

## Übung 4

13. November 2017

---

### Aufgabe 4.1: *Benutzung der Kommandozeile*

In dieser Aufgabe sollen Sie den Umgang mit der Kommandozeile lernen. Laden Sie hierfür zunächst das BO-Logo von der Moodle Webseite auf [https://idp.hs-bochum.de/idp/images/logo\\_hs\\_bochum.gif](https://idp.hs-bochum.de/idp/images/logo_hs_bochum.gif) auf Ihren Desktop. Öffnen Sie eine Kommandozeile und führen folgende Aktionen durch:

1. Navigieren Sie in der Kommandozeile auf den Desktop.
2. Legen Sie auf dem Desktop einen Ordner mit der Bezeichnung „GINF“ an.
3. Verschieben Sie das BO Logo in den GINF Ordner.
4. Öffnen Sie das BO Logo mit dem Grafikprogramm mspaint.exe (oder einer vergleichbaren Anwendung auf Ihrem Betriebssystem).

### Aufgabe 4.2: *Einstieg in die Programmierung*

Machen Sie sich mit dem Umgang der Programmierwerkzeuge für die Programmierung mit *Java* vertraut, indem Sie das folgende Programm erstellen. Verdeutlichen Sie sich die 3 erforderlichen Schritte, die zum Erstellen eines Programmes nötig sind.

1. Schritt: Eingabe des Quelltextes in eine Datei mit einem Editor.
2. Schritt: Übersetzen des Quelltextes mit einem Compiler.
3. Schritt: Starten des Programms

Führen Sie nun jeden dieser 3 Schritte durch. Hierzu ist folgende weitere Information gegeben:

#### 1. Schritt:

Öffnen Sie einen Editor auf dem Computer und geben Sie folgenden Quelltext ein. Als Editor kann zum Beispiel *Notepad* verwendet werden

```
import java.util.*;

class Versuch1{

    public static void main(String [] args){
        String [][] bild={{"#####", "####  ####", "#####"},
            {"  #####", "#####", "#####  "},
            {"  #####", "####  ####", "#####  "},
            {"  #####  ", "#####", "  #####  "},
            {"  #####  ", "####  ####", "  #####  "},
            {"  #####  ", "  #####  ", "  #####  "}};

        int zahl=new Random().nextInt(6);

        System.out.println("+-----+");
        for(int i=0;i<bild[0].length;i++){
            System.out.println("|"+bild[zahl][i]+"|");
            if(i<bild[0].length-1){
                System.out.println("|#####|");
            }
        }
        System.out.println("+-----+");
    }
}
```

Anschließend speichern Sie den Quelltext unter dem Namen der Klasse ab. Denken Sie daran, dass *Java* zwischen Groß- und Kleinbuchstaben unterscheidet. Der Name der Datei muss exakt den Namen der Klasse („class“) in der Programmquelle haben. Die Dateierweiterung „.java“ wird immer kleingeschrieben.

## 2. Schritt:

Wechseln Sie in die Eingabeaufforderung und Sie können jetzt durch Starten des Compilers Ihr Programm übersetzen. Das Starten mit dem Compiler geschieht durch Eingabe des Kommandos `javac`. Der Compiler erzeugt nun eine Datei mit dem Namen „Versuch1.class“.

```
C:\Daten > javac Versuch1.java
```

Falls Sie eine Fehlermeldung bekommen, könnte dies daran liegen, dass in der Kommandozeile entweder ihr Quelltext oder der Compiler nicht gefunden wird.

### 3. Schritt:

Wenn das Programm ohne Fehlermeldungen des Compilers übersetzt wurde, können Sie Ihre Anwendung durch den Aufruf *java* starten.

```
C:\Daten> java Versuch1
```

Die *JavaVM* (Java virtuelle Maschine) liest jetzt die vom Compiler erzeugte Klassen-datei. Die Dateierweiterung „.class“ darf hierbei nicht angegeben werden. Ihr Programm wird ausgeführt und auf dem Bildschirm sollte ein Text erscheinen. Sollten Fehlermel-dungen auf dem Bildschirm auftauchen, überprüfen Sie Ihren Quelltext auf Fehler und korrigieren diese. Führen Sie das Programm mehrfach aus.

- Führen Sie das Programm mehrfach aus. Verändert sich die Ausgabe?
- Das Programm soll als eine Art Zufallsgenerator fungieren. Was soll das Programm simulieren? (Tipp: Kennen Sie einen „Zufallsgenerator“ den man im alltäglichen Leben benutzt?)

### Aufgabe 4.3: Einstieg in die Programmierung mit Eclipse

Grundsätzlich ist es möglich mit Texteditor und Kommandozeilencompiler jede Art von Java-Applikation zu erstellen. Allerdings ist diese Form der Softwareentwicklung selbst bei kleinen Projekten umständlich und fehleranfällig, da der Programmierer beispielswei-se keine Unterstützung bei der Verwaltung von Quellcode oder der Suche nach Fehlern erhält.

