

Übungen zur Vorlesung "Wissenschaftliches Rechnen I"

Nicolas Gauger, René Lamour

Wintersemester 2006/2007

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
Achtung: Passwort **niemals** weitergeben!
- ▶ Das Dateisystem ist speziell strukturiert

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Passwort* einloggen.
Achtung: Passwort **niemals** weitergeben!
Änderung des Passwortes über das Internet:
Browser (firefox) über Menü öffnen:
Mathe-Homepage → Computerpool
- ▶ Das Dateisystem ist speziell strukturiert

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert
Filemanager öffnen (Blaues Häuschen):
Homeverzeichnis → Wurzel des Filesystem ansehen.

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert und durch eine Kombination von Nutzern, Systemnutzern und Zugriffsrechten geschützt.

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert
und durch eine Kombination von Nutzern, Systemnutzern und
Zugriffsrechten geschützt.
Terminal öffnen (Bildschirmsymbol): Befehl: `ps aux`

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert und durch eine Kombination von Nutzern, Systemnutzern und Zugriffsrechten geschützt.
Filemanager: Verzeichnis (directory) aussuchen,
rechte Maustaste → Eigenschaften → Berechtigungen → erweiterte Berechtigungen

Linux

- ▶ Linux ist ein *OpenSource*-Betriebssystem
- ▶ Es erlaubt Mehrbenutzerbetrieb.
Daher muss man sich mit seinem *login*-Namen und dem *Password* einloggen.
- ▶ Das Dateisystem ist speziell strukturiert und durch eine Kombination von Nutzern, Systemnutzern und Zugriffsrechten geschützt.

man findet auch: $d \underbrace{rwx}_{u(\text{ser})} \underbrace{rwx}_{g(\text{roup})} \underbrace{rwx}_{o(\text{thers})}$

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
- ▶ Virtuelle Arbeitsflächen einrichten:

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
Hintergrundbilder, Buttons, virtuelle Arbeitsflächen
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
- ▶ Virtuelle Arbeitsflächen einrichten:

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
Maus auf dem Desktop, rechte Maustaste → Neu erstellen →
Verknüpfung zu Programm
- ▶ Virtuelle Arbeitsflächen einrichten:

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
Maus auf dem Desktop, rechte Maustaste → Neu erstellen →
Verknüpfung zu Programm
Icon auswählen, Bezeichnung festlegen, Befehl eintragen
- ▶ Virtuelle Arbeitsflächen einrichten:

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
- ▶ Virtuelle Arbeitsflächen einrichten:

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
- ▶ Virtuelle Arbeitsflächen einrichten:
Maus auf virtuellen Arbeitsflächenanzeiger, rechte Maustaste
→ Arbeitsflächen einrichten

Graphische Oberfläche

- ▶ Richten Sie sich Ihre Oberfläche nach Ihren Wünschen ein!
- ▶ Button (Knopf) zum Starten eines Programmes einrichten:
- ▶ Virtuelle Arbeitsflächen einrichten:
Maus auf virtuellen Arbeitsflächenanzeiger, rechte Maustaste
→ Arbeitsflächen einrichten
empfohlene Anzahl: 4, Namen anzeigen und keine Nummern

Linux-Befehle

Befehlsabarbeitung im Terminal (auch Konsole, Shell) über die Tastatur (keyboard) erlaubt ein schnelles, konzentriertes Arbeiten.

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:

`/home/<login-name>` oder gleichwertig `/u/<login-name>`

Der Befehl

`pwd`

print name of current/working directory

- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen **name**, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen **name**, so wechselt
`cd name` dorthin.
`cd ..` geht eine Stufe zurück (aus dem Verzeichnis heraus)
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen **name**, so wechselt
`cd name` dorthin.
`cd ..` geht eine Stufe zurück (aus dem Verzeichnis heraus)
`cd` wechselt in das eigene Homeverzeichnis
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
`ls -la` zeigt detaillierte Informationen incl. Dateirechten.
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶ `man <befehl>` zeigt das Manual (man-page) dieses Befehls. Ende durch Eingabe von `q`.

