FINAL Exam: OOP

WEP 2022

{3h00}

JUNE 2022

- All documents allowed
- Chatting and talking to other students are forbidden

EXERCICES	POINTS
EXERCICE 1	30
EXERCICE 2	70
TOTAL	**100**

You need to return on google classroom:

- ✓ **A ZIP file containing all necessary files**
- ✓ **The ZIP file should be named :**

<YOU FIRST NAME>-<YOU LAST NAME>.ZIP

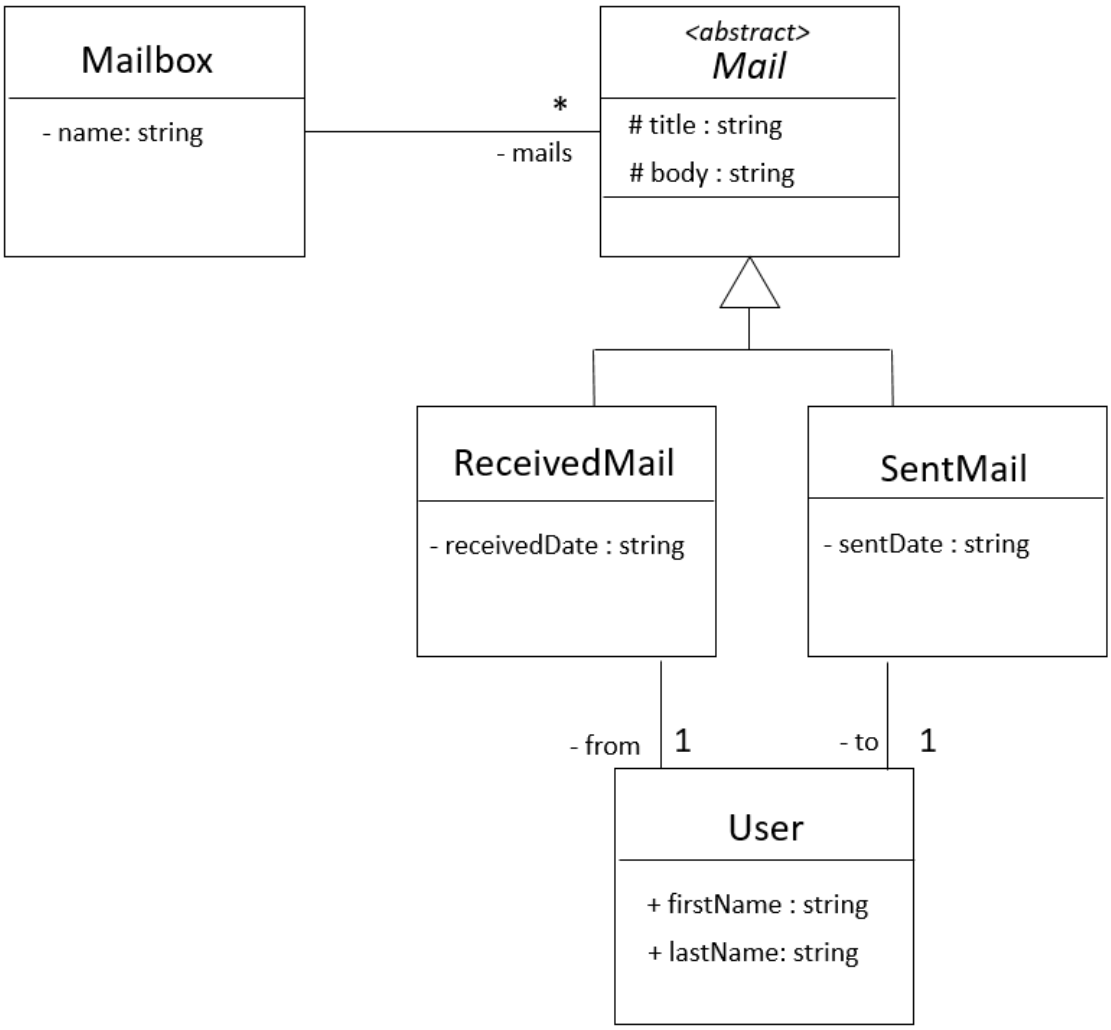
EXERCICE1 - 30 POINTS

Open the START CODE / EXERCICE-1

Convert the bellow diagram below to **TypeScript** code, including:

- Attributes (visibility, type)
- Inheritances
- Constructors

This model represents a mailbox, containing sent and received mails.



EXERCICE2 - 70 POINTS

We want to create an application to manage a **library** (why not!)

A **library** has a name and an address
An **address** has a city and street

Example of library

name	PNC Library
address	St. 371 - PHNOM PENH

A **library** manages many books
Each **book** has a tittle, an author, a price and can have many categories:

title	Public Private, oh yeah
author	Ronan
price	9999
categories	OOP, SONG, CRAZY

title	I like the constructor
author	Him
price	3
categories	OOP, IT, TYPESCRIPT

The library can welcome customers.
A customer has a first name and a last name

Examples

First name	Ronan
Last name	The King of OOP

A customer can withdraw (*ខ្ចី in Khmer*) book.
When a customer withdraw a **book**, we want to know:

- The withdrawn book
- The customer
- The date of the withdraw

Example of book withdraw:

Book	[Public Private, oh yeah]
Customer	Kea
Date withdraw	JUNE 2022

Q1 – 10 POINTS

On paper, write the **UML diagram** corresponding to your solution

Tips:

- ✓ Your diagram should contain the following classes:
  - Address
  - Date
  - Library
  - Book
  - Customer
  - Withdraw
  - Category
- ✓ Use the correct UML syntax (visibility, multiplicity …)

Q2 – 05 POINTS

Open the START CODE / EXERCICE-2

- ✓ Write **classes attributes and constructors** corresponding to your solution for this problem

Q3 – 05 POINTS

Write **methods** to:

- ✓ Add a customer to the library
- ✓ Add a category to a book

Q4 – 10 POINTS

2 books are the same, if they have the same title and same author.

- ✓ In the class book, define the **isEqual** method

- ✓ In the library, write a method to add a book to the library (if a same book does not exist already in the library)

Q5 – 10 POINTS

Write a **method** to withdraw a **book** from the library, for given **customer**, at given **date**.

- Create and add a withdraw item to the list of withdraws
- Return true if operation is successful *(for now it's always true)*

```
withdrawBook(book: Book, customer: Customer, date: Date) : boolean
```

Write a **method** to return a **book** to the library.

- Find the withdraw item related to this book, and remove it from the list.

```
returnBook(book: Book)
```

Q5 – 10 POINTS

Now we can withdraw a book **only if this book is not already withdrawn** by another customer
Write a **method** to check if a book is available

```
isAvailable(book: Book): boolean
```

Update the **method** withdraw, to prevent the operation when the book is not available

```
withdraw (book: Book) : boolean
```

Q6 – 10 POINTS

Write a **method** to get all books available in the library (not withdrawn)

```
      getAvailableBooks(): Book[]
```

Q7 – 10 POINTS

Write a **method** to get all books related to a category

```
      getBooksFor(categogy: Category): Book[]
```