Vorkurs zur Einführung in die Programmierung (HS 2023)

Aufgaben Kapitel 1 - 4

1 Einleitung

Die Programme der ersten beiden Tage sind einfach – oft nur eine Java Anweisung oder eine einfache Folge von Anweisungen. Versuchen Sie, diese Aufgaben mit den im Unterricht vorgestellten Konstrukten/Konzepten zu lösen.

Manche der Programme motivieren auch Konzepte, die später in der Woche eingeführt werden. Das Hauptziel dieser Aufgaben ist es, Ihnen die Gelegenheit zum Üben mit der JShell zu geben.

1. Schreiben Sie ein Programm, dass einen String bildet, der aus dem 1., 3. und 5. Zeichen eines Urprungsstrings besteht. Sie können davon ausgehen, dass der Ursprungsstring mindestens 5 Zeichen enthält.

Beispiel:

2. Gegeben sind drei ganze Zahlen. Finden Sie die grössten der drei Zahlen. (Sie können die Methode Math.max(int, int) verwenden um zwei Zahlen zu vergleichen.)

Beispiel:

3. Gegeben sind vier ganze Zahlen. Finden Sie die grössten der vier Zahlen. (Sie können die Methode Math.max(int, int) verwenden um zwei Zahlen zu vergleichen.)

Beispiel:

4. Gegeben sind drei ganze Zahlen. Finden Sie die Zahl, die den grössten absoluten Wert hat. Sie können die Methode Math.abs(int) verwenden, um den absoluten Wert einer Zahl zu erhalten.

Beispiel:

2 Variable

Für diese Programme sollen Sie (soweit sinnvoll) Variable verwenden, so dass Sie einfach diese Programmschritte für verschiedene Werte durchspielen können.

1. Gegegen Sie ein Ursprungsstring str. Schreiben Sie ein Programm, das einen String bildet, der aus dem 1., 3. und 5. Zeichen von str besteht. Sie können davon ausgehen, dass der Ursprungsstring str mindestens 5 Zeichen enthält.

Beispiel:

```
str: "abcdefg" --> "ace"
```

2. Gegeben sind zwei Strings ax und bx. Schreiben Sie ein Programm, das einen neuen String bildet, der aus den ersten drei Zeichen von ax und bx besteht. Sie können davon ausgehen, dass sowohl ax als auch bx je mindestens drei Zeichen enthalten.

Beispiel:

```
ax: "1234", bx: "abcde" --> "123abc"
```

3. Gegeben sind drei int Variable a, b, c. Setzen Sie die int Variable x auf den Wert der grössten der drei Variablen a, b, c. (Sie können die Methode Math.max(int, int) verwenden.) Beispiel:

```
a: 6, b: 9, c: 2 ---> x: 9
```

4. Gegeben sind vier int Variable a, b, c und d. Setzen Sie die int Variable x auf den Wert der grössten der vier Zahlen. (Sie können die Methode Math.max(int, int) verwenden um zwei Zahlen zu vergleichen.)

Beispiel:

```
a: -10, b: 6, c: 9, d: 7 ---> x: 9
```

5. Gegeben sind drei int Variable a, b, c. Setzen Sie die int Variable x auf den Wert der Variable, deren Wert den grössten absoluten Wert hat. (Sie können die Methode Math.abs(int) verwenden um den absoluten Wert einer Zahl zu erhalten.)

Beispiel:

```
a: -10, b: 6, c: 9, d: 7 ---> x: -10
```

3 Schritte und ihre Reihenfolge

Wenn Sie int Variable verwenden, so können Sie davon ausgehen, dass der Wertebereich des Basistyps int ausreicht, um alle Ergebnisse etc. zu speichern.

- 1. Gegeben sind drei int Variable a, b, c. Setzen Sie die int Variable x auf den Wert der grössten der drei Variable a, b, c. (Sie sollen die Methode Math.max(int, int) *n*icht verwenden.)
- 2. Gegeben sind vier int Variable a, b, c und d. Setzen Sie die int Variable x auf den Wert der grössten der vier Zahlen. (Sie sollen die Methode Math.max(int, int) *n*icht verwenden.)
- 3. In der Variable students ist die Anzahl der Studierenden, die das Study Center besuchen, gespeichert. Die Variable findetStatt gibt an, ob an einem Tag das Study Center geöffnet ist. Setzen Sie die Variable bedarf auf true, wenn an einem Tag, an dem das Study Center geöffnet hat, zwischen 10 und 20 (einschliesslich) oder zwischen 40 und 60 (einschliesslich) Studierende das Study Center besuchen.

Beispiele:

```
(a) geoeffnet: false
    students: 15
    --> bedarf: false
(b) geoeffnet: true
    students: 15
    --> bedarf: true
```

4. Die Stadtpolizei führt eine Geschwindigkeitskontrolle durch. Die Geschwindigkeit wird in einer int Variable geschwindigkeit gespeichert (und der gespeicherte Wert wird für alle weiteren Schritte als die gemessene Geschwindigkeit verwendet). Setzen Sie die int Variable strafe zu o wenn es keine Strafe gibt, zu 1 wenn es eine Ordnungsbusse gibt, und zu 2 wenn es einen Strafbefehl gibt. Wenn die Geschwindigkeit kleiner als 55 km/Std ist, dann gibt es keine Strafe. Für eine Geschwindigkeit zwischen 55 und 60 km/Std (einschliesslich) gibt es eine Ordnungsbusse, und wenn die Geschwindigkeit grösser ist, gibt es einen Strafbefehl.

