Batch No. B2 **Roll No. 160101211110**

Experiment / Assignment/ Tutorial No. 6

Grade : AA/AB/BB/BC/CC/CD/DD

Signature of the Staff In-charge with date

TITLE : Case Study (for Class Diagram)

AIM: Draw class Diagram for the chosen Case Study . Clearly show

- Attributes
- Multiplicities between classes
- Aggregations/compositions/Association between classes
- Generalization between classes in the class diagram.

And show the implementation of aggregation, association, composition and generalization between the classes.

Expected OUTCOME of Experiment:

CO1: Understand the features of object oriented programming compared with procedural approach with C++ and Java.

CO2: Explore arrays, vectors, classes and objects in C++ and Java.

CO3: Implement scenarios using object oriented concepts (Drawing class diagram, relationship between classes, sequence diagram)

CO4: Explore the interface, exceptions, multithreading, packages

Books/ Journals/ Websites referred:

1.Ralph Bravaco , Shai Simoson , “Java Programing From the Group Up” Tata McGraw-Hill.

2.Grady Booch, Object Oriented Analysis and Design .

Pre Lab/ Prior Concepts:

Define Class, Methods, Object.

Understanding of Aggregation, Association, Composition and Generalization between classes

Classes and Objects:
• A class is a user-defined data type with a template that serves to define its properties.
• Class that defines the state and behavior of the basic program components known as objects
• Classes provide a convenient method for packing together a group of logically related data items and functions that work on them.
• In Java, data items are called fields and the functions are called methods.
• Calling a specific method in an object is described as sending the object a message Class Diagram:
• Class diagrams are the main building block in object-oriented modeling.
• They are used to show the different objects in a system, their attributes, their operations and the relationships among them.

UML Relations:
• Relationships in UML are used to represent a connection between structural, behavioral, or grouping things.
1. Association
• It is a set of links that connect classes of the class diagram.
2. Aggregation
• Aggregation is identified by the phrase "is-part-of".
• Aggregation implies a relationship where the child can exist independently of the parent.
3. Composition
• It is a two-way association between the objects.
• Composition implies a relationship where the child cannot exist independent of the parent
4. Inheritance
• Inheritance is a mechanism which allows a class A to inherit members (data and functions) of a class B.

List Of Classes:

- Public class Pokemon
- Public class Properties
- Public class Charmander
- Public class Charmelon
- Public class Charizard
- Public class burn
- Public class FlameThrower
- Public class Fire
- Public class Flying

- Interface <<Type>>
- Interface <<Move>>

Identify Attributes for each class:

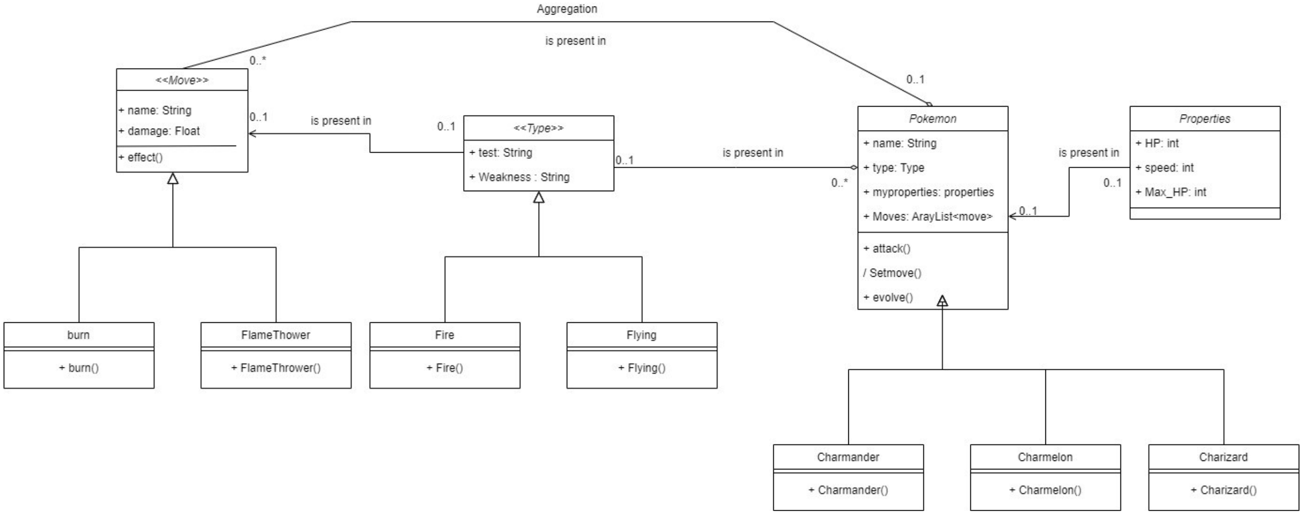
Class	Attributes
Pokemon	• Public name • Public type • Public myproperties • Public Moves
Properties	• Public HP • Public Speed • Public Max_HP
<<Move>>	• Public name • Public damage
<<Type>>	• Public test • Public Weakness

Identify List of Methods in each classes:

Class	Methods
Pokemon	• Public attack() • Derived Setmove() • Public evolve()
Charmander	• Public Charmander()
Charmelon	• Public Charmelon()
Charizard	• Public Charizard()
Flying	• Public Flying()
Fire	• Public Fire()
burn	• Public burn()
FlameThrower	• Public FalmeThrower()
<<Move>>	• Public effect()

Implementation details:

Draw a UML class diagram for representing the following elements from the problem domain for a fictional Pokemon game world. Pokemon class has a name, type, properties and moves. Its methods are attack, Setmove and evolve. Every pokemon inherits properties class. All pokemons have 1 type and 1 move class.



https://github.com/Aatmaj-Zephyr/Pallet-town-Object-modelling

Conclusion

We implemented a Pokemon game world by using the problem statement provided to us. We made a UML Class diagram that abides by all the UML properties and depicts a clear relation between the classes and their associations

Date: 04/11/2022 **Signature of faculty in-charge**

<u>Post Lab Descriptive Questions</u>

1. Consider the following class:

```
public class TypeOfVariable{
        public static int a;
        int b,c;
        public void printValue(){
                int x = 10;
        }
        public static void main(String args[]){
                TypeOfVariable object=new TypeOfVariable();
                object.printValue();
        }
}
```

a). What are the class/static variables?
Ans: a

b). What are the instance variables?
Ans: b and c

c.)What are local variables?
Ans: x

2.What is the output from the following code:

```
public class Test
{
   static int x = 11;
   private int y = 33;
   public void method1(int x)
   {
      Test t = new Test();
      this.x = 22;
```

```
        y = 44;

        System.out.println("Test.x: " + Test.x);
        System.out.println("t.x: " + t.x);
        System.out.println("t.y: " + t.y);
        System.out.println("y: " + y);
    }

    public static void main(String args[])
    {
        Test t = new Test();
        t.method1(5);
    }
}
```

Output:

```
Test.x: 22
t.x: 22
t.y: 33
y: 44
```