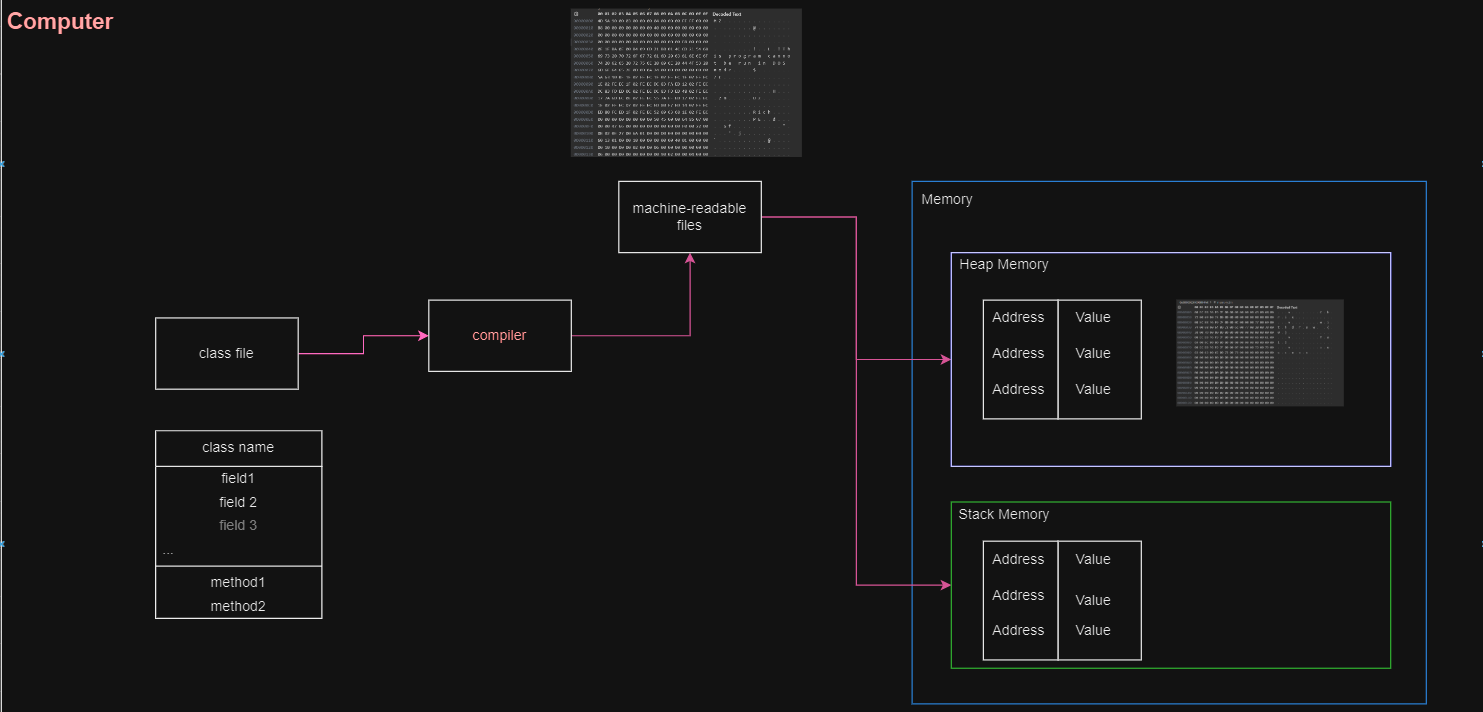
# How computer react when create an object



(Figure 1)

