Die Hausaufgaben müssen von den Studierenden einzeln bearbeitet und abgegeben werden. Für die Hausaufgabe sind die aktuellen Informationen vom Blog `https://seblog.cs.uni-kassel.de/ws2223/einfuhrung-in-die-informatik/` zu berücksichtigen.

Abgabefrist ist der 17.11.2022 - 23:59 Uhr

Vorbereitung

Zur Abgabe dieser Hausaufgabe muss zunächst ein neues Respository angelegt werden. Mit dem folgenden Link kannst du das Respository für Hausaufgabe 4 anlegen bzw. einsehen.

`https://classroom.github.com/a/Rb4_1HVK`

Achte bei jeder Aufgabe auf das verlangte Dateiformat. Andere Formate werden nicht akzeptiert und folglich mit **0** Punkten gewertet. Handschriftliche Abgaben werden nicht akzeptiert.

Es sollen nur die bis zur jeweiligen Übung vermittelten Konzepte verwendet werden.

Nicht oder zu spät abgegebene (Teil-)Aufgaben werden mit 0 Punkten bewertet.

Aufgabe 1 - Gültigkeit (8P)

Gegeben ist folgender Programmcode:

```
1  int sitUpRotation = 5;
2  int armsRotation = 2;
3  int chillRotation = 4;
4  int daysCounter = 1;
5
6  for (int week = 1; week < 5; week++) {
7      for (int day = 1; day < 8; day++) {
8          String trainingType = null;
9          boolean doTraining = false;
10
11         if ((daysCounter % sitUpRotation) == 0){
12             trainingType = " situps";
13             doTraining = true;
14         } else if ((daysCounter % armsRotation) == 0){
15             trainingType = " arms";
16             doTraining = true;
17         }
18
19         if (doTraining && (daysCounter % chillRotation) == 0){
20             trainingType = " chilling";
21         }
22
23         if (doTraining) {
24             System.out.println("Week " + week + " | Day " + day + ": " + trainingType);
25         }
26
27         daysCounter++;
28     }
29 }
30 /exit
```

Abbildung 1: gymPlanner.jsh

a) Erstelle eine Tabelle mit Variablenbelegung für den ersten, zweiten und vierten Durchlauf der inneren `for`-Schleife (Woche 1 Tag 1 und Woche 1 Tag 2 und Woche 1 Tag 4)

Die gesamte Aufgabe 1 soll in einem Textdokument abgegeben werden. Um die Tabelle darzustellen, halte dich bitte an das folgende Format. Im Folgenden sind bereits die Kopfzeile und die ersten drei Tabellen-Zeilen vorgegeben. Verwende wie dargestellt das Zeichen `|` zum visuellen Trennen von Zellen.

Verwende außerdem das Format `Z<Nummer>` um die Zeilennummer anzugeben, z. B. `Z6` für Zeile 6. Verwende ebenso das Format `D<Nummer>` um die Durchlaufnummer anzugeben, z. B. `D1` für den ersten Durchlauf.

```
Zeile   | daysCount | week | day | trainingType | doTraining
Z4      | 1         | -    | -   | -            | -
Z6      | 1         | 1    | -   | -            | -
Z7 D1   | 1         | 1    | 1   | -            | -
```

b) Erläutere die Ausgabe für Woche 3 Tag 6:

- b1) Wie lautet die Ausgabe?
- b2) Welche Art von Training wurde angegeben, oder wurde diese durch einen Ruhetag ersetzt? Wenn ja, welches Training wäre ersetzt? Erläutere!

Gib deine Lösung in einer Datei mit dem Namen `exercise.txt` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 2 - verschachtelte `for`-Schleife (6P)

Schreibe ein Programm, welches mit Hilfe einer verschachtelten for-Schleife die Multiplikations-Tafel einer eingegebenen Zahl ausgibt.

Zunächst muss vom User eine Zahl eingelesen werden. Für alle Eingaben zwischen (inklusive) 1 und (inklusive) 15 soll die Tafel schön formatiert ausgegeben werden, sodass alle Zahlen einer Spalte linksbündig untereinander stehen (siehe Ausgabe). Das Programm soll auch für weitere Integer-Eingaben funktionieren, dabei ist es aber in Ordnung, wenn die Formatierung aufgrund der vielen Zahlen nicht mehr so ordentlich aussieht.

Tipp: Bei der Formatierung der Ausgabe kannst du den String `"\t"` verwenden. Dieser fügt einen Tab ein.

Die Zählvariablen der äußeren Schleife soll `outer` heißen. Die Zählvariablen der inneren Schleife soll `inner` heißen.

Setze am Ende deines Programms `/exit`.

