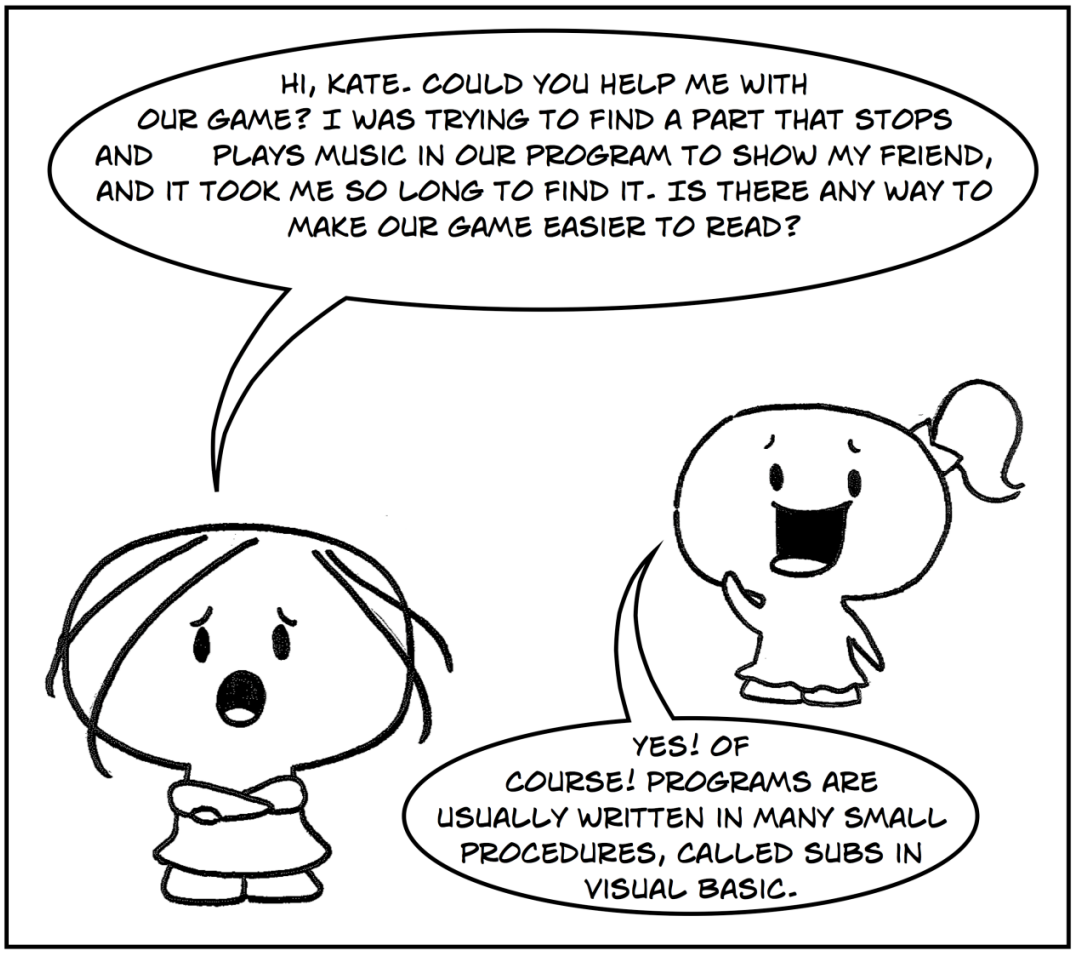
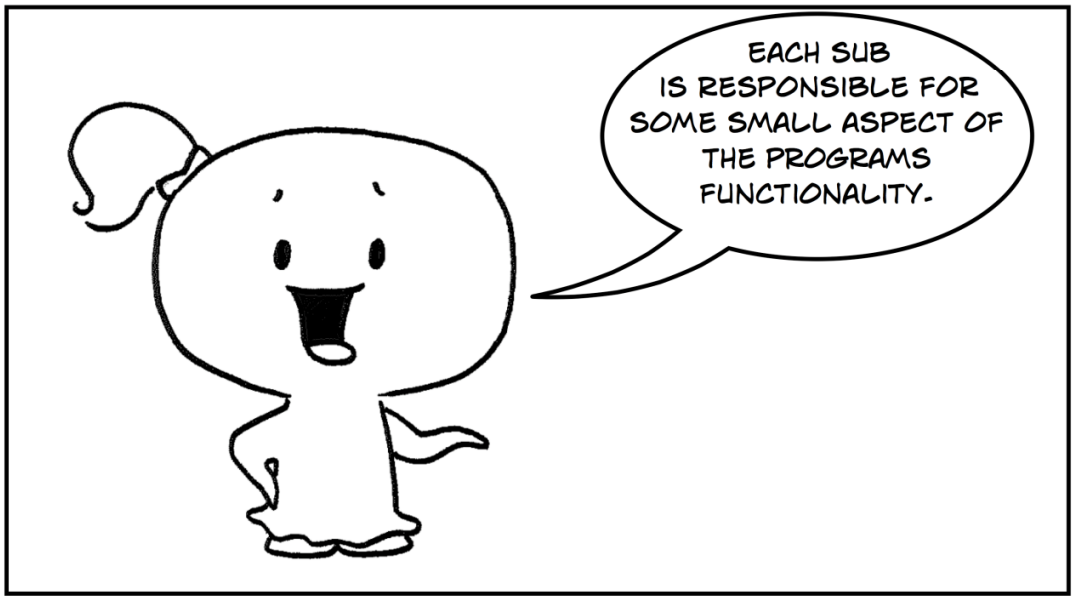
Chapter 6