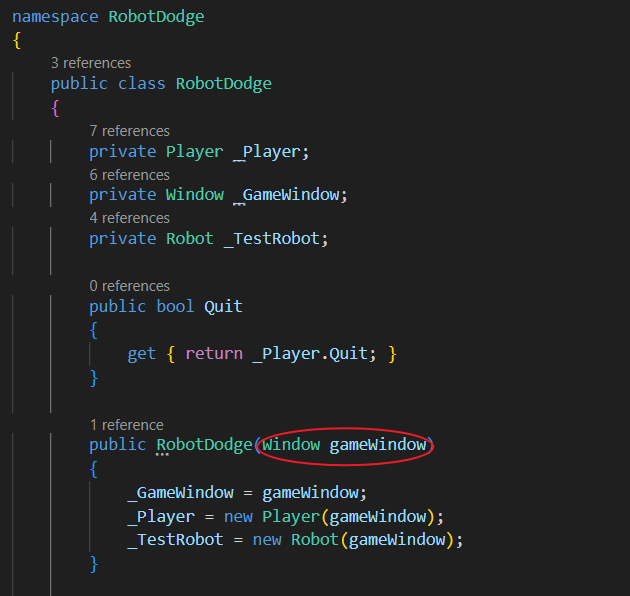
## Class & Object

As we can see from Figure 1, the essence of a class is a file that can be compiled into machine-readable code. And Object is a reference to an address allocated in memory.

We can create an object by a class file.

## Constructor & method & filed & property

### constructor



(Figure 2)

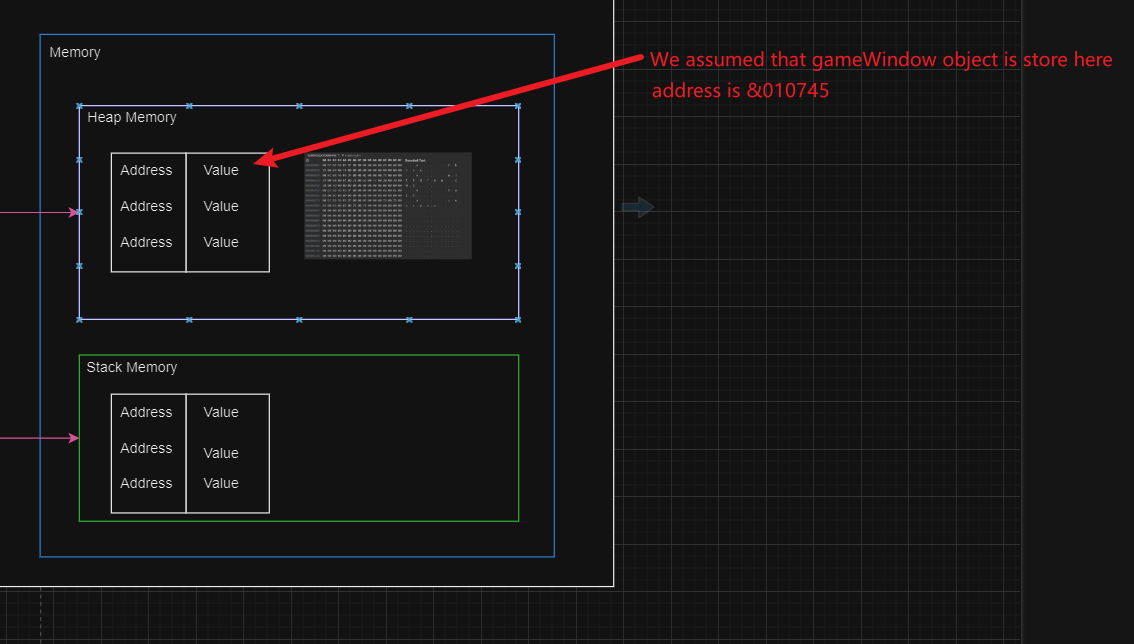
Constructor plays an important role in creating objects. When we want to assign certain properties to an object when it is initialized, we can pass the object through the constructor.

From Figure 2, a Window object **gameWindow** is passed to the constructor of **RoboDodge**， then **Robotdodge** object can be created with the filed **\_GameWindow** whose reference is from **gameWindow**.

