

UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE



Topic: Exceptions Inspections Grupal

Assignments N°13

Object-oriented programming

NRC: 9642

Team 6: Code Warriors

Members: Ediosn Verdesoto

Leonel Tipan

ING. Jorge Edison Lascano.

```

public void selecOption() {
    PurchaseInterface purchaseInterface = new PurchaseInterface();
    while (true) {

        showMenu();

        String input = scanner.next();
        int option;
        try{
            option= option = Integer.parseInt(s: input);
        }catch(NumberFormatException e){
            System.out.println(x: "Error: entrada invalida");
            scanner.nextLine();
            continue;
        }
    }
}

```

Exception to control the option.

```

public double nextDouble() {
    while (true) {
        try {
            double num = scanner.nextDouble();
            if(num < 0) {
                throw new IllegalArgumentException();
            }
            return num;
        } catch (InputMismatchException e) {
            System.out.println(x: "Por favor, ingrese un numero decimal.");
            scanner.nextLine();
        } catch (IllegalArgumentException e) {
            System.out.println(x: "No se aceptan numeros negativos. Por favor, ingrese un número positivo.");
        }
    }
}

public int nextInt(){
    while (true) {
        try {
            int num = scanner.nextInt();
            if(num < 0) {
                throw new IllegalArgumentException();
            }
            return num;
        } catch (InputMismatchException e) {
            System.out.println(x: "Por favor, ingrese un numero entero.");
            scanner.nextLine();
        } catch (IllegalArgumentException e) {
            System.out.println(x: "No se aceptan numeros negativos. Por favor, ingrese un numero positivo.");
        }
    }
}
}

```

Exceptions to take just positive numbers.

```

public String readData() {
    try (BufferedReader reader = new BufferedReader(new FileReader(path + filename + ".json"))) {
        String content = "";
        String line;
        while ((line = reader.readLine()) != null) {
            content += line;
        }
        return content;
    } catch (IOException e) {
        System.out.println(x: "Hubo un error al leer el archivo");
        return "";
    }
}

```

Exception to read a .json file if it's any created.

```

public void createInventoryFile() {
    try {
        File file = new File(path + fileName + ".json");

        if(file.exists()){
        }
        else if(file.createNewFile()){
            System.out.println("El archivo se creo correctamente");
        }
    }
    catch (IOException ex) {
        System.out.println("Hubo un error al crear el archivo");
    }
}

```

Exception to control the creation of an .json file when there isn't any.

```

}
public void writeIfInventoryFileDoesntExists(Inventory inventory) {
    try {
        BufferedWriter writer = new BufferedWriter(new FileWriter(path + fileName + ".json"));
        Gson gson = new Gson();

        writer.write(gson.toJson(inventory));
        writer.flush();
    }
    catch (IOException ex) {
        System.out.println("Hubo un error al escribir el archivo");
    }
}

```

Exception to controll if there is a .json file to write in.