument " + (i+1) + ": int = " + zahl );
            }
            catch ( NumberFormatException e ) { // "fängt" Exception auf
                System.out.println( "Fehler: " + e.getMessage() );
            }
        } // end for
    } // end main
} // end class
```

1.3.2 Arten von Applikationen

■ Unterhaltungssoftware

- Spiele
- Funware
- Heim-Software

*Fließender
Übergang*

■ Professionelle Systeme (mission critical)

- Kommerzielle Datenverarbeitung
 - Bank, Börse, ...
 - Flugsicherung, Verkehrsleitsysteme, ...
- Office-Anwendungen
- Web-Anwendungen (Portale, Services, Shops, ...)
- Maschinensteuerung

*Niveau
Praktikum*

1.3.3 Schnittstelle Mensch – Maschine



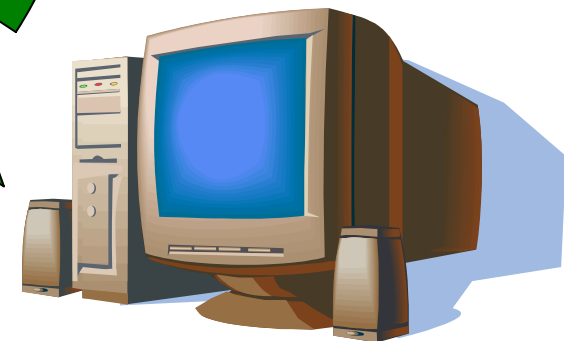
kreativ, intuitiv



Präsentation,
Visualisierung

Beobachten,
Bedienen

monoton, schnell



1.3.3 Entwickler und Anwender ... (I)

... spielen einen neuen Dialog durch:

Entwickler: ... So, jetzt hast Du einen Teil der Aufgabe bereits ausgeführt. Willst Du nun noch etwas verändern?

Anwender: Ja, ich möchte eine Komponente löschen. Was muss ich tun?

E: Deute auf das Menu mit dem Namen 'CD'.

A: CD ?

E: CD bedeutet 'Component Delete'.

A: OK, ich habe verstanden ... Oh, was passiert jetzt?

1.3.3 Entwickler und Anwender ... (II)

E: Du bist jetzt im Analyse-Modus.

Du musst aus Versehen 'AM' ausgewählt haben.

A: Ach so, ich dachte, ich hätte 'CD' ausgewählt.

Wie verlasse ich den Analyse-Modus?

E: Kein Problem. Gib einfach 'control Q' ein.

A: [tippt C O N T R O L ...]

E: Nein! Drücke die 'Strg'-Taste und die 'Q'-Taste.

A: Wie dumm von mir. ... Ich versuch's noch einmal.

... Es passiert gar nichts. Was habe ich falsch gemacht?

1.3.3 Entwickler und Anwender ... (III)

E: Du hast nichts falsch gemacht. Die Komponente ist gelöscht aber noch nicht vom Bildschirm entfernt. Gib noch 'control-J' ein.

A: Ich habe verstanden. Wenn ich die nächste Komponente lösche, gebe ich wieder 'control-Q' ein.

... Hoppla, warum ist der Bildschirm jetzt weiß?

E: Du hast nur 'Q' eingegeben. 'Q' bedeutet Quit und beendet das Programm. Jetzt musst Du leider alles noch einmal eingeben!

A: Wenn das so ist, machen wir nächste Woche weiter.

1.3.3 Entwickler und Anwender ... (IV)

Folgende Fehler wurden gemacht:

- Kommandos (CD, AM) sind nicht intuitiv
 - ➔ erzeugt unnötige Belastung für den Anwender
- Auf welches Objekt wird das Kommando angewendet?
- Keine Statusinformation zu Zustandsänderungen
- Kryptische Kommandos (control-J, control-Q) sind ungeeignet für unerfahrene, gelegentliche Benutzer
- Kein Feedback bei Delete
 - ➔ erzeugt Frustration und Ablehnung beim Anwender
 - ➔ ergibt Negativausstrahlung auf das Gesamtsystem