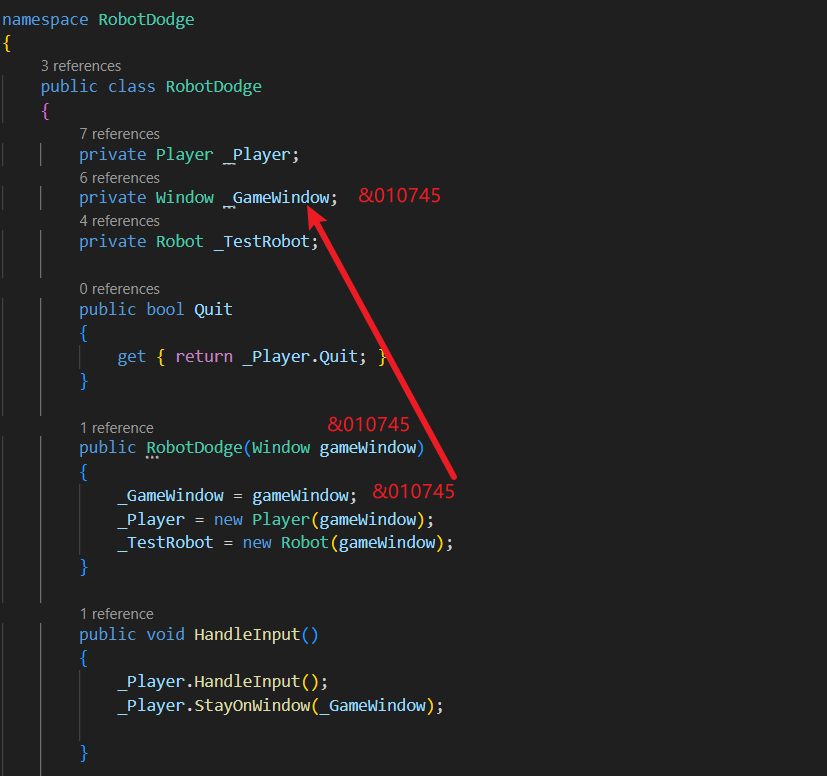
For better understanding:

Figure 3, we assumed that **gameWindow** object is stored in **&010745**;

Figure 4, **RoboDodge** object’s field **\_GameWindow** has been assigned the address **&010745**.



(Figure 3)



(Figure 4)

### Method

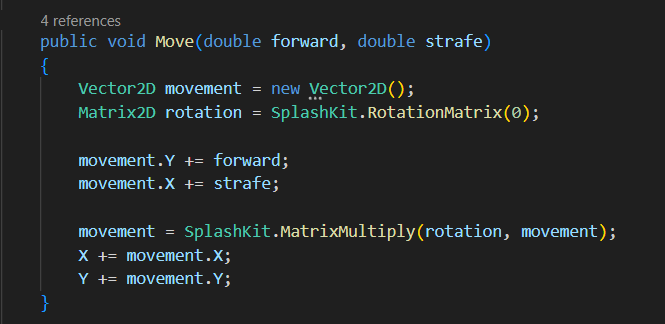
### Methods are used to describe the actions of an object and can correspond to the actions of objects in the real world.

### When the program executes a method, the method will be loaded into the call stack, and the local variables created in the method will be recycled after the method call ends.

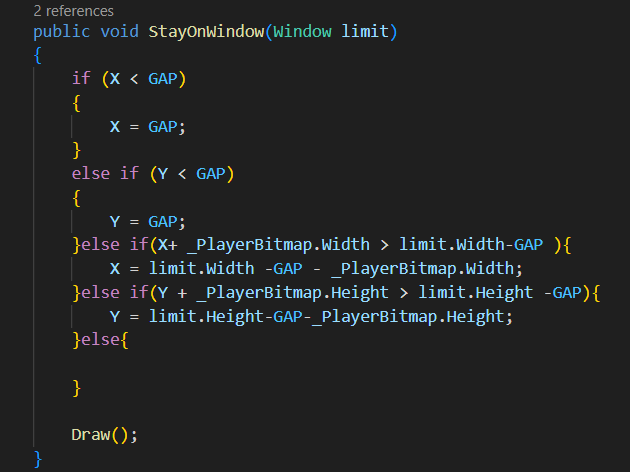
### Methods accept two type of parameters, value types and reference types.