Linux-Befehle

- ▶ Jedes neu geöffnete Terminal startet in Ihrem Homeverzeichnis:
`/home/<login-name>` oder gleichwertig `/u/<login-name>`
- ▶ Haben Sie ein Verzeichnis mit dem Namen `name`, so wechselt `cd name` dorthin.
- ▶ `ls` zeigt den (sichtbaren) Inhalt des Verzeichnisses an
- ▶ `kate` öffnet den Editor `kate`
- ▶ Eingabe von `man:<befehl>` in der Adresszeile des `Filemanagers` gibt eine Beschreibung, nur `man:` erzeugt eine Übersicht über viele Befehle.

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ **Der Tabulator dient auch der Textergänzung.**
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
Beispiel Eingabe: `mog` und Tabulatortaste
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
Beispiel Eingabe: `mog` und Tabulatortaste
`mogrify` ist ein sehr nützlicher Befehl zur
Formatumwandlung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
Textergänzung beim Einstellen eines Pfades:
`cd pu` und Tabulatortaste
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ **Copy and paste:**
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste: Öffnen Sie eine zweite Shell, markieren Sie einen Text durch Überstreichen mit der Maus mit gedrückter linker Taste, wechseln Sie ins zweite Terminal und drücken Sie die Mitteltaste der Maus.
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ **Wahl der Namen von Dateien und Verzeichnissen**
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit
angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
 - Groß- und Kleinbuchstaben verwenden,
 - keine Leerzeichen, keine Sonderzeichen,
 - in der Regel nur einen Punkt (vor der Extension),
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
 - Groß- und Kleinbuchstaben verwenden,
 - keine Leerzeichen, keine Sonderzeichen,
 - in der Regel nur einen Punkt (vor der Extension),Befehl: `file <filename>`
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ **Anlegen eines neuen Verzeichnisses (Ordners, directory):**
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ **Anlegen eines neuen Verzeichnisses (Ordners, directory):**
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
oder im Terminal
`mkdir <ordnername>`
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Hilfreiche Tipps

- ▶ Befehlswiederholung mittels Pfeiltasten nach ↑ oder ↓
- ▶ Der Tabulator dient auch der Textergänzung.
- ▶ Copy and paste:
- ▶ Wahl der Namen von Dateien und Verzeichnissen
- ▶ Anlegen eines neuen Verzeichnisses (Ordners, directory):
im Filemanager: rechte Maustaste → neu Erstellen → Ordner
- ▶ Erstellen Sie im Verzeichnis P-WRI einen neuen Ordner mit angemessenem Namen (Bsp.: Java, testwiese,...)

Programmierung - Einführung

- ▶ Berechnung einer Formel

$$y = x + 5,6$$

kann als Vorschrift interpretiert werden, y zu berechnen, wenn x gegeben ist.

- ▶ Im allgemeinen benötigt ein Computerprogramm **Eingabedaten** (x) und erzeugt daraus **Ausgabedaten** (y).
- ▶ Unser erstes Programm hat nur **Ausgabedaten**. Es soll die Namen ihrer Programmiergruppe ausgeben.

Programmierung - Einführung

- ▶ Berechnung einer Formel

$$y = x + 5,6$$

kann als Vorschrift interpretiert werden, y zu berechnen, wenn x gegeben ist.

- ▶ Im allgemeinen benötigt ein Computerprogramm **Eingabedaten** (x) und erzeugt daraus **Ausgabedaten** (y).
- ▶ Unser erstes Programm hat nur **Ausgabedaten**. Es soll die Namen ihrer Programmiergruppe ausgeben.

Programmierung - Einführung

- ▶ Berechnung einer Formel

$$y = x + 5,6$$

kann als Vorschrift interpretiert werden, y zu berechnen, wenn x gegeben ist.

- ▶ Im allgemeinen benötigt ein Computerprogramm **Eingabedaten** (x) und erzeugt daraus **Ausgabedaten** (y).
- ▶ Unser erstes Programm hat nur **Ausgabedaten**. Es soll die Namen ihrer Programmiergruppe ausgeben.

