****

**OBJECT-ORIENTED PARADIGM: INHERITANCE**

**Worksheet 2.1**

**Unscramble each word from the list below and then use them to complete their meaning.**

1. Ecntahierin:Inheritance----Herencia
2. Sintiher:Inherits--Heredar
3. Dfee:Feed--Alimentar
4. Aly sgeg: Lay eggs—poner huevos
5. Inps wbes: Spin webs—Hacer redes
6. Haebvirou: Behaviour--Comportamiento
7. Poh:Hop—Saltar en un solo pie
8. Lishterh: Slither--Deslizarce
9. Poen-ededah wsorra: Open headed-arrows—Flecha con la cabeza abierta

**Meaning of the vocabulary words**

1. \_\_\_\_inheritance\_\_\_\_\_\_\_\_\_\_\_ the use of code that is pre-written or created previously.
2. \_\_\_\_\_\_open headed arrows\_\_\_\_\_\_\_\_\_ a type class used in programming to describe computations in a pure and declarative fashion.
3. \_\_\_\_\_inherit\_\_\_\_\_\_\_\_\_\_ is a way to add functionality.
4. \_\_\_\_\_\_\_\_slither\_\_\_\_\_\_\_ to move easily and quickly across a surface.
5. \_\_\_\_\_\_feed\_\_\_\_\_\_\_\_\_ describe the process of inserting an object into a device or moving data into another location.
6. \_\_spin webs\_\_\_\_\_\_\_\_\_\_\_\_\_ is an intermediate connection in a string of connections linking two devices.
7. \_\_\_\_\_\_lay eggs\_\_\_\_\_\_\_\_\_ to produce an output or result.
8. \_\_\_\_\_\_\_hop\_\_\_\_\_\_\_\_ is a globally targeted programming technique.
9. \_\_\_\_\_behavior\_\_\_\_\_\_\_\_\_\_ is a construct of our minds about what we want the system to do.

**Worksheet 2.2**

**Object-Oriented Paradigm: Inheritance**

**OOP design: inheritance**

Inheritance makes it possible to create a new child class that inherits the attributes and methods of the original (parent) class. The term 'parent class' is also referred to as superclass or base class, while 'child classes' can also be referred to as subclasses or derived classes.

Inheritance can be used when you observe a 'kind-of' or 'is-a' relationship between classes. You can say, for example, that "a dog is an animal" or "a dog is a kind of animal".

**Inheritance diagrams**

A classic example of inheritance involves an animal superclass.

Imagine that you are going to develop a game for children that allows them to play the role of a zookeeper. The zoo will have many different animals, and you could make a long list of every different animal type and develop a class definition for each. However, you would soon notice that every class has a set of attributes and methods in common. For example, all of the animals would have a name and would be able to feed and move.

You could use a single animal class and add an animal-type attribute that would specify the type of animal. But different types of animals will need different methods (for example, birds need to be able to lay eggs and spiders to spin webs).

Even where the animals have a method in common, such as move, the implementation would need lots of 'if' statements to ensure the correct movement behaviour (for example, kangaroos hop, whereas snakes slither). Inheritance can solve this problem.

Inheritance is an OOP technique where child classes are derived from a parent class and inherit all of the attributes and methods from the parent. These properties don't need to be redefined for the child (the child shares the code from the parent class). However,

you can add extra attributes and methods and change the behaviour of any inherited methods as required.

In a UML diagram, inheritance is represented with open-headed arrows. The arrows point up from the child class to the parent class. The figure below shows a high-level inheritance diagram for the zoo game.

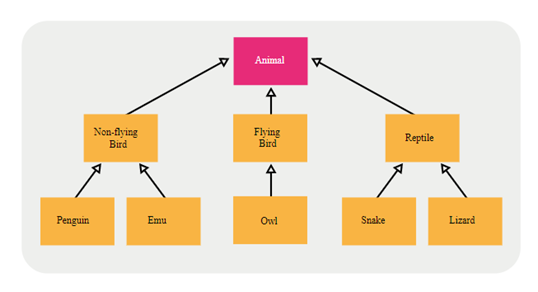


Figure 1: Inheritance diagram for the zoo game

This high-level diagram shows us the names of the classes and the relationship between them. You can see that a penguin is a non-flying bird and that a snake is a reptile. It is important to note that this is not a biological taxonomy; the classes and relationships are those relevant for the zoo game

**Worksheet 2.3**

**WRAP-UP**

After reading the text **“Object-Oriented Paradigm: Inheritance”** you will have to create a UML diagram, similar to the one in the text, but using a different class. For example, it can be “Transport”. At the end, you will include the diagram to this worksheet and you will also share it on the Whatsapp group so your classmates can see yours and you can see theirs.

Vehicle

Motor

Doors

Wheel

speedUp()

brake()

Bus

Truck

Car

**Worksheet 2.4**

**SELF-EVALUATION**

**Ask yourself the following questions**

1. Entiendo qué es Inheritance.

Yes **😃** Maybe **😐** No **😟**

1. Entiendo cómo se aplica el concepto de Inheritance a situaciones de la vida real.

Yes **😃** Maybe **😐** No **😟**

1. Entiendo cómo se usa la estrategia text-to-world.

Yes **😃** Maybe **😐** No **😟**