Figure 5: The value type passes a specific value

Figure 6: Reference types pass addresses



(Figure 5)

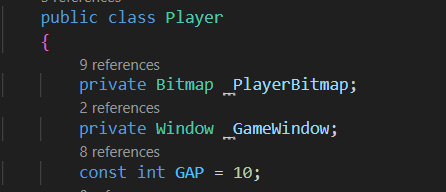


(Figure 6)

### Field

Fields are used to store data for objects.

Figure 7: **\_PlayerBitmap** and **\_GameWindow** is **Player’s** filed

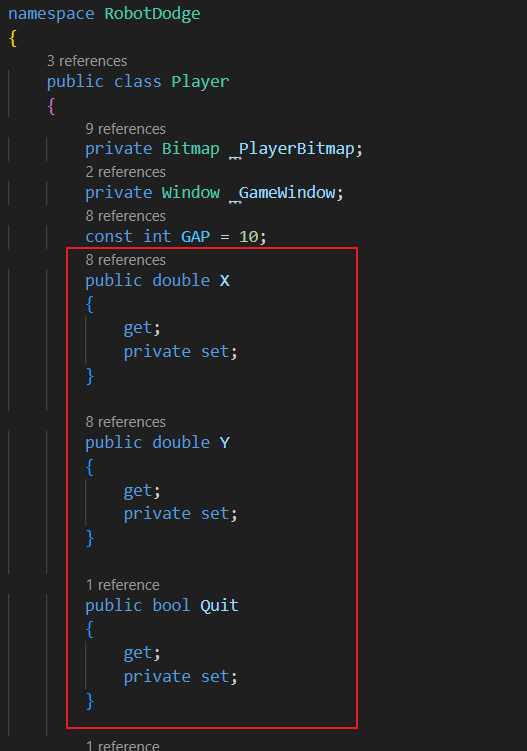


(Figure 7)

### Property

Properties provide a controlled access to fields.

Figure 8: **X** is a read only property, which allows other objects get its value.



(Figure 8)

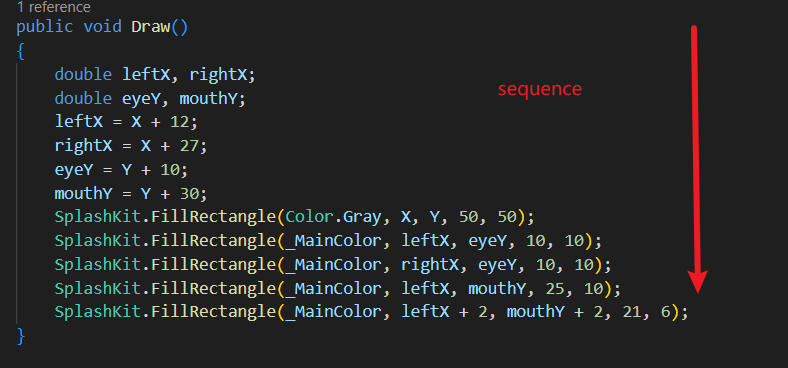
## Control flow

Control flow help us to implement our logic.