Programmierung - Einführung

Programmieren bedeutet:

1. Algorithmische Aufbereitung des zu programmierenden Prozesses (z.B: in welcher Reihenfolge berechne ich was?)
2. Schreiben des Quelltextes `<name>.java` nach den Regeln der Programmiersprache mit einem Texteditor (empfohlen `kate`)
3. Compilieren (übersetzen) und evtl. linken (verbinden) des Programmes:

```
javac <name>.java
```

4. Starten des Programmes mittels

```
java <name>
```

Programmierung - Einführung

Programmieren bedeutet:

1. Algorithmische Aufbereitung des zu programmierenden Prozesses (z.B: in welcher Reihenfolge berechne ich was?)
2. Schreiben des Quelltextes `<name>.java` nach den Regeln der Programmiersprache mit einem Texteditor (empfohlen `kate`)
3. Compilieren (übersetzen) und evtl. linken (verbinden) des Programmes:
`javac <name>.java`
4. Starten des Programmes mittels
`java <name>`

Programmierung - Einführung

Programmieren bedeutet:

1. Algorithmische Aufbereitung des zu programmierenden Prozesses (z.B: in welcher Reihenfolge berechne ich was?)
2. Schreiben des Quelltextes `<name>.java` nach den Regeln der Programmiersprache mit einem Texteditor (empfohlen `kate`)
3. Compilieren (übersetzen) und evtl. linken (verbinden) des Programmes:
`javac <name>.java`
4. Starten des Programmes mittels
`java <name>`

Programmierung - Einführung

Programmieren bedeutet:

1. Algorithmische Aufbereitung des zu programmierenden Prozesses (z.B: in welcher Reihenfolge berechne ich was?)
2. Schreiben des Quelltextes `<name>.java` nach den Regeln der Programmiersprache mit einem Texteditor (empfohlen `kate`)
3. Compilieren (übersetzen) und evtl. linken (verbinden) des Programmes:

```
javac <name>.java
```

4. Starten des Programmes mittels

```
java <name>
```

Programmierung - Einführung

Programmieren bedeutet:

1. Algorithmische Aufbereitung des zu programmierenden Prozesses (z.B: in welcher Reihenfolge berechne ich was?)
2. Schreiben des Quelltextes `<name>.java` nach den Regeln der Programmiersprache mit einem Texteditor (empfohlen `kate`)
3. Compilieren (übersetzen) und evtl. linken (verbinden) des Programmes:
`javac <name>.java`
4. Starten des Programmes mittels
`java <name>`

My first JAVA code

Beim Programmieren ist eine gewisse Struktur des Programmtextes einzuhalten.

Der minimal nötige Rahmen ist durch folgende Zeilen gegeben:

```
public class <name> {  
    public static void main(String[] args){  
  
        <programmtext>  
    }  
}
```

My first JAVA code

```
public class <name> {  
    public static void main(String[] args){  
  
        <programmtext>  
    }  
}
```

Übertragen Sie obige Zeilen mit einem Editor (z.B. [kate](#)) in die Datei [<name>.java](#).

Wählen Sie einen [<name>](#) und lassen Sie den [<programmtext>](#) noch leer.

Speichern Sie die Datei in Ihrem neuen Ordner in P-WRI ab.

My first JAVA code

Zeichenketten (Text) werden als **String** bezeichnet.

My first JAVA code

Zeichenketten (Text) werden als **String** bezeichnet.

Dazu werden die Zeichen in Anführungszeichen eingeschlossen.

Bsp: "das ist ein **String**".

My first JAVA code

Zeichenketten (Text) werden als **String** bezeichnet.

Ein Ausgabebefehl (output) von Strings in die Standardausgabe ist: `System.out.println(String);`

Strings können durch + zusammengefügt werden:

Bsp. "Vorname "+"Nachname" \cong "Vorname Nachname".

My first JAVA code

Ein Ausgabebefehl (output) von Strings in die Standardausgabe ist: `System.out.println(String);`

Strings können durch + zusammengefügt werden:

Bsp. "Vorname " + "Nachname" \cong "Vorname Nachname".

Ergänzen Sie in Ihrem File `<name>.java` den `<programmtext>` durch einen Ausgabebefehl mit den Namen Ihrer Programmiergruppe.

