

```

1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          ControladorAtendimento ctrl = new ControladorAtendimento(10);
7
8          try {
9              ctrl.inserir(new Paciente("Marcelo", 19, "123"));
10             ctrl.inserir(new Paciente("Marcos", 19, "222"));
11             ctrl.inserir(new Paciente("Victor", 36, "796"));
12             ctrl.inserir(new Paciente("Machado", 19, "55555"));
13             ctrl.inserir(new Paciente("Marilda", 61, "456"));
14             ctrl.inserir(new Paciente("Altair", 63, "031"));
15             ctrl.inserir(new Paciente("Ignes", 86, "555"));
16             ctrl.inserir(new Paciente("Roseli", 60, "22555"));
17         } catch (Exception e) {
18             System.out.println(e.getMessage());
19         }
20
21         while (true) {
22             System.out.println("1. Solicitar nova senha (Comum ou Prioridade)");
23             System.out.println("2. Excluir uma senha");
24             System.out.println("3. Listar todas as senhas");
25             System.out.println("4. Visualizar próximo");
26             System.out.println("5. Chamar próximo");
27             System.out.println("6. Sair");
28             System.out.print("Escolha: ");
29             int op = sc.nextInt();
30             sc.nextLine();
31             try {
32                 if (op == 1) {
33                     System.out.print("Nome: ");
34                     String nome = sc.nextLine();
35                     System.out.print("Idade: ");
36                     int idade = sc.nextInt();
37                     sc.nextLine();
38                     System.out.print("CPF: ");
39                     String cpf = sc.nextLine();
40                     ctrl.inserir(new Paciente(nome, idade, cpf));
41                 } else if (op == 2) {
42                     System.out.print("Fila (1 - Prioritária, 2 - Comum): ");
43                     int filaTipo = sc.nextInt();
44                     System.out.print("Posição da senha: ");
45                     int pos = sc.nextInt();
46                     sc.nextLine();
47                     ctrl.excluirSenha(filaTipo == 1, pos);
48                 } else if (op == 3) {
49                     ctrl.listarSenhas();
50                 } else if (op == 4) {
51                     System.out.println("Próximo: " + ctrl.proximo());
52                 } else if (op == 5) {
53                     ctrl.chamarProximo();
54                 } else if (op == 6) {
55                     break;
56                 } else {
57                     System.out.println("Opção inválida.");
58                 }
59             } catch (Exception e) {
60                 System.out.println(e.getMessage());
61             }
62         }
63         sc.close();
64     }

```

```

65 }
66
67 public class ControladorAtendimento {
68     private FilaPrioritaria filaPrioritaria;
69     private FilaComum filaComum;
70     private int prioridadeCount;
71
72     public ControladorAtendimento(int capacidade) {
73         filaPrioritaria = new FilaPrioritaria(capacidade);
74         filaComum = new FilaComum(capacidade);
75     }
76
77     public void inserir(Paciente p) throws Exception {
78         if (p.isPreferencial())
79             filaPrioritaria.inserir(p);
80         else
81             filaComum.inserir(p);
82     }
83
84     public Paciente proximo() throws Exception {
85         return selecionarProximo(false);
86     }
87
88     public void chamarProximo() throws Exception {
89         Paciente p = selecionarProximo(true);
90         System.out.println("Chamando: " + p);
91     }
92
93     private Paciente selecionarProximo(boolean remove) throws Exception {
94         if (prioridadeCount < 3 && filaPrioritaria.tamanho() > 0) {
95             Paciente p = filaPrioritaria.proximo();
96             if (remove) filaPrioritaria.chamarProximo();
97             prioridadeCount++;
98             return p;
99         }
100         if (filaComum.tamanho() > 0) {
101             Paciente p = filaComum.proximo();
102             if (remove) filaComum.chamarProximo();
103             prioridadeCount = 0;
104             return p;
105         }
106         if (filaPrioritaria.tamanho() > 0) {
107             Paciente p = filaPrioritaria.proximo();
108             if (remove) filaPrioritaria.chamarProximo();
109             prioridadeCount = Math.min(prioridadeCount + 1, 3);
110             return p;
111         }
112         throw new Exception("Fila vazia");
113     }
114
115     public void excluirSenha(boolean isPrioritaria, int pos) throws Exception {
116         if (isPrioritaria)
117             filaPrioritaria.remove(pos);
118         else
119             filaComum.remove(pos);
120     }
121
122     public void listarSenhas() {
123         System.out.println("Fila Prioritária:");
124         filaPrioritaria.listar();
125         System.out.println("Fila Comum:");
126         filaComum.listar();
127     }
128 }
129
130 public class FilaPrioritaria {