### Sequence

The program will execute the code step by step;

Figure 9: The program will be executed in order



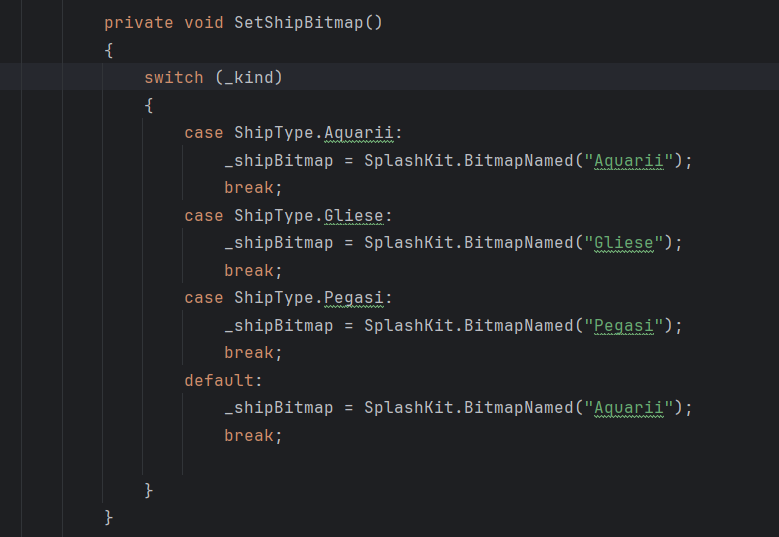
(Figure 9)

### Selection

Selection is a logical filter that we can use to limit the execution of branches only when the condition I met; Usually, we use if to implement selection, but we can also use switch to implement selection.



(Figure 10)



(Figure 11)

### Repetition

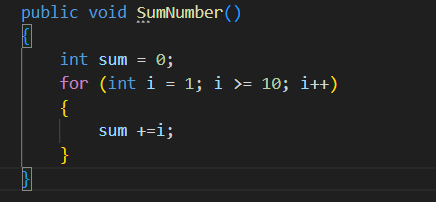
Loop is used to repeat a path a number of times and repeat code a variable a number of times.

Usually, there are two types of repetition, while and for. And the while loop also have two types, pre-test loops and post-test loops.

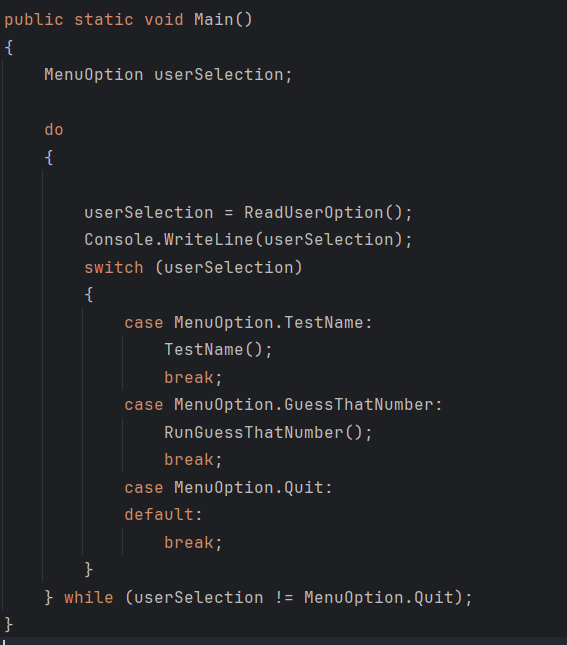
Figure 12: for loop can help you easily to repeat a path a number of times.

Figure 13: do while loop runs at least once.

