



Escola Superior de Tecnologia e Gestão  
Instituto Politécnico da Guarda

## Laboratorial Work

### Nº 2

#### Group: 3.1

Number	1012164	Name	Carina Tomé
Number	1012208	Name	Dário Ribeiro
Number	1012180	Name	Pedro Sanches
Number	1012396	Name	Rui Manta

Fill in the header record with the names and numbers of the group members and add “(missing)” next to the name, in case any member of the group missed the class. After finishing the laboratorial work, submit it via the form at <http://bit.ly/2TL0IKS>.

1. Identify and describe the 3 phases of the development of a program for your robot.

Phase 1: Edit- Programa o código java

Phase 2: Build –Compila, liga e envia o programa para o robot

Phase 3: Test – Testa o programa

2. Describe, in your own words, what the Compiler and the Linker does.

Compiler: Analisa e converte o código escrito em java para código binário.

Linker: É o que junta/liga o código que desenvolvemos com o código das bibliotecas que estamos a utilizar.

3. Relate the definitions in the right table with the file types in the left table. Use the letters.

A	MyBot.java
B	MyBot.class
C	MyBot\$.hex
D	MyBot.rjp

B	File generated by the Compiler.
D	File with the properties of the project.
A	File with the source code.
C	Executable file generated by the Linker.

4. Modify the program you did in class to introduce a new compilation error of your choice. Write down the error message you receive. Describe how the message relates to the actual error in your program.

R:

Código:

```
import com.ridgesoft.intellibrain.IntelliBrain;  
public class mybot {
```



```
private static String myname = "HELLO" ;  
public static void main(String args[]) {
```

```
    IntelliBrain.setTerminateOnStop(false); // Bloqueia a função de default do stop permitindo ao  
    user poder programar o botao
```

```
    System.out.println(myname);
```

```
    System.out.println("Waiting for STOP...");
```

```
    IntelliBrain.getStopButton().waitPressed(); // Ação de clicar no botao
```

```
    System.out.println("STOP PRESSED);
```

```
    }
```

```
}
```

Erro:

Compiling...

Found 1 lexical error in "C:/MyBot/mybot.java":

```
12. System.out.println("STOP PRESSED);
```

```
      ^-----^
```

\* Lexical Error: String constant not properly terminated.

Found 2 syntax errors in "C:/MyBot/mybot.java":

```
12. System.out.println("STOP PRESSED);
```

```
      ^-----^
```

\* Syntax: ")" inserted to complete MethodInvocation

```
12. System.out.println("STOP PRESSED);
```

```
      ^-----^
```

\* Syntax: ";" inserted to complete ExpressionStatement

Compilation failed

Descricao:

O erro que introduzimos, foi um erro de sintaxe, que consistiu em remover uma das aspas da funcao System.out.println.



Apos compilação , o compilador detetou 1 erro léxico e 2 erros de sintaxe

Na Parte lexical detetou a falta de um elemento que finaliza a função (String constant not properly terminated)

Na parte da sintaxe detetou 2 caracteres mal posicionados sendo eles o

5. Look for information about the Display class on the API JavaDoc. Present a modified version of the program you did in class, that uses the Display class to write on the LCD, instead of using the System.out.println instruction.

```
R: import com.ridgesoft.intellibrain.IntelliBrain;
import com.ridgesoft.io.Display;
public class IntelliBrainTwoLineDisplay {
public static void main(String args[]) {
Display display = IntelliBrain.getLcdDisplay();
display.print(0, "Write on");
display.print(1, "the LCD");
while(true);
}
}
```

6. Try some of the examples that are in the folder C:\Programas\RoboJDE\Examples\IntelliBrain  
They are examples about how to program the IntelliBrain hardware resources.

Try the examples about using:

- LEDs
- BUZZER
- Motor ports (be careful, the fan of the robot might move).
- Analog ports (make changes needed to use the analog sensor to detect white lines in the floor, connected to port 7 in your robot).

Analyze these examples and notice that the declaration of the objects to program the different robot controller hardware resources is always very similar: In order to program an hardware



resource of the robot controller, we need to declare an object of the correspondent class, and we need to create the object using the correspondent methods of the IntelliBrain class. To exemplified that pattern, write the source code needed to declare and create objects for the following hardware resources.

**LED:** `import com.ridgesoft.intellibrain.IntelliBrain; import com.ridgesoft.io.LED; LED statusLED = IntelliBrain.getStatusLed(); LED faultLED = IntelliBrain.getFaultLed();`

**BUZZER:** `import com.ridgesoft.intellibrain.IntelliBrain; import com.ridgesoft.io.Speaker; Speaker buzzer = IntelliBrain.getBuzzer();`

**Motor port:** `import com.ridgesoft.intellibrain.IntelliBrain; import com.ridgesoft.robotics.Motor; Motor motor = IntelliBrain.getMotor(1);`

**Analog port:** `import com.ridgesoft.intellibrain.IntelliBrain; import com.ridgesoft.robotics.AnalogInput; AnalogInput input = IntelliBrain.getAnalogInput(PORT_NUMBER);`

**Stop Button:** `import com.ridgesoft.intellibrain.IntelliBrain; import com.ridgesoft.robotics.PushButton; PushButton stopButton = IntelliBrain.getStopButton();`

7. Look for information about how to control the servomotors of the robot. Write down the name of the class we need to use and all the functions of that class. Look for that information in the following sources (<http://www.ridgesoft.com/intellibrain2/intellibrain2.htm>):

- User Guide of the IntelliBrain
- API Overview
- API Javadoc

Import necessário para servomotores:

`-import com.ridgesoft.robotics.servo;`

Nome da classe para servomotores:

`Public interface Servo;`

Funções para classe de servomotores:

- `Void off()`, que serve para desligar servomotor;



**Escola Superior de Tecnologia e Gestão**  
Instituto Politécnico da Guarda

## **Laboratorial Work**

- Void setPosition(int position), que recebe como parametron um inteiro com “range” entre 0 e 100.