Die Variable feiertag gibt an, ob der/die Fahrer/in an dem Tag Geburtstag feiert. In dem Fall wird die Grenze für jede Art von Strafe oben um 5 km/Std erhöht.

Beispiele:

```
(a) feiertag: false geschwindigkeit: 54
--> strafe: 0
(b) feiertag: true geschwindigkeit: 62
--> strafe: 1
(c) feiertag: false geschwindigkeit: 54 strafe: 0
```

5. Die Regeln für Strafen bei Geschwindigkeitsübertretungen werden geändert. Neu gibt es zusätzlich eine Variable 1ernfahrt, die angibt ob eine Lernfahrt durchgeführt wurde. In dem Fall gibt es einen Strafbefehl wenn die Geschwindigkeit grösser als 70 km/Std ist. Sonst bleiben die Bestimmungen unverändert. Schreiben Sie ein Programm, um die Variable strafe richtig zu setzen.

Beispiele:

(a) feiertag: false
 lernfahrt: true
 geschwindigkeit: 55
--> strafe: 1

(b) feiertag: false
 lernfahrt: true
 geschwindigkeit: 69

strafe: 1

4 Methoden

Für diese Programme sollen Sie Methoden entwickeln, die die gewünschten Rückgabewerte liefern. In den meisten Fällen geben die Aufgabenstellungen den Namen der Methode, die Parameter (sowohl Typen als auch Reihenfolge) sowie den Rückgabewert vor. Sie sollten diese nicht ändern. (Wenn Sie die Aufgaben auf Ihrem Computer lösen, spielen die Namen der Methoden keine Rolle, aber wenn Sie später im Kurs Aufgaben lösen, dann sind diese Vorgaben sehr wichtig. Gewöhnen Sie sich daran, diese Vorgaben zu erfüllen.) Die Namen der Parameter können Sie natürlich immer frei wählen. In vielen Fällen können Sie bessere Namen finden als wir vorschlagen.

Jede Ihrer Lösungen sollten Sie mit verschiedenen Beispielwerten (für die geforderten Parameter) ausprobieren.

Wenn Sie int Variable verwenden, so können Sie davon ausgehen, dass der Wertebereich des Basistyps int ausreicht, um alle Ergebnisse etc. zu speichern.

 Schreiben Sie eine Methode, die einen String entgegen nimmt, und den Buchstaben in der Mitte (in einem String) zurückgibt. Wenn der String der leere String ist, so soll die Methode den leeren String zurückgeben.

```
String grabMiddle(String s)

Beispiel:

"Eisenbahn" -> "n"

"Autobahn" -> "o"

"ab" -> "a"

"b" -> "b"

"" -> ""
```

 Schreiben Sie eine Methode, die prüft ob die drei int Parameter der Grösse nach aufsteigend geordnet sind. Wenn das der Fall ist, soll die Methode den Wert true zurück geben, andernfalls den Wert false.

```
boolean inOrder(int a, int b, int c)
Beispiel:
inOrder(3, 5, 7) -> true
inOrder(5, 3, 7) -> false
```

3. Gegeben sind vier Zahlen. Schreiben Sie eine Methode die den den Wert der grössten der vier Zahlen zurück gibt.

```
int iMax(int a, int b, int c, int d)
```

4. Schreiben Sie eine Methode, die einen String und eine positive ganze Zahl width entgegen nimmt, und einen String der mittleren (2*width+1) Zeichen des Strings zurückgibt.

```
String grabMiddlePlus(String s, int width)
```

Sollte der Ursprungsstring nicht lang genug sein, also nicht mindestens (2*width+1) Zeichen enthalten, so geben Sie den Ursprungsstring zurueck. Für einen String s gibt grabMiddlePlus(s, o) den selben Wert zurück wie grabMiddle(s). Die Methode grabMiddlePlus(s, 1) haben wir im Unterricht kennengelernt.

5. Schreiben Sie eine Methode, die jedes Zeichen eines Strings verdoppelt. Für den leeren String soll Ihre Methode den leeren String zurück geben.

```
String doubleString(String s)
Beispiele:
doubleString("ab") -> "aabb"
doubleString("112") -> "111122"
```

6. Schreiben Sie eine Methode die einen String entgegennimmt und einen neuen String konstruiert, in dem zwischen je zwei Zeichen des Ursprungsstrings ein "*" eingefügt ist.

```
String insertStar(String s)
Beispiel:
insertStar("a") -> "a", insertStar("abc") -> "a*b*c".
```

- 7. Gegeben sei ein String s der Länge 1s; mit 1s > 0. Dieser String hat die Zeichen s_0 , s_1 , ..., s_{ls-1} . s kann in einen String t mit Länge 1t eingebettet werden wenn
 - lt >= ls, und
 - für jedes $i \in \{0, ..., ls 1\}$ gibt es ein $m(i) \in \{0, ..., lt 1\}$ so dass $s_i = t_{m(i)}$ ist, und
 - wenn für $i, j \in \{0, ..., ls 1\}$ i < j gilt, dann (und nur dann) gilt auch m(i) < m(j), und
 - wenn für $i, j \in \{0, ..., ls 1\}$ $i \neq j$ gilt, dann (und nur dann) gilt auch $m(i) \neq m(j)$.

Schreiben Sie eine Methode embed(String s, String t) die prüft, ob der String s in t eingebettet werden kann. In diesem Fall ist der Rückgabewert true, sonst false.

```
boolean embed(String s, String t)
Beispiel:
embed("ast", "faster") -> true
embed("mars", "Aminotransferase") -> true
embed("1135", "0123456") -> false
```