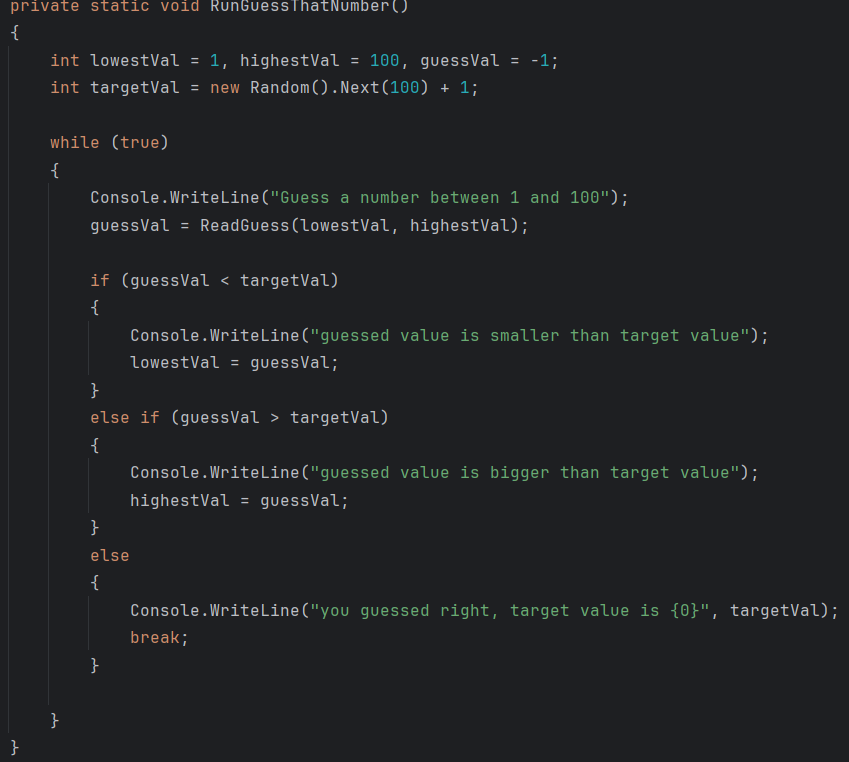
Figure 14: while loops runs 0 to many times.



(Figure 12)



(Figure 13)



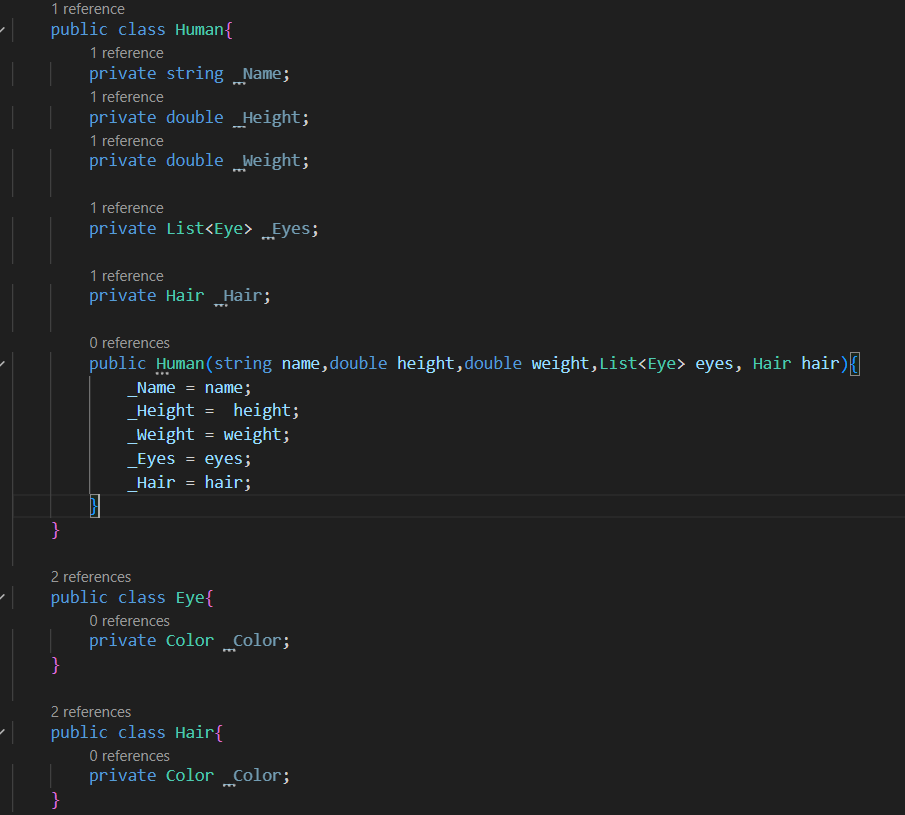
(Figure 14)

## Simulation

We can take a simulation to create a human by code.

