CHALLENGE 1- WEEK 1

INSTRUCTIONS

- **E-mail with challeng link must be sent by:** May 12, 2023 at 6 p.m.
- **Last commit:** May 12, 2023 at 6 p.m.

- Each challenge must be in a repository on GitHub
- The repository must be set to “private”
- Allow instructors to view GitHub
- They must send an email with the repository link
- “To”: include instructors
- “C.C” (with copy): for Scrum Masters and Coordinators

Settings

- Java version 17

Contacts:

Instructor: Carlos Fernando Gonçalves (Java Spring Boot + AWS)

- E-mail: carlos.goncalves@compasso.com.br
- GitHub:
- GitHub E-mail:

Instructor: Cassiano Jovino Mandelli Stefani (Java Spring Boot + AWS)

- E-mail: cassiano.stefani@compasso.com.br
- GitHub: https://github.com/CjMS2103
- GitHub E-mail: cassiano_stefani@hotmail.com

Instructor: Juliane Valiati Maran (Java Spring Boot)

- E-mail: juliane.maran@compasso.com.br
- GitHub: https://github.com/JulianeMaran32
- GitHub E-mail: julianemaran@gmail.com

Instructor: Mateus Cardoso de Moraes (Java Spring Boot)

- E-mail: mateus.moraes@compasso.com.br
- GitHub: https://github.com/mateusCoder
- GitHub E-mail: mateus.moraes0507@gmail.com

Instructor: Mateus de Oliveira e Silveira (Java Spring Boot)

- E-mail: mateus.silveira@compasso.com.br
- GitHub: https://github.com/mosilveira
- GitHub E-mail: svrmateus@gmail.com

Instructor: Yasmin Wichinievski Zimermann (Java Spring Boot)

- E-mail: yasmin.wichinievski@compasso.com.br
- GitHub: https://github.com/yasminwz
- GitHub E-mail:

Scrum Master: Guilherme Moreira de Paiva

- E-mail: guilherme.paiva@compasso.com.br
- GitHub:
- GitHub E-mail:

Scrum Master: Vitor de Oliveira Mendes

- E-mail: vitor.mendes@compasso.com.br
- GitHub:
- GitHub E-mail:

Coordinator: Fabiane Telles Leite Maciel

- Email: fabiane.maciel@compasso.com.br

Coordinator: Liliv Hana Vasconcelos

- E-mail: liliv.vasconcelos@compasso.com.br

QUESTIONS

Question 1.

Write a Java program that aims to validate whether a given employee is entitled to a bonus. The program needs to receive the number of employees that will be registered and after that, the program must store the name and salary of each employee.

If the user receives up to R$ 1000.00 real, he will receive a 20% bonus. If you receive more than 1000 and up to 2000, you receive 10%. If it is above 2000, instead of a bonus, the employee will have a 10% discount. At the end of the program, there should be an output with the names of the employees, salary, bonus or deduction and net salary.

Obs.: Use array to store the data.

Example:

```
Employee: XPTO
Salary: R$ 1.200,00
Bonus: R$ 120,00
Liquid salary: R$ 1.320,00

Employee: XYZ
Salary: R$ 3.000,00
Discount: R$ 300,00
Liquid salary: R$ 2.700,00

Employee: ABC
Salary: R$ 700,00
Bonus: R$ 140,00
Liquid salary: R$ 840,00
```

Question 2 (OBI-2020)

Emoticons are symbols used to express the feeling of the person writing a message. Scott Fahlman, a professor at an American university, was the first to propose the use of the character sequences :-) and :-(, which 2 respectively became symbols for “amused” and “annoyed”.

Let's define the sentiment expressed in a message as:

- neutral: if the number of “fun” symbols is equal to the number of “upset” symbols
- fun: if the number of “fun” symbols is greater than the number of “annoying” symbols
- upset: if the number of “upset” symbols is greater than the number of “fun” symbols.

Given a message consisting of a string of characters, write a program to determine the sentiment expressed in the message.

Your program must use the Scanner class to read an input line and return the expressed sentiment from that line.

Attention: Input validations is a point to pay close attention to

Example:

Entrada	Saída
`I hope everything is alright :-)`	`fun`
`I found the film very entertaining.`	`neutral`
`:-) :-( :-( :-)`	`neutral`
`I dreamed about the test :-( ( I'm going to study)`	`upset`

Question 3.

Write a Java program that aims to be a question and answer quiz. On the screen you will ask the user and he will answer.

If the user is right or wrong, you need to inform whether he was right or wrong and move on to the next question (while there is a next one).

Use array to store questions and answers.

The screen will ask the user the name, and at the same time the questions, if the user gets it right, it will demonstrate the errors and successes (as in the example below):

```
User XPTO
right: 6
wrong:4
```

Question 4. Script Postgres

Create an SQL script that creates two Person and Address tables.

The Person table must contain the following fields: person_id, name, age, phone, height, email, CPF, date of birth and id_address.

The Address table must contain the following fields: address_id, zip code, address, complement, neighborhood, city, state and country.

When the person is registered, he must be registered associated with an Address that already exists in the bank, which will have a foreign key id from the address table.

Person	**Address**
person_id	addresId
name	zip
age	street
phone	complement
height	neighborhood
email	city
cpf	state
birthday	country
address_id	

Question 5. Script MongoDB

Create a script for MongoDB that contains a Person Document, it must contain the following fields: object, name, age, phone, height, e-mail, CPF, date of birth and address that will be an object that must have the fields: zip code, street, neighborhood, complement, city, state and country.

Person	**Address**
Object	zip
name	street
age	complement
phone	neighborhood
height	city
email	state
cpf	country
birthday	
address	