

Desenvolvimento de Sistemas Software

LEI - 3º ano

Ficha Prática - Diagramas de Sequência 1

2011/12

Compras online

Considere o seguinte extracto de código Java (o método `comprados` determina um `ArrayList` com todos os bilhetes comprados por um dado comprador). Note que o `ArrayList` `res` é passado por referência no método `addBilhetes`.

```
public class Compras {

    private HashMap<String,Comprador> baseDados; // idComprador -> Comprador
    ...

    public ArrayList<String> comprados(String bi) {
        ArrayList<String> res;
        boolean existe;

        existe = this.baseDados.containsKey(bi);
        if (existe)
            res = this.calcula(bi);
        return res;
    }

    public ArrayList<String> calcula(String bi) {
        Comprador c = this.baseDados.get(bi);
        ArrayList<String> res = new ArrayList<String>();
        c.addBilhetes(res);
        return res;
    }

    ...
}

public class Comprador {

    private ArrayList<String> bilhetes;
    ...
}
```

```

public void addBilhetes(ArrayList<String> res) {
    String o;
    int i=0;
    tam = this.bilhetes.size();

    while(i < tam) {
        o = this.bilhetes.get(i);
        res.add(o);
        i++;
    }

}
...
}

```

1. Escreva um **Diagrama de Sequência** que descreva o comportamento do método.
2. Considere agora que no método `addBilhetes` o ciclo `while` é substituído por:

```
res.addAll(this.bilhetes);
```

Escreva o **Diagrama de Sequência** correspondente ao diagrama de sequência da pergunta anterior, mas com a nova versão do método.

Sistema de Avaliação de Trabalhos

Considere o excerto de código Java que a seguir se apresenta:

```

interface Identificavel {
    void getID();
}

abstract class Pessoa {
    private String nome;
    abstract void setNome(String n); }

class Aluno extends Pessoa implements Identificavel {
    private Grupo m_g;
    private int numAluno;
    private int notaTeo;
    private int bounsPrat;
    void regista(Grupo g) {...};
}

```

```

class Grupo {
    private int cod;
    private int nota;
    private ArrayList<Entrega> entregas;
    void addEntrega(Entrega e) {...}
}

class Entrega implements Identificavel {
    private Date data;
    private int nota_docente;
    private Aluno avaliador;
    private int nota_avaliador;
    private String comentarios;
}

class Docente extends Pessoa implements Identificavel{
    private int cod;
}

class SGT {
    private Docente responsavel;
    private ArrayList <Docente> docentes_praticas;
    private TreeMap <Integer,Aluno> alunos;
    private ArrayList <Grupo> grupos;
    void getNotaAluno(int codAluno) {...}
    boolean validaAvaliadores() {...}
}

```

Relativamente ao código apresentado:

1. Apresente o **Diagrama de Sequência** para o método boolean `validaAvaliadores()`.
O método deverá verificar que nenhum aluno seja avaliador do seu próprio grupo.