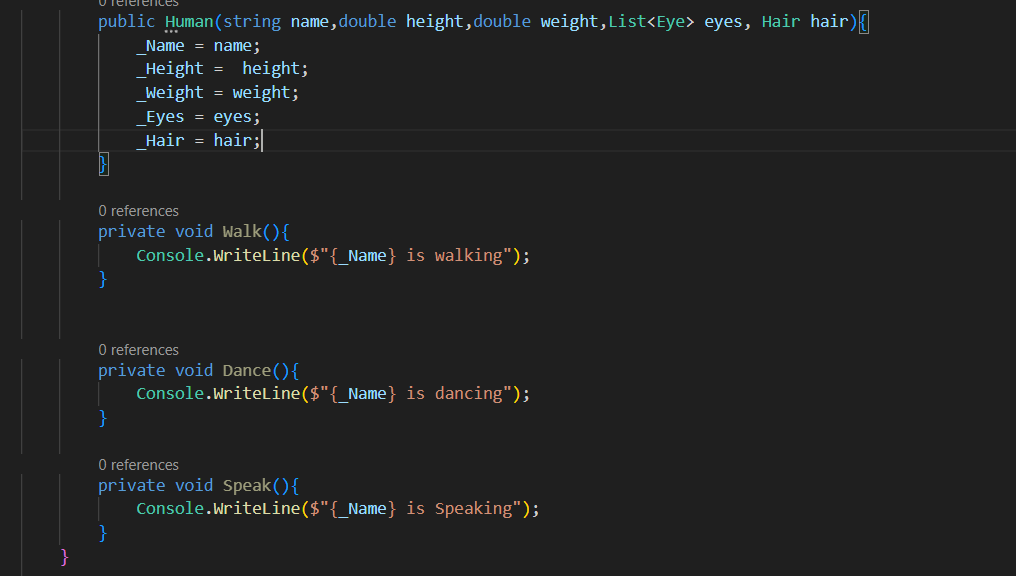
First of all, we need to know that when creating a specific person, we should not use the specific person as a coding template, but abstract the specific person into a concept and use it as a template. Because, if we create a code template for a specific person, then the code will have low reusability, but if we create a template for a concept, then the reusability of the code will be very high. Why emphasize reusability? Because computer resources are precious and should be cherished.

As we all know, humans have height, weight, eyes and hair, etc. so we can create a class as shown in Figure 15, add a constructor, and when we create a specific person, we can provide the necessary fields for this person.



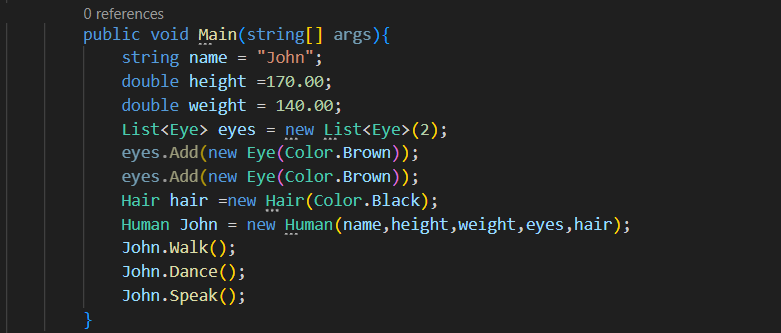
(Figure 15)

Human can walk, dance, and speak, etc.



(Figure 16)

Now, we can build a human to do somethings.



(Figure 17)

Console will output this.

