https://classroom.github.com/a/kuxW8hj1 Markdown Introduction

Skill Diary Database Vöcklabruck

Informational Stuff

The trainers of CODERS.BAY decided to let you write a documentation of what you have learned.

Please keep this as your diary and write a short summary with code snippets and some theoretical topics daily in the last half an hour of the day.

Just to let you know this will be your personal reference book.

You are going to have three GIT Repositories where you document your progress of the course.

In this Repository we are gonna collect all information of the database classes.

I'm going to give you all reviews in the form of issues every week, where I'm going to review your documentary by the following criteria:

- code quality
- quality of content
- totality of your documentation.

As it is a personal documentation the styling and structure is up to your personal preferences.

Markdown Cheatsheet

Headlines

Markdown Syntax	How it looks
# TEXT	

This is a H1

TEXT

This is a H2

TEXT

This is a H3

TEXT

This is an H4

TEXT

This is an H5

TEXT

This is an H6

Emphasis

Italic

Markdown Syntax	How it looks
This text will be italic	*This text will be italic*
This text will be italic too	*This text will be italic too*

Bold

Markdown Syntax	How it looks
This text will be bold	**This text will be bold**
__This text will be bold too__	**This text will be bold too**

Lists

Unordered

Markdown Syntax	How it looks

* Item 1
* Item 2
TAB * Item 2a
TAB * Item 2b
	• Item 1
	• Item 2
	  • Item 2a
	  • Item 2b

Ordered

Markdown Syntax	How it looks
1. Item 1 2. Item 2	1. Item 1 2. Item 2

Images

Markdown Syntax	How it looks
![GitHub Logo](/images/logo.png) Format: ![Alt Text](url)	Displays the image in folder images which is called logo.png

Links

Markdown Syntax	How it looks
![GitHub](https://www.github.com) Format: ![Alt Text](url)	GitHub

Backslash Escape

If you need the characters, which are already used as special characters in Markdown you can use them if you put an \ before the character. For example you have to write * to use a *

Using code in Markdown

Markdown coverts text with four leading spaces into a code block; with GFM you can wrap your code with ``` to create a code block without the leading spaces. Add an optional language identifier and your code will get syntax highlighting.

````
```javascript
function test() {
 console.log("look ma', no spaces");
}
```
````

```javascript
function test() {
  console.log("look ma', no spaces");
}
```

Tables

You can create tables by assembling a list of words and dividing them with hyphens - (for the first row), and then separating each column with a pipe |

Markdown Syntax	How it looks
First Header | Second Header ---------- | ---------- Content cell 1 | Content cell 2 Content column 1 | Content column 2	

First Header	Second Header
Content cell 1	Content cell 2
Content column 1	Content column 2

https://classroom.github.com/a/vRYzdyrQ Erzähl was über dich

Java

Variablen und Datentpyen

Aufgabe 1.1 - Erzähl mir etwas über dich

Deklariere Variablen für dein Alter, deinen Vornamen, dein Geschlecht, deinen Nachnamen, dein Geburtsdatum, deinen Notendurchschnitt und dafür ob du verheiratet bist oder nicht. Überleg dir gut welchen Datentyp du für welche Variablen am besten verwenden solltest.

Gib alle Variablen mit `System.out.println` auf der Konsole aus.

Java

Variables and Datatypes

Exercise 1.1 - Tell me about yourself

Declare variables to express your age, first name, gender, last name, birthday, average grade and whether you are married or not. Think which datatype is well suited for which variable.

Print all variables to the console with `System.out.println`

https://classroom.github.com/a/LYZW_Y2- Einfache Rechenaufgaben

Java

Variablen und Datentypen

Aufgabe 1.2 - Einfache Rechenaufgaben

Deklariere zwei numerische Variablen mit beliebigen Werten. Errechne deren Summe, Differenz, Produkt und Quotient und gib die Rechnung mit dem Ergebnis mit `System.out.println`/`System.out.printf` auf der Konsole aus. Rechne einmal mit ganzzahligen und einmal mit gebrochenen Zahlen!

Bonus: Gib die Kommazahlen schön formatiert auf der Konsole aus.

Eine Ausgabe sollte folgendermaßen aussehen:

```
138 + 235 = 373
138 - 235 = -97
```

Java

Variables and datatypes

Exercise 1.2 - Simple Calculations

Declare two numeric variables with arbitary values. Calculate their sum, difference, product and quotient and print the calculation with the result on the console with `System.out.println`/`System.out.printf`. Perform all calculations with whole numbers and fractional numbers.

Bonus: Try to limit the decimal places of your calculations with the fractional numbers.