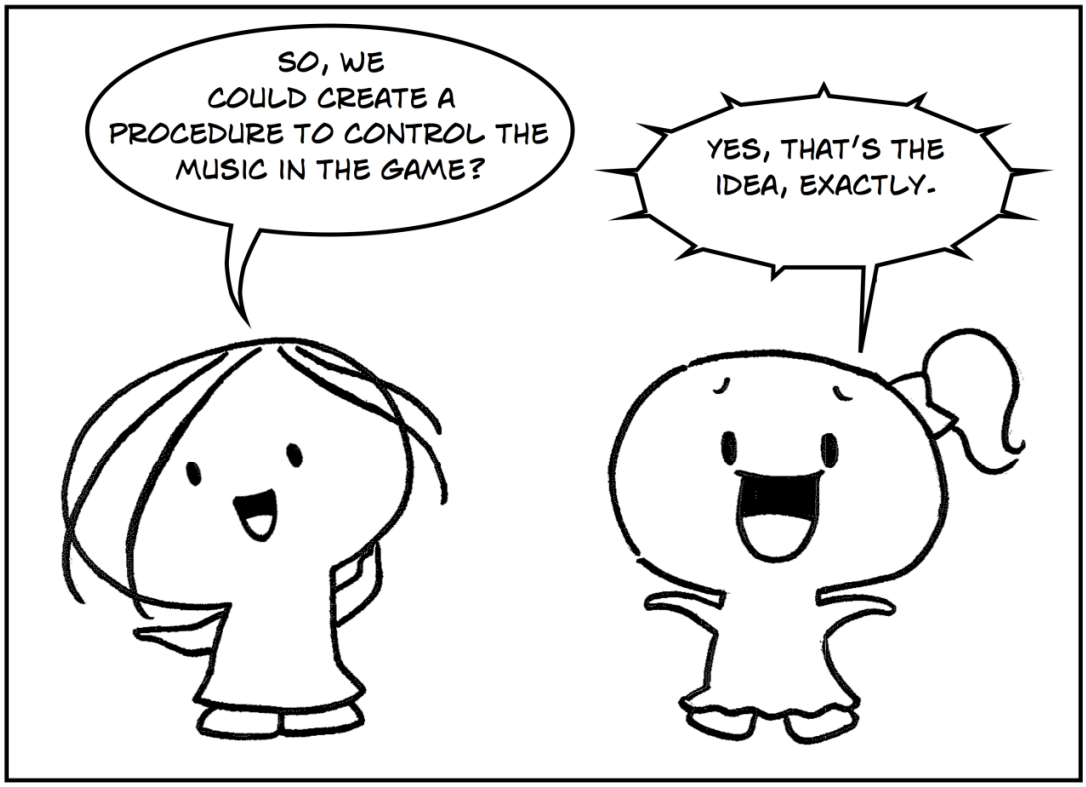
"Methods in VB .NET"