My first JAVA code

Ein Ausgabebefehl (output) von Strings in die Standardausgabe ist: `System.out.println(String);`

Strings können durch + zusammengefügt werden:

Bsp. "Vorname" + "Nachname" \cong "Vorname Nachname".

Ergänzen Sie in Ihrem File `<name>.java` den `<programmtext>` durch einen Ausgabebefehl mit den Namen Ihrer Programmiergruppe.

Compilieren Sie `<name>.java` und starten Sie das Programm.



Grundelemente von JAVA

Java ist eine objektorientierte, Hardware-unabhängige Programmiersprache, die für numerische Rechnungen **gut** geeignet ist. **Sehr gut**, wenn sie schneller wäre.



Grundelemente von JAVA

Java ist eine objektorientierte, Hardware-unabhängige Programmiersprache, die für numerische Rechnungen **gut** geeignet ist. **Sehr gut**, wenn sie schneller wäre.

Es gibt zwei Arten von Java-Programmen:

Applets - laufen innerhalb von Internetseiten und **Applikationen** - sind normal ausführbare Programme.

Zeichensatz - Unicode (2 Byte)

0000

C0 Controls and Basic Latin

007F

	000	001	002	003	004	005	006	007
0	NAUL [NL] 00000000	DLR [LF] 00000010	SP [] 00000020	@ 00000040	P 00000050	` 00000060	p 00000070	
1	NONB [NBSP] 000000A0	!	! 00000020	A 00000041	Q 00000051	a 00000061	q 00000071	
2	STX [STX] 00000002	"	" 00000022	2 00000032	B 00000042	R 00000052	b 00000062	r 00000072
3	ETX [ETX] 00000003	#	# 00000023	3 00000033	C 00000043	S 00000053	c 00000063	s 00000073
4	EOF [EOF] 00000004	\$	\$ 00000024	4 00000034	D 00000044	T 00000054	d 00000064	t 00000074
5	ENQ [ENQ] 00000005	%	% 00000025	5 00000035	E 00000045	U 00000055	e 00000065	u 00000075
6	ACK [ACK] 00000006	&	& 00000026	6 00000036	F 00000046	V 00000056	f 00000066	v 00000076
7	BEL [BEL] 00000007	'	' 00000027	7 00000037	G 00000047	W 00000057	w 00000067	w 00000077
8	BS [BS] 00000008	((00000028	8 00000038	H 00000048	X 00000058	h 00000068	x 00000078
9	HT [HT] 00000009)) 00000029	9 00000039	I 00000049	Y 00000059	i 00000069	y 00000079
A	LF [LF] 0000000A	*	* 0000002A	J 0000003A	Z 0000004A	j 0000006A	z 0000007A	
B	VT [VT] 0000000B	+	+ 0000002B	K 0000003B	[0000004B	k 0000006B	{ 0000007B	
C	FF [FF] 0000000C	,	,	L 0000003C	\ 0000004C	l 0000006C	 0000007C	
D	CR [CR] 0000000D	-	- 0000002D	M 0000003D] 0000004D	m 0000006D	} 0000007D	
E	SO [SO] 0000000E	.	.	N 0000003E	^ 0000004E	n 0000006E	~ 0000007E	
F	SI [SI] 0000000F	/	/ 0000002F	O 0000003F	_ 0000004F	o 0000006F	[DEL] 0000007F	