```

```

131     private Vetor pacientes;
132
133     public FilaPrioritaria(int capacidade) {
134         pacientes = new Vetor(capacidade);
135     }
136
137     public void inserir(Paciente p) throws Exception {
138         pacientes.adiciona(p);
139     }
140
141     public Paciente proximo() throws Exception {
142         if (pacientes.tamanho() == 0) throw new Exception("Fila vazia");
143         return (Paciente) pacientes.busca(0);
144     }
145
146     public Paciente chamarProximo() throws Exception {
147         if (pacientes.tamanho() == 0) throw new Exception("Fila vazia");
148         Paciente p = (Paciente) pacientes.busca(0);
149         pacientes.remove(0);
150         return p;
151     }
152
153     public void remover(int pos) throws Exception {
154         pacientes.remove(pos);
155     }
156
157     public int tamanho() {
158         return pacientes.tamanho();
159     }
160
161     public void listar() {
162         for (int i = 0; i < pacientes.tamanho(); i++) {
163             try {
164                 System.out.println(pacientes.busca(i));
165             } catch (Exception e) {}
166         }
167     }
168 }
169
170 public class FilaComum {
171     private Vetor pacientes;
172
173     public FilaComum(int capacidade) {
174         pacientes = new Vetor(capacidade);
175     }
176
177     public void inserir(Paciente p) throws Exception {
178         pacientes.adiciona(p);
179     }
180
181     public Paciente proximo() throws Exception {
182         if (pacientes.tamanho() == 0) throw new Exception("Fila vazia");
183         return (Paciente) pacientes.busca(0);
184     }
185
186     public Paciente chamarProximo() throws Exception {
187         if (pacientes.tamanho() == 0) throw new Exception("Fila vazia");
188         Paciente p = (Paciente) pacientes.busca(0);
189         pacientes.remove(0);
190         return p;
191     }
192
193     public void remover(int pos) throws Exception {
194         pacientes.remove(pos);
195     }
196

```

```

197     public int tamanho() {
198         return pacientes.tamanho();
199     }
200
201     public void listar() {
202         for (int i = 0; i < pacientes.tamanho(); i++) {
203             try {
204                 System.out.println(pacientes.busca(i));
205             } catch (Exception e) {}
206         }
207     }
208 }
209
210 public class Vetor {
211     public Object[] elementos;
212     public int tamanho;
213
214     public Vetor(int capacidade) {
215         elementos = new Object[capacidade];
216         tamanho = 0;
217     }
218
219     public void adiciona(Object elemento) throws Exception {
220         aumentaCapacidade();
221         elementos[tamanho] = elemento;
222         tamanho++;
223     }
224
225     public int tamanho() {
226         return tamanho;
227     }
228
229     public Object busca(int pos) throws Exception {
230         if (pos < 0 || pos >= tamanho) throw new Exception("Posição inválida");
231         return elementos[pos];
232     }
233
234     private void aumentaCapacidade() {
235         if (tamanho == elementos.length) {
236             Object[] novo = new Object[elementos.length * 2];
237             for (int i = 0; i < elementos.length; i++) {
238                 novo[i] = elementos[i];
239             }
240             elementos = novo;
241         }
242     }
243
244     public void remove(int pos) throws Exception {
245         if (pos < 0 || pos >= tamanho) throw new Exception("Posição inválida");
246         for (int i = pos; i < tamanho - 1; i++) {
247             elementos[i] = elementos[i + 1];
248         }
249         tamanho--;
250     }
251 }
252
253 public class Paciente {
254     private String nome;
255     private int idade;
256     private String cpf;
257     private boolean preferencial;
258
259     public Paciente(String nome, int idade, String cpf) {
260         this.nome = nome;
261         this.idade = idade;
262         this.cpf = cpf;

```

```
263     preferencial = idade >= 60;
264 }
265
266 public boolean isPreferencial() {
267     return preferencial;
268 }
269
270 public String toString() {
271     return "Nome: " + nome + ", Idade: " + idade + ", CPF: " + cpf;
272 }
273
274 public boolean equals(Object obj) {
275     if (this == obj) return true;
276     if (!(obj instanceof Paciente)) return false;
277     Paciente p = (Paciente) obj;
278     return nome.equals(p.nome) && idade == p.idade && cpf.equals(p.cpf);
279 }
280 }
281
```