

# Orientação a objetos

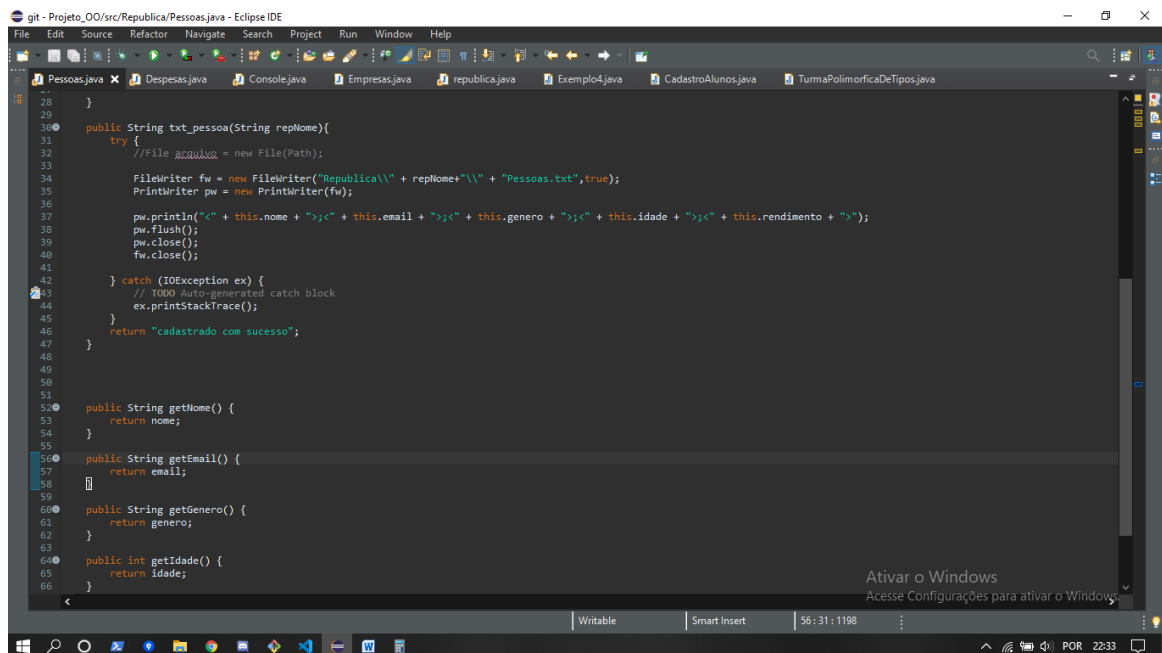
Professor: André Lanna

Alunos: Ítalo Fernandes Sales de Serra / 180102613

Gabriel Avelino Freire / 180100831

Para realizar esse projeto de trabalho para Orientação a Objeto usamos de conceitos aprendidos durante as aulas síncronas e assíncronas ministradas pelo professor. Para o total funcionamento do projeto utilizamos da criação de classes e objetos como as classes Pessoas, Despesas, republica, Empresas e Console que contém o método main(principal), todas contidas dentro do pacote Republica.

Cada classe possui seus métodos que chamam seus atributos usando o getter and setters:

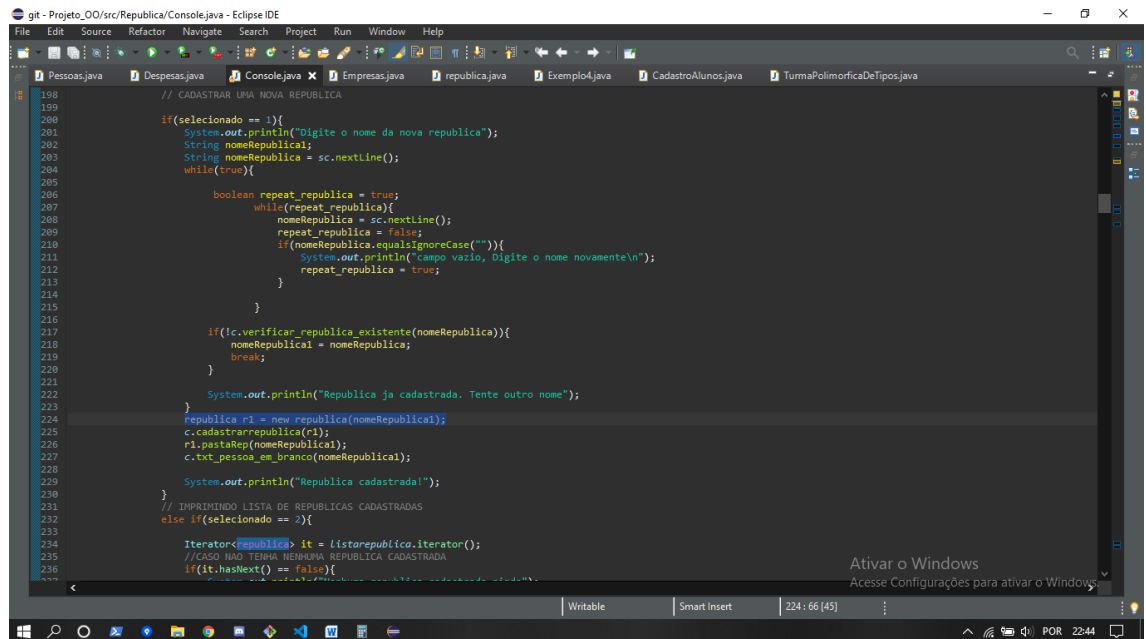


```
28 }
29
30 public String txt_pessoa(String repNome){
31     try {
32         //File arquivo = new File(Path);
33
34         FileWriter fw = new FileWriter("República\\" + repNome + "\\" + "Pessoas.txt", true);
35         PrintWriter pw = new PrintWriter(fw);
36
37         pw.println("<" + this.nome + "><" + this.email + "><" + this.genero + "><" + this.idade + "><" + this.rendimento + ">");
38         pw.flush();
39         pw.close();
40         fw.close();
41
42     } catch (IOException ex) {
43         // TODO Auto-generated catch block
44         ex.printStackTrace();
45     }
46     return "cadastrado com sucesso";
47 }
48
49
50
51
52 public String getNome() {
53     return nome;
54 }
55
56 public String getEmail() {
57     return email;
58 }
59
60 public String getGenero() {
61     return genero;
62 }
63
64 public int getIdade() {
65     return idade;
66 }
```

(Exemplo usando a Classe Pessoa)

Utilizamos também de uma das bases do conceito de Orientação a objeto que seria o polimorfismo. Usamos dois tipos de polimorfismo o de inclusão e o paramétrico.