0080

C1 Controls and Latin-1 Supplement

00FF

	008	009	00A	00B	00C	00D	00E	00F
0	XXX [] 00000080	XXX [] 00000081	o 00000082	À 00000083	Ä 00000084	à 00000085	ä 00000086	
1	XXX [] 00000087	XXX [] 00000088	ï 00000089	± 0000008A	Á 0000008B	Ñ 0000008C	á 0000008D	ñ 0000008E
2	SPRM [] 0000008F	XXX [] 00000090	ç 00000091	² 00000092	Â 00000093	Ò 00000094	â 00000095	ò 00000096
3	NON [] 00000097	STB [] 00000098	£ 00000099	³ 0000009A	Ã 0000009B	Ó 0000009C	ã 0000009D	ó 0000009E
4	XXX [] 0000009F	XXX [] 000000A0	XXX [] 000000A1	XXX [] 000000A2	XXX [] 000000A3	XXX [] 000000A4	XXX [] 000000A5	XXX [] 000000A6
5	XXX [] 000000A7	XXX [] 000000A8	XXX [] 000000A9	XXX [] 000000AA	XXX [] 000000AB	XXX [] 000000AC	XXX [] 000000AD	XXX [] 000000AE
6	XXX [] 000000AF	XXX [] 000000B0	XXX [] 000000B1	XXX [] 000000B2	XXX [] 000000B3	XXX [] 000000B4	XXX [] 000000B5	XXX [] 000000B6
7	XXX [] 000000B7	XXX [] 000000B8	XXX [] 000000B9	XXX [] 000000BA	XXX [] 000000BB	XXX [] 000000BC	XXX [] 000000BD	XXX [] 000000BE
8	XXX [] 000000BF	XXX [] 000000C0	XXX [] 000000C1	XXX [] 000000C2	XXX [] 000000C3	XXX [] 000000C4	XXX [] 000000C5	XXX [] 000000C6
9	XXX [] 000000C7	XXX [] 000000C8	XXX [] 000000C9	XXX [] 000000CA	XXX [] 000000CB	XXX [] 000000CC	XXX [] 000000CD	XXX [] 000000CE
A	XXX [] 000000CF	XXX [] 000000D0	XXX [] 000000D1	XXX [] 000000D2	XXX [] 000000D3	XXX [] 000000D4	XXX [] 000000D5	XXX [] 000000D6
B	XXX [] 000000D7	XXX [] 000000D8	XXX [] 000000D9	XXX [] 000000DA	XXX [] 000000DB	XXX [] 000000DC	XXX [] 000000DD	XXX [] 000000DE
C	XXX [] 000000DF	XXX [] 000000E0	XXX [] 000000E1	XXX [] 000000E2	XXX [] 000000E3	XXX [] 000000E4	XXX [] 000000E5	XXX [] 000000E6
D	XXX [] 000000E7	XXX [] 000000E8	XXX [] 000000E9	XXX [] 000000EA	XXX [] 000000EB	XXX [] 000000EC	XXX [] 000000ED	XXX [] 000000EE
E	XXX [] 000000EF	XXX [] 000000F0	XXX [] 000000F1	XXX [] 000000F2	XXX [] 000000F3	XXX [] 000000F4	XXX [] 000000F5	XXX [] 000000F6
F	XXX [] 000000F7	XXX [] 000000F8	XXX [] 000000F9	XXX [] 000000FA	XXX [] 000000FB	XXX [] 000000FC	XXX [] 000000FD	XXX [] 000000FE

Zeichensatz - Unicode (2 Byte)

3040

Hiragana

309F

	3040	3050	3060	3070	3080	3090
0	ぐ	だ	ば	む	ぬ	
1	あ	け	ち	ば	め	ゑ
2	あ	げ	ち	ひ	も	を
3	い	こ	つ	び	や	ん
4	い	ご	つ	び	や	づ
5	う	さ	づ	ふ	ゆ	か
6	う	ざ	て	ぶ	ゆ	け
7	え	し	で	ふ	よ	
8	え	じ	と	へ	よ	
9	お	す	ど	べ	ら	ゝ
A	お	ず	な	べ	り	ゝ
B	か	せ	に	ほ	る	ゝ
C	が	ぜ	ぬ	ぼ	れ	ゝ
D	き	そ	ね	ぼ	ろ	ゝ
E	ぎ	ぞ	の	ま	わ	ぶ
F	く	た	は	み	わ	ち

Zeichen in Java `\u3055`

Zeichensatz - Unicode (2 Byte)

Wir beschränken uns auf die Zeichen der Tabelle

C0 Controls and Basics Latin.

Warum? Da manche Rechner nur "englisch" installiert haben, sind dort z.B. die Umlaute nicht darstellbar.

Umlaute (beispielweise ä) in Zeichenketten werden mit `ae` oder `\u00e4` dargestellt.

Zeichensatz - Unicode (2 Byte)

Wir beschränken uns auf die Zeichen der Tabelle

C0 Controls and Basics Latin.