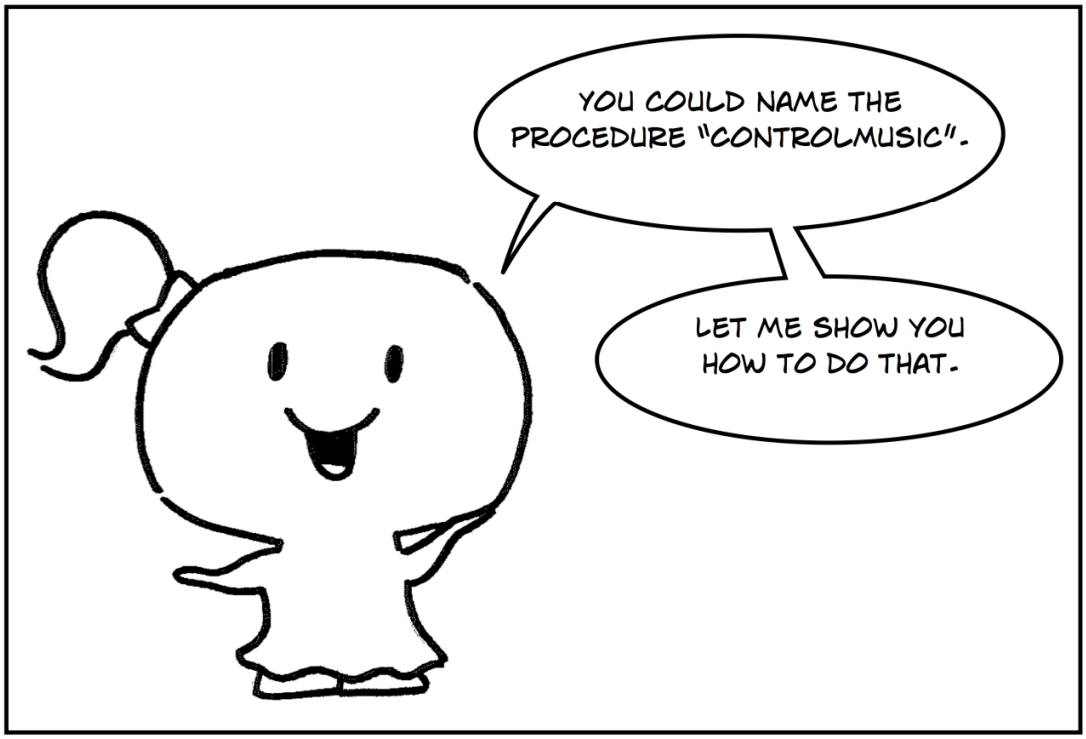
Summary:

I this chapter you will need to use the solution from the previous one. You will learn about methods in VB, their purpose, how to create and use them in your program. You will not need any additional material for this chapter.









Part 1

A Method is a procedure built into the program. They are a series of statements that are executed when called. Methods allow us to handle code in a simple and organized fashion. There are two types of methods in VB .NET: those that return a value (Functions) and those that just perform an action (Sub Procedures). Sub procedures (or just Subs) are methods which do not return a value. Each time when the Sub procedure is called the statements within it are executed until the matching End Sub is encountered. Sub Main(), the starting point of the program itself is a sub procedure. When the application starts execution, control is transferred to Main Sub procedure automatically which is called by default. Figure 1 presents an example of Sub Main() Declaration:

```
Public Sub Main()
        'this method is called by Default
        'statements to execute
        ...
End Sub
```

Figure 1

A function is a method which returns a value. Functions are used to evaluate data, make calculations or to transform data. Declaring a Function is similar to declaring a Sub procedure. Functions are declared with the Function keyword. Figure 2 shows the example of function declaration.

```
Public Function CalculateAvarage() As Integer
        'needs to be called somwhere in yur program
        'returns an integer
        'statements to execute
        ...
 End Function
```

Figure 2

Each method can take some parameters to operate with. A parameter is an argument that is passed to the method by the method that calls it. Parameters are enclosed in parentheses after the method name in the method declaration. You must specify types for these parameters.

A method is not executed until it is called. A method is called by referencing it's name along with any required parameters. Figure 3 shows a method call with required parameters and without them:

```
Public Sub Main()
        'calls the sub that prints "Hello World!"
        HelloWorld()
        'declaring variables to use
        Dim a, b As Integer
        a = 5
        b = 2
        'using a funcion and passing variables into it
        a = CalculateAvarage(a, b)
End Sub
```

Figure 3

Variables declared within methods are called method variables. They have method scope which means that once the method is executed they are destroyed and their memory is reclaimed. For example, in Figure 4 a and b variables are declared inside the CalculateAvarage() Sub. Those two variables are accessible only within the method and not from outside the method.

```
Public Sub CalculateAvarage()
        'local variables a and b
        Dim a, b As Integer
        a = 5
        b = 2
        'statements to execute
        ...
End Sub
```

