	Carátula para entrega de prácticas
Facultad de Ingeniería	Ingeniería en Computación

Profesor(a): ______________________

Asignatura: Programación Orientada a Objetos

Grupo: 1

No de Práctica(s): 6

Integrante(s): 11700202-9

____

____

____

No. de Equipo de cómputo empleado: ______________________

Semestre: 2024-1

Fecha de entrega: 13 de octubre de 2023

Observaciones: ______________________

____

CALIFICACIÓN: _______

Index.

Practice 6 - Class organization.

Dinosaur Tamagotchi.

1.- Introduction.

This report has the objective of explaining the sixth practice of the Object oriented programming subject, in which we must explore the organization for the classes in .

We must build an algorithm for simulating a dinosaur game, similar to tamagotchi with the options: Create, feed, change weather, face and fight.

1.1.- Hypothesis

We have to create only one class and make two of their object instances interact in the main class. I think we only need to program menus and analyze possible user errors inputting the data.

2.- Development

The steps to follow were:

- Create the Dinosaur class.
- Define the attributes and methods of Dinosaur.
- Define the interactions between the instances and the user.

2.1.- Dinosaur class abstraction.

The practice gives us the methods and attributes of the class, we only need to program them.

We need four attributes:

- Name (String).
- Type (Boolean)
- Habitat (Integer).
- Species (Integer)
- Energy (Integer)

We define the next setters:

```
public void setName(String name){this.name = name;}

    public void setType(boolean type){this.type = type;}

    public void setHabitat(int habitat){this.habitat = habitat;}

    public void setSpecies(int species){this.species = species;}
```

Builder: By default.

Public methods:

face (void): Faces two dinosaurs, validating if they can fight or not, and then generate a random number to make them fight or not .

eat (boolean): Check if the dinosaur is full of energy, otherwise sets its energy to 100.

fight (void): Generates random numbers for defining the attack power of each dinosaur, checking if someone (Energy = 0) faints for ending the battle.

changeWeather (void): Randomly changes the weather for the dinosaurs, lowing its energy if they are not in the correct habitat and checking if the dinosaur faints in the process.

Class Dinosaur
- Name (String). - Type (Boolean) - Habitat (Integer). - Species (Integer) - Energy (Integer)
- face (void) - eat (boolean)

- fight (void)
- changeWeather (void)

2.2.- Main class.

Initializes the two dinosaurs with the default builder, and sets its values when the user chooses the “Create dinosaur” options, also creates a dinosaur counter for avoiding exceeding the limit of two dinosaurs.

Then displays a menu with these options:

1: Create dinosaur: Checks the amount of dinosaurs, if there are slots available, allows the user to set the values of the dinosaurs.

2: Feed: Checks if there are dinosaurs, if not, exit the option, otherwise, asks the user for which dinosaur should feed, and then calls the feed method.

3: Change weather: Changes the weather of both dinosaurs.

4: Face: Calls the face method for the first dinosaur if there are already two dinosaurs.

5: Fight: Calls the fight method for the first dinosaur if there are already two dinosaurs.

6: Exit.