Warum? Da manche Rechner nur "englisch" installiert haben, sind dort z.B. die Umlaute nicht darstellbar.

Umlaute (beispielweise ä) in Zeichenketten werden mit `ae` oder `\u00e4` dargestellt.

Umlaute bzw. entsprechende Sonderzeichen im Java-Quelltext werden als **Fehler** gehandelt.

- ▶ **Format des Java-Quelltextes**
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
Der Quelltext ist an **kein** Format gebunden, schreiben Sie aber Ihre Programme zur besseren Lesbarkeit strukturiert
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

▶ Format des Java-Quelltextes

Der Quelltext ist an **kein** Format gebunden, schreiben Sie aber Ihre Programme zur besseren Lesbarkeit strukturiert
Nicht:

```
public class <name> { public static void main(String[] args){  
<programmtext> } }
```

▶ Kommentare

▶ Primitive Datentypen

▶ Bezeichnernamen

▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ **Kommentare**
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare : Java kennt drei Schreibweisen

// Kommentar in einer Zeile

/* Kommentar , der sich über viele Zeilen hinzieht */

/** Kommentar zur Auswertung mittels javadoc */

- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ **Primitive Datentypen**
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen

Datentyp	Standardwert	Beschreibung	Bemerkungen
boolean	false	wahr oder falsch	true
char	\u0000	Unicode-Zeichen	
byte	0	±8-bit Integer	
short	0	±16-bit Integer	
int	0	±32-bit Integer	
long	0	±64-bit Integer	42L (auch 42l)
float	0	32-bit Gleitkommazahl	0.0001F (auch f), 1e-4f 1. .1e+1 1.1e-1
double	0	64-bit Gleitkommazahl	0.0001, 1e-4 (auch d bzw. D möglich)

- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen

Datentyp	Standardwert	Beschreibung	Bemerkungen
boolean	false	wahr oder falsch	true
char	<code>\u0000</code>	Unicode-Zeichen	
byte	0	±8-bit Integer	
short	0	±16-bit Integer	
int	0	±32-bit Integer	
long	0	±64-bit Integer	42L (auch 42l)
float	0	32-bit Gleitkommazahl	0.0001F (auch f), 1e-4f 1. .1e+1 1.1e-1
double	0	64-bit Gleitkommazahl	0.0001, 1e-4 (auch d bzw. D möglich)

- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ **Bezeichnernamen**
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.

Wir schreiben im Quelltext: `datentyp Objektname ;`

- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.

Für die Namenswahl bestehen folgende Regeln:

- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.

Für die Namenswahl bestehen folgende Regeln:

1. Das erste Zeichen muss ein **Buchstabe**, **underscore** (`_`) oder **\$**-Zeichen sein.
2. Danach dürfen weitere Buchstaben (auch `_` und `$`) und Zahlen folgen.
3. Groß- und Kleinschreibung werden unterschieden.
4. Der Name darf **kein reserviertes Wort** von JAVA sein.
5. Der Name endet mit einem Trennzeichen: Leer-, Zeilenendezeichen, Zeilenvorschub, Tabulator, Kommentar oder `=`.
6. Wählen Sie sinnvolle Namen!

- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.

Für **primitive Datentypen** erfolgt die Zuweisung eines Anfangswertes durch

```
datentyp Objektname = wert;
```

- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen : Objekte müssen deklariert werden - sie bekommen einen Namen und evtl. gleich einen Anfangswert.

Für **primitive Datentypen** erfolgt die Zuweisung eines Anfangswertes durch

```
datentyp Objektname = wert;
```

Beispiel: **double** radius = 2.6e+2;

- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

Der Ausgabestring kann direkt durch Nutzung des Bezeichners erzeugt werden.

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

Der Ausgabestring kann direkt durch Nutzung des Bezeichners erzeugt werden.

Beispiel: `System.out.println("Radius = " + radius);`

- ▶ Format des Java-Quelltextes
- ▶ Kommentare
- ▶ Primitive Datentypen
- ▶ Bezeichnernamen
- ▶ Ausgabe der Werte primitiver Datentypen

Beispiel: `System.out.println(" Radius = " + radius);`