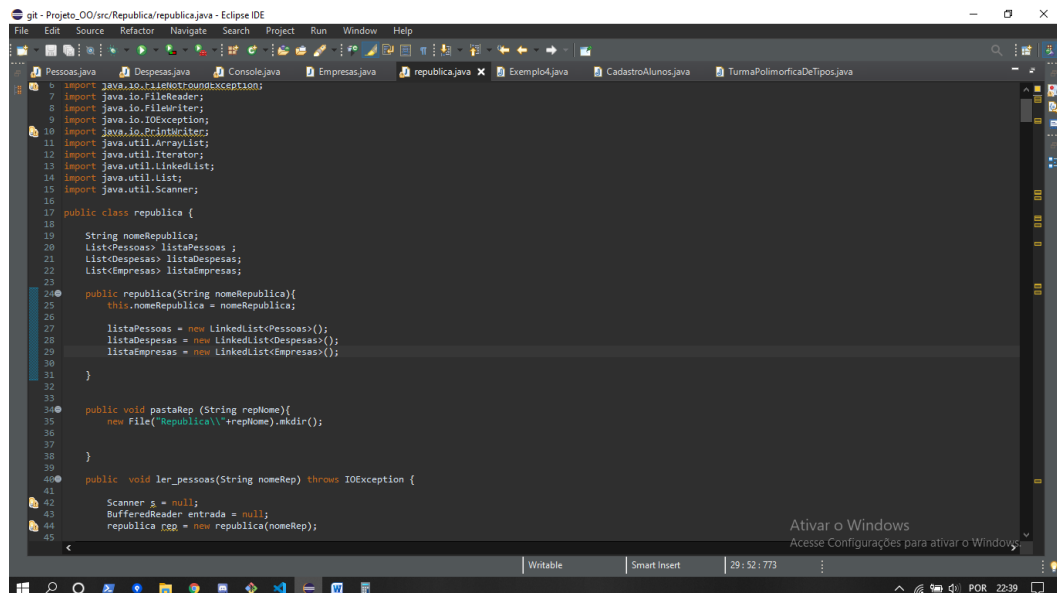
## Polimorfismo de inclusão:



```
198 // CADASTRAR UMA NOVA REPUBLICA
199
200 if(selecionado == 1){
201     System.out.println("Digite o nome da nova republica");
202     String nomeRepublica;
203     String nomeRepublica = sc.nextLine();
204     while(true){
205
206         boolean repeat_republica = true;
207         while(repeat_republica){
208             nomeRepublica = sc.nextLine();
209             repeat_republica = false;
210             if(nomeRepublica.equalsIgnoreCase("")){
211                 System.out.println("campo vazio, Digite o nome novamente\n");
212                 repeat_republica = true;
213             }
214         }
215
216         if(!lc.verificar_republica_existente(nomeRepublica)){
217             nomeRepublica = nomeRepublica;
218             break;
219         }
220     }
221     System.out.println("Republica ja cadastrada. Tente outro nome");
222 }
223
224 republica r1 = new republica(nomeRepublica);
225 c.cadastrarrepublica(r1);
226 r1.pastaRep(nomeRepublica);
227 c.txt_pessoa_em_branco(nomeRepublica);
228
229 System.out.println("Republica cadastrada!");
230
231 // IMPRIMINDO LISTA DE REPUBLICAS CADASTRADAS
232 else if(selecionado == 2){
233
234     Iterator<Republica> it = listorepublica.iterator();
235     //CASO NAO TENHA NENHUMA REPUBLICA CADASTRADA
236     if(it.hasNext() == false){
237         System.out.println("Nenhuma republica cadastrada");
238     }
239 }
```

(Exemplo de polimorfismo de inclusão na linha 224)

## Polimorfismo Paramétrico:



```
6 import java.io.FileNotFoundException;
7 import java.io.FileReader;
8 import java.io.FileWriter;
9 import java.io.IOException;
10 import java.io.PrintWriter;
11 import java.util.ArrayList;
12 import java.util.Iterator;
13 import java.util.LinkedList;
14 import java.util.List;
15 import java.util.Scanner;
16
17 public class republica {
18
19     String nomeRepublica;
20     List<Pessoas> listaPessoas ;
21     List<Despesas> listaDespesas;
22     List<Empresas> listaEmpresas;
23
24     public republica(String nomeRepublica){
25         this.nomeRepublica = nomeRepublica;
26
27         listaPessoas = new LinkedList<Pessoas>();
28         listaDespesas = new LinkedList<Despesas>();
29         listaEmpresas = new LinkedList<Empresas>();
30     }
31
32
33
34     public void pastaRep (String repNome){
35         new File("republica\\"+repNome).mkdir();
36     }
37
38
39
40     public void ler_pessoas(String nomeRep) throws IOException {
41
42         Scanner s = null;
43         BufferedReader entrada = null;
44         republica rep = new republica(nomeRep);
45     }
```

(Exemplo de polimorfismo na classe republica)