Pseudocode of the main class:

```
Create a scanner named "consomelog" to read input from the user
Initialize an integer variable "dinosaurCount" to 0
Create a Dinosaur object named "D1"
Create a Dinosaur object named "D2"
Initialize a boolean variable "out" to true
Initialize an integer variable "choice" to 0

Display "Bienvenido al Tamagotchi de los Dinosaurios! :D"

While "out" is true:
    Display "Menu principal:"
    Display "1 - Crear dinosaurio"
    Display "2 - Comer"
    Display "3 - Cambiar Clima"
    Display "4 - Afrontar"
    Display "5 - Pelear"
    Display "6 - Salir"

    Read the user's choice into the "choice" variable

    Switch "choice":
        Case 1:
            If "dinosaurCount" is 0:
                Initialize variables "name," "tch," "type," "habitat," and "species"
                Display "Ingresa el nombre de tu dinosaurio:"
                Read the dinosaur's name into "name"
                Set "D1" name to "name"
                Display "Ingresa el tipo del dinosaurio"
                Display "0 - Carnivoro"
                Display "Cualquier otra cosa - Herbivoro"
                Read "tch" from the user
                If "tch" is 0:
                    Set "type" to false
                Else:
                    Set "type" to true
                Display "Ingresa el habitat del dinosaurio"
                Display "1 - Caluroso"
                Display "2 - Frio"
                Display "3 - Templado"
                Read "habitat" from the user
                If "habitat" is less than or equal to 0 or greater than 3:
                    Set "habitat" to 3
                Set "D1" habitat to "habitat"
                Display "Ingresa la especie del dinosaurio"
                Display "1 - Volador"
                Display "2 - Acuatico"
                Display "3 - Terrestre"
                Read "species" from the user
                If "species" is less than or equal to 0 or greater than 3:
                    Set "species" to 3
                Set "D1" species to "species"
```

```
    Increment "dinosaurCount" by 1
Else if "dinosaurCount" is 1:
    Initialize variables "name," "tch," "type," "habitat," and "species"
    Display "Ingresa el nombre de tu dinosaurio:"
    Read the dinosaur's name into "name"
    Set "D2" name to "name"
    Display "Ingresa el tipo del dinosaurio"
    Display "0 - Carnivoro"
    Display "Cualquier otra cosa - Herbivoro"
    Read "tch" from the user
    If "tch" is 0:
        Set "type" to false
    Else:
        Set "type" to true
    Display "Ingresa el habitat del dinosaurio"
    Display "1 - Caluroso"
    Display "2 - Frio"
    Display "3 - Templado"
    Read "habitat" from the user
    If "habitat" is less than or equal to 0 or greater than 3:
        Set "habitat" to 3
    Set "D2" habitat to "habitat"
    Display "Ingresa la especie del dinosaurio"
```

```
        Display "1 - Volador"
        Display "2 - Acuatico"
        Display "3 - Terrestre"
        Read "species" from the user
        If "species" is less than or equal to 0 or greater than 3:
            Set "species" to 3
        Set "D2" species to "species"
        Increment "dinosaurCount" by 1
    Else:
        Display "Limite de dinosaurios excedido"
Case 2:
    If "dinosaurCount" is 0:
        Display "No hay dinosaurios disponibles"
    Else:
        Initialize an integer variable "eatChoice"
        Display "Que dinosaurio quieres que coma"
        Display "1 o 2"
        Read "eatChoice" from the user
        If "eatChoice" is greater than or equal to 2 and "dinosaurCount" is 1:
            Display "El dinosaurio 2 no existe"
        Else if "eatChoice" is 1:
            Call the "eat" method on "D1"
        Else:
            Call the "eat" method on "D2"
        Case 3:
            If "dinosaurCount" is 0:
                Display "No hay dinosaurios disponibles"
            Else if "dinosaurCount" is 1:
                Call the "changeWeather" method on "D1"
            Else:
                Call the "changeWeather" method on "D1"
                Call the "changeWeather" method on "D2"
        Case 4:
            If "dinosaurCount" is less than 2:
                Display "No hay dinosaurios para pelear"
            Else:
                Call the "face" method on "D1" with "D2" as the parameter
        Case 5:
            Call the "fight" method on "D1" with "D2" as the parameter
        Default:
            Display "Adios! :D"
            Set "out" to false

Close the "consomelog" scanner
```

Conclusion: The hypothesis was correct, but writing the menus was very tedious, because there were multiple possibilities of user wrong inputs, but I considered we handled them in a very efficient way.

Handling the packages was tedious and confusing, but we learned a lot about packages and it's a useful way to manage a project.

Programming the game was very fun, like all the recent practices of the subject, we've enjoyed a lot the practice.

References:

- [1] Oracle. (2023, July). Java API, Math Class [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html