Your output should look like this:

```
138 + 235 = 373
138 - 235 = -97
```

https://classroom.github.com/a/tK6Xzl3t Arbeiten mit Strings

Lerninhalte dieser Übung

In dieser Übung ist das primäre Lernziel, den Umgang mit einer offiziellen Dokumentation zu lernen. Für einen Entwickler bzw. eine Entwicklerin ist es wichtig, sich die für den konkreten Fall notwendigen Informationen aus einer offiziellen Dokumentationsseite herauszuschen. Dafür ist es wichtig zu lernen, wie solche offiziellen Dokumentationsseiten aufgebaut und strukturiert sind und wonach man suchen soll.

Das zweite Lernziel ist es, den Umgang mit Strings zu Üben, da dieser Datentyp oft verwendet und bearbeitet werden muss.

Java

Variablen und Datentypen

Aufgabe 1.3 - Arbeiten mit Strings [🤓 Advanced]

Lege einen String mit dem Inhalt `“ Hello World! ”` an (enthält vorne und hinten Leerzeichen).

- Gib den String und seine Länge auf der Konsole aus.
- Gib den String in einigen Abwandlungen auf der Konsole aus:
    - alle Buchstaben in Großbuchstaben
    - alle Buchstaben in Kleinbuchstaben
    - ersetze “World” mit “Codersbay”
    - ohne Leerzeichen zu Beginn des Texts
- Gib den String 15 mal wiederholt mit einem Zeilenumbruch zum Trennen aus (ohne die Codezeile 15 mal zu kopieren 😉)

Suche auf der offiziellen Dokumentations-Seite von Strings nach nützlichen Methoden.

Java

Variables and datatypes

Exercise 1.3 - Working with Strings [🤓 Advanced]

Create a variable of type String with `“ Hello World! ”` as its content (contains leading and trailing spaces).

- Print the String and its length to the console.
- Print the String with some variations:
    - all letters in uppercase
    - all letters in lowercase
    - replace “World” with “Codersbay”
    - without the leading spaces
- Repeat the printed String 15 times seperated with linebreaks (don't copy the code 15 times 😉)

You might find the official documentation of the String class helpful.

Java

Zuweisungs- und logische Operatoren

Aufgabe 2.2 Eigenschaften einer Zahl

Schreibe ein Programm, dass von der Konsole eine Zahl einliest und ausgibt:

- ob es sich um eine runde Zahl handelt
- ob die Zahl gerade ist
- ob die Zahl deiner Glückszahl entspricht (denk dir hierfür einfach eine eigene Glückszahl aus und gib sie zu Beginn des Programms auf der Konsole aus)
- ob die Zahl zweistellig ist

Tipp: Für die ersten beiden Punkte wirst du die Modulo Funktion brauchen.

Java

Assignment and logical operators

Aufgabe 2.2 Characteristics of a number

Write a program that reads a number from the console and calculates and prints characteristics of it:

- is the number round
- is the number even
- does the number equal your lucky number (choose any lucky number for this and print it to the console at the start of the program)
- does it have two digits

Hint: You might find the modulo operation helpful for the first two exercises.

Java

Zuweisungs- und logische Operatoren

Aufgabe 2.3 Arbeits- oder Freizeit?

In der CODERS.BAY arbeiten wir von 8 bis 16 Uhr. Schreibe ein Programm, dass eine Zahl von der Konsole einliest und ausgibt ob die Stunde in der Arbeitszeit liegt oder nicht.

Bonus: von 12 bis 13 Uhr ist immer Mittagspause, gib also in der Zeit aus, dass Mittag ist.

Java

Assignment and logical operators

Exercise 2.3 free time or work time?

At CODERS.BAY we usually work from 8 to 16 o'clock. Write a program, that reads a number from the console and prints whether that hour is during our work time or not.

Bonus: We go for a lunch break between 12 and 13 o'clock, print that info in case the given number is between 12 and 13.

Java

Bedingungen

Noten übersetzen

Schreibe ein Programm, dass eine Schulnote in numerischer Form (1-5) in seine textuelle Form übersetzt:

- Bei einer 1 wird "Sehr gut" auf die Konsole geschrieben
- Bei einer 2 wird "Gut" auf die Konsole geschrieben
- Bei einer 3 wird "efriedigend" auf die Konsole geschrieben
- Bei einer 4 wird "Genügend" auf die Konsole geschrieben
- Bei einer 5 wird "Nicht Gengügend" auf die Konsole geschrieben

---

Java

Conditionals

Translating grades

Write a program that translates grades from its numerical form to a word representation.

- Print "Very good" in case of a 1.
- Print "Good" in case of a 2.
- Print "Satisfactory" in case of a 3.
- Print "Sufficient" in case of a 4.
- Print "Not sufficient" in case of a 5.

