Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Programmierung 1
WiSe 25/26
Moritz Balg
27.10.2025

Übungsblatt 4

- Die Datentypen: char und String -

Aufgabe 1: Einlesen von Daten

Schreiben Sie zwei Java-Programme, die jeweils drei ganze Zahlen addieren und das Ergebnis auf dem Bildschirm ausgeben. Die Programme sollen sich in der Eingabe der drei Zahlen unterscheiden: Eingabe über die Kommandozeile und Eingabe über die Tastatur während des Programmlaufs.

Info

In der Vorlseung finden Sie entsprechende Beispiele.

Aufgabe 2: 👾 Caesar-Verschlüsselung

Eine einfache und alte Verschlüsselungmethode für Texte ist die Caesar-Verschlüsselung. Ausgangspunkt ist ein Buchstabe b und ein Schlüssel $k \in \mathbb{N}$. Das Verfahren arbeitet so, dass der Buchstabe durch den Buchstaben ersetzt wird, der k Buchstaben weiter im Alphabet erscheint. Dabei ist zu beachten, dass beim „Hinauslaufen“ über Z mit A weiter gezählt wird. Der nächste Buchstabe nach Z ist also A. b = H und k = 1 ergibt als Ergebnis I.

Schreiben Sie ein Java-Programm Caesar, das für einen Schlüssel k mit $k \in \mathbb{N}$ (also z.B. $k = 100$) und einen Großbuchstaben b den verschlüsselten Großbuchstaben auf dem Bildschirm ausgibt. Die beiden Argumente werden über die Tastatur zur Laufzeit dem Programm übergeben, zuerst der Schlüssel k in einer Zeile und anschließend der Buchstabe b in einer Zeile.

Die Ausgabe besteht nur aus dem kodierten Resultatbuchstaben zu Beginn der Ausgabezeile gefolgt von einem Zeilenumbruch.

Beispiel

Eingabe:
5
X
Ausgabe:
C

Überlegen Sie sich, wie Sie mit den Hilfmitteln, die bis jetzt bekannt sind, das „Hinauslaufen“ geeignet behandeln können. Nutzen Sie Eigenschaften der Zeichencodierung aus sowie geeignete arithmetische Operationen. Wie könnte die Dekodierung funktionieren?

Info

Modulo-Operation, Zeichen - ’A’, Typumwandlung zwischen int und char

Aufgabe 3: Erweiterte Caesar-Verschlüsselung

Erweitern Sie die frühere Aufgabe zur Caesar-Verschlüsselung dahingehend, dass statt eines einzelnen Buchstabens ein kompletter String verschlüsselt werden kann, indem sukzessive alle einzelnen Zeichen des Strings verschlüsselt und auf dem Bildschirm ausgegeben werden.

Info

`t.length()` und `t.charAt(i)`

Beispiel

Eingabe:
1
HALLO

Ausgabe:
Buchstabe H verschluesselt I Buchstabe A verschluesselt B Buchstabe L verschluesselt M Buchstabe L verschluesselt M Buchstabe O verschluesselt P

Aufgabe 4: Werte von der Kommandozeile einlesen

Schreiben Sie ein Java-Programm, das bei Laufzeit von der Tastatur eine Eingabezeile in folgendem Format (in EBNF beschrieben) einliest:

`(vor|nach) <c> in <string>`

Darin bezeichnet `<c>` ein vom Nutzer gewähltes Zeichen und `<string>` eine vom Nutzer gewählte Zeichenfolge — vor, nach und in sind buchstäbliche Zeichenfolgen (“Terminale” in der EBNF-Formulierung). Die Zeichenfolge `<string>` soll kein Leerzeichen enthalten.

Ein Beispiel einer solchen Eingabe wäre:
`nach s in durstiger`

Das Programm soll dann die Zeichenkette ausgeben, die nach (bzw. vor bei Eingabe von vor) dem ersten (bzw. letzten) Vorkommen des Zeichens `<c>` in der Zeichenkette `<string>` steht. Nach der Ausgabe soll ein Zeilenumbruch erfolgen. Im gegebenen Beispiel wäre dies:

`tiger`

Kommt das Zeichen `<c>` nicht in der Zeichenkette `<string>` vor, soll eine leere Zeile ausgegeben werden. Wenn die Eingabe nicht dem vorgeschriebenen Format entspricht, soll eine Fehlermeldung ausgegeben werden.

Info

- Verwenden Sie Methoden der Klasse String (siehe Aufgabe 1)!
- Sie können mit einer Selektionsanweisungen arbeiten, die in Java den Aufbau hat:

```
if ( ein logischer Ausdruck ) {
  Aktion 1
} else {
  Aktion 2
}
```

Aufgabe 5: 👾 Erweiterung von Datumsangaben

Schreiben Sie ein Java-Programm DatumErweiterung, das zu einer Datumsangabe eine Zeit addiert und das resultierende Datum ausgibt. Die Eingabe geschieht dabei über die Tastatur zur Laufzeit und enthält in dieser Reihenfolge folgende Angaben (alles natürliche Zahlen):

- eine Jahreszahl j ($1600 \leq j \leq 3000$)
- eine Monatszahl m ($1 \leq m \leq 12$)
- eine Tageszahl t ($1 \leq t \leq 31$) (je nach Monat bis zum zulässigen Maximalwert des Monats)
- eine Stundenzahl s ($0 \leq s \leq 23$)
- eine Minutenzahl m1 ($0 \leq m \leq 59$)
- eine Minutenzahl m2, die zu dem Datum addiert werden soll ($0 \leq m2 \leq 40000$).

Auf das Datum, das mit den ersten 5 Werten spezifiert wird, soll man die letzte Angabe m2 addieren, also das Datum bestimmen, das m2 Minuten später ist. Ignorieren Sie Schaltjahre, der Februar hat also in jedem Jahr 28 Tage! Aufgrund der Beschränkung bei m2 kann es höchstens einen Monat Überlauf geben. (Sie können sich ja einmal überlegen, welche zusätzliche Komplexität durch den Einbezug von Schaltjahren und durch größere m2 -Werte hinzukommen würde). Die Ausgabe soll nacheinander in einer Zeile und durch ein Leerzeichen getrennt die Resultatwerte für Jahr, Monat, Tag, Stunde und Minute ausgeben. Die Zeile wird durch ein Zeilenende abgeschlossen.

Beispiel

Eingabe:
2025 10 7 16 30 30
Ausgabe:
2025 10 7 17 0

weil 30 Minuten nach 16:30 am 7.10.2023 ist: 17:00 am 7.10.2023.

Beispiel

Eingabe:
2025 12 31 23 59 2
Ausgabe:
2026 1 1 0 1

weil 2 Minuten nach 23:59 am 31.12.2025 ist: 0:01 am 1.1.2026

Achtung

Verwenden Sie geeignete Operationen auf ganzzahligen Datentypen. Sie brauchen keine Schleifen.