Daher findet die professionelle Softwareentwicklung praktisch nur noch in sogenannten IDEs (Integrated Development Environment) statt. Dabei handelt es sich um umfang-reiche Programme die praktisch alle für die Programmierung notwendigen und hilfrei-chen Tools wie zum Beispiel Editor, Compiler, Debugger, Laufzeitumgebung (JavaVM), Quellcodeverwaltung, etc. enthalten.

Im Java-Umfeld erfreut sich die Entwicklungsumgebung *Eclipse* großer Beliebtheit. *Eclip-se* ist opensource und daher auch für Sie frei verfügbar. *Eclipse* besitzt einen modularen Aufbau und kann so für die verschiedensten Arten der Softwareentwicklung, von We-bapplikationen bis zu eingebetteten System z.B. im Auto, genutzt werden. Wir verwen-den hier Eclipse in der Standard-Java-Konfiguration.

- Starten Sie nun Eclipse in dem Sie auf das Icon auf dem Desktop klicken.

**Hilfestellung:** Als erstes müssen wir jetzt einmalig unseren Workspace anlegen. Dies ist nach einer Neuinstallation oder bei einem neuen Nutzer notwendig. Gehen Sie

hierzu auf *File/Switch Workspace/Other* geben Sie im daraufhin geöffneten Dialog den Workspace: *d : \workspace\_nachname* an. In diesem Workspace können Sie während des Semesters alle Ihre Übungsaufgaben bearbeiten und Ihre gesamten Quellcodes abspeichern. Sie Sollten daher zum Ende einer Stunde den Workspace auf einen USB-Stick sichern.

Jetzt kann die eigentliche Softwareentwicklung mit Eclipse beginnen. Die nachfolgende Abbildung gibt eine Übersicht über die für Sie wesentlichen Funktionen.

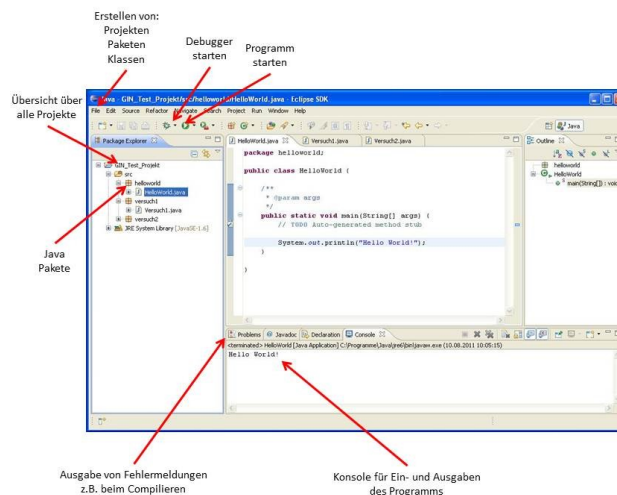


Abbildung 1: Arbeitsbereich eines mobilen Roboters als Grid

Über das Menu *File/new* können Sie jede Art von Komponente erzeugen. Von besonderem Interesse für Sie sind Projekte, Pakete und Klassen.

Erzeugen Sie zunächst ein neues Projekt mit dem Namen *Uebung* in dem Sie sämtliche Übungsaufgaben dieses Semesters bearbeiten werden. Das neue Projekt erscheint dann im Package Explorer auf der linken Seite.

Als nächstes legen Sie ein neues Paket für die aktuelle Aufgabe mit dem Namen *ErstesProgramm* in diesem Projekt an.

Jetzt erzeugen innerhalb des Pakets eine neue Klasse mit dem Namen *MeinErstesProgramm*

Geben Sie über den Texteditor den folgenden Quellcode für diese Klasse ein:

```
import java.util.*;

class MeinErstesProgramm{
```

```

public static void main(String [] args){
    String [][] bild={"#####", "####  ####", "#####"},
        {"  #####", "#####", "#####  "},
        {"  #####", "####  ####", "#####  "},
        {"  #####  ", "#####", "  #####  "},
        {"  #####  ", "####  ####", "  #####  "},
        {"  #####  ", "  #####  ", "  #####  "}};

    int zahl=new Random().nextInt(6);

    System.out.println("+-----+");
    for(int i=0;i<bild[0].length;i++){
        System.out.println("|"+bild[zahl][i]+"|");
        if(i<bild[0].length-1){
            System.out.println("|#####|");
        }
    }
    System.out.println("+-----+");
}
}

```

Die Klasse muss jetzt nicht mehr explizit kompiliert werden, dies geschieht automatisch beim Speichern der Datei. Sollte es dabei zu Problemen gekommen sein, werden die Fehlermeldungen im unteren Fenster unter dem Reiter Problems angezeigt.

War die Compilierung erfolgreich kann das Programm gestartet werden. Auch das geht aus der IDE heraus durch einen Klick auf den Run-Button. Die Ein- und Ausgabe erfolgt ebenfalls im unteren Fenster unter dem Reiter Console.