Java

Schleifen

Aufgabe 5.1 FizzBuzz

FizzBuzz ist ein Spiel um Kindern Division näher zu bringen. Die Regeln sind einfach - Reih um wird beginnend bei der Zahl 1 nach oben gezählt. Ist die Zahl durch drei teilbar, darf die Zahl allerdings nicht genannt werden - man muss Fizz sagen, ist die Zahl durch fünf teilbar muss Buzz gesagt werden. Und - ist die Zahl durch drei und fünf teilbar, muss FizzBuzz gesagt werden.

Als Beispiel: “Eins! Zwei! Fizz(3)! Vier! Buzz(5)! Fizz(6)! Sieben! Acht! Fizz(9)! Buzz(10)!”

Schreibe ein Programm, dass die Zahlen von 1 - 100 nach diesem Schema ausgibt. Du musst die Zahlen nicht in Wortform ausgeben, numerisch reicht.

Java

Repetitions

Exercise 5.1 FizzBuzz

FizzBuzz is a game to explain divisions to children. The rules are simple - Starting with the number one players count up by one. Is the number dividable by three, you are not allowed to say the number but say "Fizz". Is it dividable by five "Buzz" needs to be said and if it's dividable by both three and five "FizzBuzz" needs to be said.

A round would start like this: “One! Two! Fizz(3)! Four! Buzz(5)! Fizz(6)! Seven! Eight! Fizz(9)! Buzz(10)!”

Write a program, that prints the numbers from 1 to 100 by those rules. You don't have to print the numbers as words, the numerical representation is sufficient.

Java

Wiederholungen

Aufgabe 5.2 - Das kleine Einmal-Eins

Schreibe mit Hilfe von Schleifen das kleine 1 x 1 auf der Konsole aus. Deine Ausgabe sollte in etwa folgendermaßen aussehen:

```
1er Reihe:
1 * 1 = 1
2 * 1 = 2
...
10 * 1 = 10
2er Reihe:
1 * 2 = 2
…
9 * 10 = 90
10 * 10 = 100
```

Java

Repetitions

Exercise 5.2 - Multiplications

Use loops to print the 1 x 1 to the console. Your result should look something like this:

```
times 1:
1 * 1 = 1
2 * 1 = 2
...
10 * 1 = 10
times 2:
1 * 2 = 2
…
9 * 10 = 90
10 * 10 = 100
```

Java

Schleifen

Aufgabe 5.3 - Caesar Chiffre

Wir wissen ja, dass jedes Zeichen in Java einen eindeutigen Code hat. Genauergesagt ist die numerische Repräsentation eines Characters/Zeichens der Unicode.

Die Caesar Chiffre ist ein simpler Verschlüsselungsalgorithmus bei dem alle Buchstaben um einen bestimmten offset X verschoben werden, erreichst du das Z solltest du wieder beim A starten, Sonderzeichen werden wir jetzt einmal auslassen.

Mit einer Caesar Chiffre von 6 würde aus dem Text:

```
Ein Charakter namens Caesar!
```

folgendes werden:

```
Kot Ingxgqzkx tgskty Igkygx!
```

wenn wir diesen Text nochmal um 6 verschlüsseln würde aus dem Text:

```
Quz Otmdmwfqd zmyqze Omqemd!
```

Schreib ein Programm dass von der Konsole einen beliebiglangen Text einliest, danach zufällig einen Schlüssel zwischen 1 und 26 wählt und den verschlüsselten String ausgibt.

siehe Schrödinger s.84

Java

Loops

Exercise 5.3 - Caesar Chiffre

We know that every letter in Java has a unique code. In detail the numerical representation of a character is the Unicode.

The caesar chiffre is a simple encryption algorithm, that rotates every letter with an offset X. When you reach the Z you should start again with the A for the purpose of this exercise. Moreover we will ignore all special characters for now.

With a Caesar Chiffre of 6 the following text

```
Ein Charakter namens Caesar!
```

would be encrypted to:

```
Kot Ingxgqzkx tgskty Igkygx!
```

when an encrypted again with a key of 6 we get:

```
Quz Otmdmwfqd zmyqze Omqemd!
```

Write a program that reads any text from the console, generates a key between 1 and 26 and prints the encrypted message.

see Schrödinger s.84

https://classroom.github.com/a/BQ80UoMO Berechnung des Maximums

Java

Arrays

Aufgabe 6.1 - Berrechnung des Maximums

Schreibe ein kleines Programm, welches eine Liste von Zahlen über die Konsole zahlenweise einliest. Mit 'q' solltest du die Eingabe beenden können. Errechne dir danach aus der eben eingelesenen Liste das Maximum und gib es auf der Konsole aus.

Java

Arrays

Exercise 6.1 - Maxima calculation

Provide a program that reads numbers from the console until it reads the letter 'q'. Calculate the maximum of all values and print it to the console.