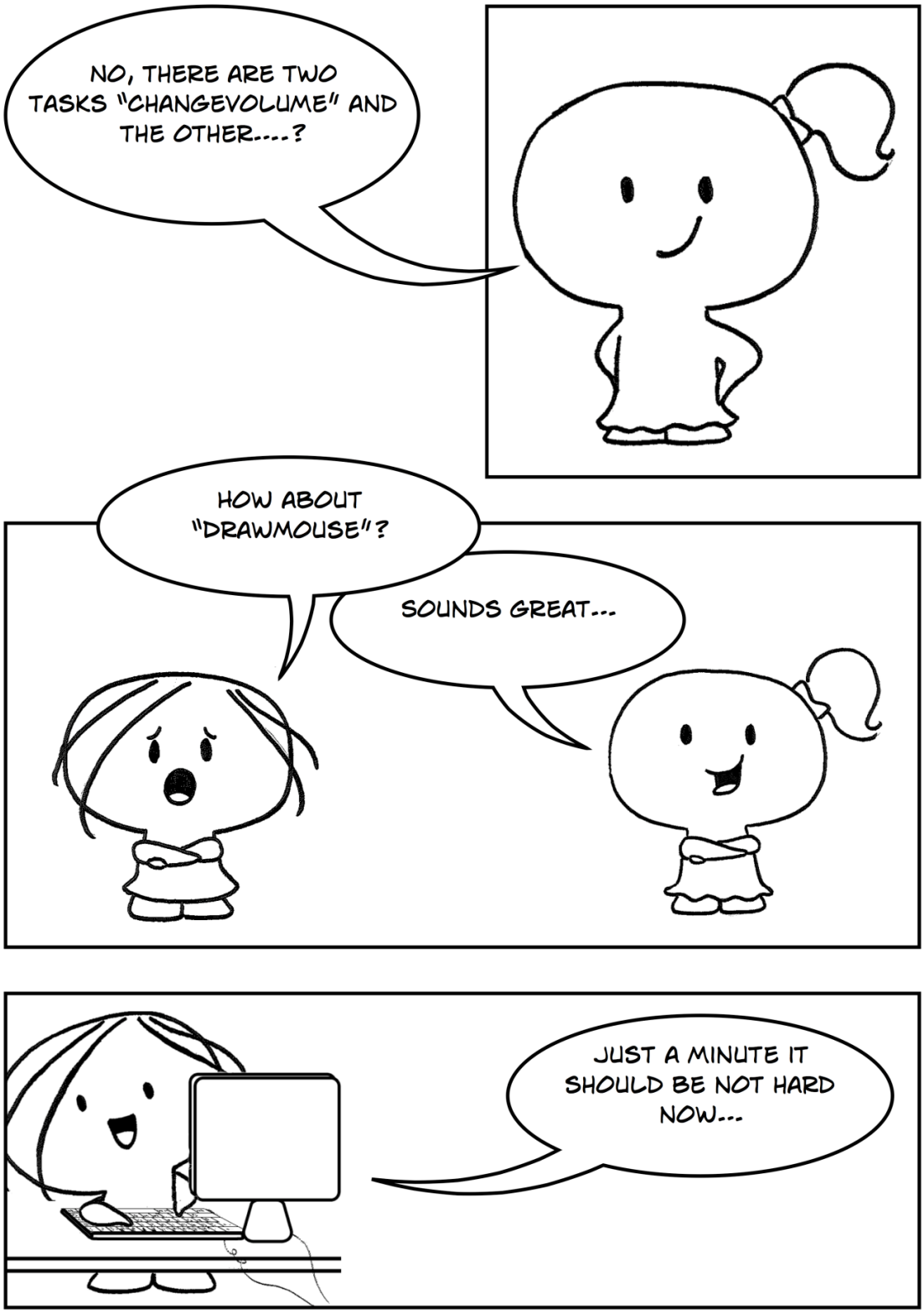
Figure 4

In our case to create public ControlMusic() sub right above the , we need to cut the code that is responsible for stopping and starting music from the Game Loop, and paste it inside the ControlMusic() Sub.

Exercise 1: Creating ControlMusic() Sub Procedure


Make the following changes in your program and write your solutions to the worksheet:

1. Create ControlMusic() Sub, which will stop and start the music during the game. Call this method inside the Game Loop.



Part 2

Once you have learned how to create subs, you can make our program much more readable by creating two new subs – ChangeVolume() which will be responsible for changing the volume up and down and DrawMouse() which will draw the target image instead of default mouse pointer.

To move code that is responsible for changing the volume, proceed the same steps as in part 1 for ControlMusic() sub. As for DrawMouse() sub, you will need to add a mousePoint variable as Point2D type.

Exercise 1: Making additional changes


Make the following changes in your program and write your solutions to the worksheet:

1. Create a new Sub Procedure called ChangeVolume() which will add and lower the volume of the music. Put this sub before the Main() method and call it inside the Game Loop.
2. Create a new Sub Procedure called DrawMouse() which will draw a target instead of the default mouse pointer. Call this sub inside the Game Loop.

NOTE: You will need to have `mousePoint = Input.GetMousePosition()` both in the Main procedure and in the DrawMouse procedure.