Ausgabe für User-Input 3:

```
Please enter positive number: 3
1       2       3
2       4       6
3       6       9
```

Ausgabe für User-Input 9:

```
Please enter positive number: 9
1       2       3       4       5       6       7       8       9
2       4       6       8       10      12      14      16      18
3       6       9       12      15      18      21      24      27
4       8       12      16      20      24      28      32      36
5       10      15      20      25      30      35      40      45
6       12      18      24      30      36      42      48      54
7       14      21      28      35      42      49      56      63
8       16      24      32      40      48      56      64      72
9       18      27      36      45      54      63      72      81
```

Ausgabe für User-Input 15 (die Schriftgröße wurde kleiner gemacht, damit alles aufs Aufgabenblatt passt):

```
Please enter positive number: 15
1       2       3       4       5       6       7       8       9       10      11      12      13      14      15
2       4       6       8       10      12      14      16      18      20      22      24      26      28      30
3       6       9       12      15      18      21      24      27      30      33      36      39      42      45
4       8       12      16      20      24      28      32      36      40      44      48      52      56      60
5       10      15      20      25      30      35      40      45      50      55      60      65      70      75
6       12      18      24      30      36      42      48      54      60      66      72      78      84      90
7       14      21      28      35      42      49      56      63      70      77      84      91      98      105
8       16      24      32      40      48      56      64      72      80      88      96      104     112     120
9       18      27      36      45      54      63      72      81      90      99      108     117     126     135
10      20      30      40      50      60      70      80      90      100     110     120     130     140     150
11      22      33      44      55      66      77      88      99      110     121     132     143     154     165
12      24      36      48      60      72      84      96      108     120     132     144     156     168     180
13      26      39      52      65      78      91      104     117     130     143     156     169     182     195
14      28      42      56      70      84      98      112     126     140     154     168     182     196     210
15      30      45      60      75      90      105     120     135     150     165     180     195     210     225
```

Gib deine Lösung in einer Datei mit dem Namen `table.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 3 - Arrays (9P)

Schreibe ein Programm, bei dem ein Array eingelesen wird. Anschließend werden alle Einträge des Arrays addiert und die Rechnung ausgegeben.

Im ersten Teil des Programms wird der User gefragt, wie viele Zahlen addiert werden sollen. Mithilfe einer `for`-Schleife sollen entsprechend viele Zahlen eingelesen werden und dabei nach jeder Eingabe der aktuelle Stand des Arrays ausgegeben werden. Nutze dafür die kennengelernte Methode `Arrays.toString()` zum Ausgeben. Die Zählvariable der Schleife soll `i` heißen.

Nachdem das Einlesen abgeschlossen ist, soll mithilfe einer weiteren `for`-Schleife die Berechnung der Summe stattfinden. Zusätzlich wird die Schleife genutzt, um den String der Berechnung zusammenzusetzen. Die Zählvariable der Schleife soll `j` heißen.

Setze am Ende deines Programms `/exit`.

Ausgabe:

```
Please enter the amount of numbers you want to sum up: 5

Please enter your 1. number: 6
[ 6, 0, 0, 0, 0 ]
Please enter your 2. number: 2
[ 6, 2, 0, 0, 0 ]
Please enter your 3. number: 1
[ 6, 2, 1, 0, 0 ]
Please enter your 4. number: 1
[ 6, 2, 1, 1, 0 ]
Please enter your 5. number: 9
[ 6, 2, 1, 1, 9 ]

6 + 2 + 1 + 1 + 9 = 19
```

Gib deine Lösung in einer Datei mit dem Namen `arrays.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.

Aufgabe 4 - Methoden (12P)

a) Schreibe eine Methode `isPrimeNumber`. Als Parameter soll die Methode einen Integer haben. Dieser soll `candidate` genannt werden. Die Methode berechnet, ob die übergebene Zahl eine Primzahl ist und gibt entsprechend einen boolean zurück. Nutze zur Berechnung unter anderem eine `for`-Schleife und den Modulo-Operator!

b) Schreibe eine Methode `getNextPrimeNumber`. Als Parameter soll die Methode einen Integer haben. Dieser soll `startNumber` genannt werden. Die Methode berechnet die nächstgrößere Primzahl zur gegebenen Zahl und gibt diese Primzahl (Integer) zurück. Nutze zur Berechnung unter anderem eine `while`-Schleife und deine zurvor selbstgeschriebene Methode `isPrimeNumber`.

Hinweis: Eine Primzahl ist eine natürliche Zahl, die größer als 1 und ausschließlich durch sich selbst und durch 1 teilbar ist.

Das Grundgerüst des Programms ist vorgegeben (siehe Repository). Bearbeite den Code nur im dafür vorgesehen Bereich, der mit `TODO` markiert ist. Verändere ansonsten nichts am Code:

```
1  /open EidiInput.jsh
2
3  // ------ methods section ------
4
5  // TODO add methods here:
6
7  // ------ methods section end ------
8
9  System.out.print("Enter positive number: ");
10 int givenNumber = EidiInput.readInt();
11
12 boolean prime = isPrimeNumber(givenNumber);
13
14 if (prime) {
15     System.out.println("Your entered number is a prime number.");
16 } else {
17     System.out.println("Your entered number is NOT a prime number.");
18     int nextP = getNextPrimeNumber(givenNumber);
19     System.out.println("The next prime number is " + nextP + ".");
20 }
21
22 /exit
```

Das Programm muss für alle Integer-Eingaben funktionieren, die ≥ 2 sind und maximal dem größtmöglichen Integer-Wert entsprechen. Hier einige Ausgaben:

Ausgabe für User-Input 4:

```
Enter positive number: 4
Your entered number is NOT a prime number.
The next prime number is 5.
```

Ausgabe für User-Input 123456789:

```
Enter positive number: 123456789
Your entered number is NOT a prime number.
The next prime number is 123456791.
```

Ausgabe für User-Input 13:

```
Enter positive number: 13
Your entered number is a prime number.
```

Gib deine Lösung in einer Datei mit dem Namen `methods.jsh` im GitHub-Repository zur aktuellen Hausaufgabe ab.