

Vorkurs Informatik WS 2020/2021 Ergänzende praktische Aufgaben

Aufgabe 1

Schreiben Sie eine Java-Methode, welche die Seitenlänge eines Quadrats übergeben bekommt, die dann im Textmodus ein Quadrat dieser Größe ausgibt. Wird beispielsweise 5 übergeben, so soll die Ausgabe wie folgt aussehen:

```
*****
*   *
*   *
*   *
*   *
*****
```

Schreiben Sie zum Testen der Methode ein Java-Programm, das diese Methode (als statische Methode) sowie ein Hauptprogramm umfasst. Das Hauptprogramm soll die Quadrate der Größe 1 bis 5 ausgeben.

Aufgabe 2

Schreiben Sie eine nicht-rekursive Java-Methode `power`, die zwei Ganzzahlen x und y akzeptiert und die Potenz x^y berechnet.

Schreiben Sie zum Testen der Methode ein Java-Programm, das diese Methode (als statische Methode) sowie ein Hauptprogramm umfasst. Das Hauptprogramm soll alle Potenzen für x und y zwischen -2 und 2 ausgeben.

Aufgabe 3

Schreiben Sie eine rekursive Variante der Methode `power` analog zur Aufgabe 2. Diese Methode soll keine expliziten Schleifen verwenden, sondern das Ergebnis durch rekursiven Aufruf von sich selbst ermitteln.

Aufgabe 4

Nehmen Sie an, wir hätten in Java keine Addition und Subtraktion. Stattdessen gibt es die Anweisungen `x++`; (die Variable x erhält ihren um 1 erhöhten Wert zugewiesen) und `x--`; (die Variable x erhält ihren um 1 erniedrigten Wert zugewiesen).

Schreiben Sie eine Klasse `Arithmetik`, die statische Methoden `addition` und `subtraktion` zur Verfügung stellt. Diese Funktionen haben jeweils zwei ganzzahlige Parameter und geben die Summe beziehungsweise die Differenz der Parameter als Wert zurück. Zur Berechnung stehen nur die am Anfang genannten Operationen zur Verfügung.

Schreiben Sie zum Testen der Methode ein Java-Programm, in dessen Hauptprogramm die Summen und Differenzen aller Paare von Zahlen zwischen -2 und 2 mit den Methoden der Klasse `Arithmetik` und mit der üblichen Addition berechnet werden. Für jedes Paar sollen die Werte der beiden Zahlen sowie die beiden Summen auf einer Zeile, jeweils durch Komma getrennt, ausgegeben werden.

Aufgabe 5

Schreiben Sie ein Java-Programm, das den ganzzahligen 2er-Logarithmus einer natürlichen Zahl berechnet. Erinnerung: Der ganzzahlige 2er-Logarithmus einer natürlichen Zahl n ist die Zahl k mit der Eigenschaft

$$2^k \leq n < 2^{k+1}$$

Aufgabe 6

Schreiben Sie eine rekursive Prozedur `binomial(int n, int k)`, die den Binomialkoeffizienten $\binom{n}{k}$ nach folgender Vorschrift bestimmt:

$$\binom{n}{n} = \binom{n}{0} = 1$$

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \text{ für } n > 0 \text{ und } n > k > 0$$

Testen Sie Ihr Programm mit den Werten $(n = 12, k = 6)$, $(n = 11, k = 5)$ und $(n = 11, k = 6)$.

Aufgabe 7

Schreiben Sie eine rekursive Methode `max`, die das größte Element eines vorgegebenen `int`-Arrays liefert. Sehen Sie den rekursiven Ansatz darin, dass das größte Element eines Arrays entweder dessen erstes Element oder das größte der verbleibenden Elemente ist.

Aufgabe 8

Eine seit über 2000 Jahren bekannte Form der Primzahlbestimmung wird als ‘‘Sieb des Eratosthenes‘‘ bezeichnet. Angenommen, es sollen alle Primzahlen im Bereich von 1 bis 1000 bestimmt werden. Dazu wird wie folgt vorgegangen:

- Es wird ein boolesches Array mit 1000 Elementen erstellt und alle Elemente auf `true` initialisiert.
- Das erste Element wird auf `false` geändert.
- In einer Schleife werden folgende Schritte ausgeführt:
 - Es wird das nächste Element gesucht, dessen Wert `true` ist. Falls kein weiteres Element diese Bedingung erfüllt, wird die Schleife beendet.
 - Der Wert aller Elemente, deren Position ein ganzzahliges Vielfaches der aktuellen Position ist, wird auf `false` geändert. Wurde beispielsweise das fünfte Element gefunden, so werden auf diese Weise die Elemente 10, 15, 20, . . . , 995 und 1000 ausgestrichen.
- Es werden nun die Positionen aller Arrayelemente ausgegeben, deren Wert `true` ist. Es sind die Primzahlen zwischen 1 und 1000.

Schreiben Sie ein Java-Programm, das diese Berechnung durchführt und die gefundenen Primzahlen ausdrückt.