

Nome: Vinícius Andrade Frederico RA: 139223 Turma: A

Obs: O arquivo lab1b.zip contém o código por completo. Abaixo estão somente os códigos necessários para o relatório.

Pacote *br.unicamp.ic.mc302.contador*)

Item 1)

Temos como saída:

0

1

Item 2)

Implementando o contador de vogais abaixo recebemos que tem 5 A, 6 E, 1 I, 6 O e 1 U:

ContadorVogais.java:

```
package br.unicamp.ic.mc302.contador;
```

```
// Arquivo ContadorVogais.java
```

```
public class ContadorVogais {
```

```
    public static void main(String args[]) {
```

```
        char fraseExaminada[] = {'T','e','x','t','o',' ','E','x','e','m','p','l','o',' ','P','a','r','a',' ','A',' ','C','o','n','t','a','g','e','m',' ','D','o',' ','N','u','m','e','r','o',' ','D','e',' ','V','o','g','a','l','i','s','.'};
```

```
        Contador[] contVogais = new Contador[5]; // um para cada vogal
```

```
        for(int i = 0; i < 5; i++){
```

```
            contVogais[i] = new Contador();
```

```
        }
```

```
        for(int i = 0; i < fraseExaminada.length; i++){
```

```
            switch(Character.toLowerCase(fraseExaminada[i])){
```

```
                case 'a':
```

```
                    contVogais[0].incrementa();
```

```
                    break;
```

```
                case 'e':
```

```
                    contVogais[1].incrementa();
```

```
                    break;
```

```
                case 'i':
```

```
                    contVogais[2].incrementa();
```

```
                    break;
```

```
                case 'o':
```

```
                    contVogais[3].incrementa();
```

```
                    break;
```

```
                case 'u':
```

```
                    contVogais[4].incrementa();
```

```
                    break;
```

```
                default:
```

```
                    break;
```

```
            }
```

```
        }
```

```
        System.out.println("A: " + contVogais[0].retornaNum());
```

```
        System.out.println("E: " + contVogais[1].retornaNum());
```

```
        System.out.println("I: " + contVogais[2].retornaNum());
```

```
        System.out.println("O: " + contVogais[3].retornaNum());
```

```
        System.out.println("U: " + contVogais[4].retornaNum());
```

```
    }
```

```
} // Fim da classe ContadorVogais
```

Pacote *br.unicamp.ic.mc302.circulo*)

Item 1)

Temos como saída:

c1: (3, 3)

c2: (2, 1)

c3: (3, 3)

Raio de c1: 1; Circunferência de c1: 6

Item 2)

Alterando o código de TestaCirculo.java como abaixo, conclui-se que quando se copia a referência ou o objeto, ao alterar um deles (no caso o original), altera-se todas as cópia. Para isso não acontecer precisa-se criar um "clone" do objeto a ser copiado, assim, terá outra referência. Foi criado o método setRaio na classe Circulo para alterar o valor dos raios.

Código de TestaCirculo.java:

```
package br.unicamp.ic.mc302.circulo;
// arquivo TestaCirculo.java
class TestaCirculo {
    public static void main (String args[ ]) {
        Circulo v1[] = new Circulo[5];
        v1[0] = new Circulo(1,2,7);
        v1[1] = new Circulo(3,2,3);
        v1[2] = new Circulo(1,5,8);
        v1[3] = new Circulo(7,4,5);
        v1[4] = new Circulo(6,6,2);

        System.out.println("b) Imprimindo v1");
        for(int i = 0; i < 5; i++){
            System.out.println("v1[" + i + "]: (" + v1[i].getX() + ", " + v1[i].getY() + ", " +
+ v1[i].getRaio() + ")");

            System.out.println("e) Imprimindo v1 e v2");
            Circulo v2[] = v1;
            for(int i = 0; i < 5; i++){
                System.out.print("v1[" + i + "]: (" + v1[i].getX() + ", " + v1[i].getY() + ", " +
+ v1[i].getRaio() + ") ");
                System.out.println("v2[" + i + "]: (" + v2[i].getX() + ", " + v2[i].getY() + ", " +
+ v2[i].getRaio() + ")");
            }

            Circulo v3[] = new Circulo[5];
            for(int i = 0; i < 5; i++)
                v3[i] = v1[i];

            v1[0].setRaio(3);
            v1[1].setRaio(3);
            v1[2].setRaio(3);
            v1[3].setRaio(3);
            v1[4].setRaio(3);

            System.out.println("i) Imprimindo v1, v2, v3");
            for(int i = 0; i < 5; i++){
                System.out.print("v1[" + i + "]: (" + v1[i].getX() + ", " + v1[i].getY() + ", " +
+ v1[i].getRaio() + ") ");
                System.out.print("v2[" + i + "]: (" + v2[i].getX() + ", " + v2[i].getY() + ", " +
+ v2[i].getRaio() + ") ");
                System.out.println("v3[" + i + "]: (" + v3[i].getX() + ", " + v3[i].getY() + ", " +
+ v3[i].getRaio() + ")");
            }
        }
    }
}

} // Fim da classe TestaCirculo
```

Código do método setRaio abaixo:

```
public void setRaio(int r){  
    this.raio = r;  
}
```

Saída:

b) Imprimindo v1

v1[0]: (1, 2, 7)

v1[1]: (3, 2, 3)

v1[2]: (1, 5, 8)

v1[3]: (7, 4, 5)

v1[4]: (6, 6, 2)

e) Imprimindo v1 e v2

v1[0]: (1, 2, 7) v2[0]: (1, 2, 7)

v1[1]: (3, 2, 3) v2[1]: (3, 2, 3)

v1[2]: (1, 5, 8) v2[2]: (1, 5, 8)

v1[3]: (7, 4, 5) v2[3]: (7, 4, 5)

v1[4]: (6, 6, 2) v2[4]: (6, 6, 2)

i) Imprimindo v1, v2, v3

v1[0]: (1, 2, 3) v2[0]: (1, 2, 3) v3[0]: (1, 2, 3)

v1[1]: (3, 2, 3) v2[1]: (3, 2, 3) v3[1]: (3, 2, 3)

v1[2]: (1, 5, 3) v2[2]: (1, 5, 3) v3[2]: (1, 5, 3)

v1[3]: (7, 4, 3) v2[3]: (7, 4, 3) v3[3]: (7, 4, 3)

v1[4]: (6, 6, 3) v2[4]: (6, 6, 3) v3[4]: (6, 6, 3)

Item 3)

Independentemente de se obter diretamente do atributo ou do método, os resultados obtidos são iguais pois o método simplesmente retorna o valor do atributo.

Pacote *br.unicamp.ic.mc302.listaInts*)

Item 1)

Temos como saída:

3 pertence à list1? true

42 está na list1? false

3 está na list2 (vazia)? false

Soma dos elementos da list1: 45

Soma dos elementos da list2: 0

list1: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

list2: []

Item 2)

O objeto `StringBuilder` tem como função montar uma `String`, ou seja, ele não é uma `String`, mas um objeto com métodos para facilitar a "formatação" da `String` que se deseja.

Item 3)

Alterando o código como abaixo, temos como saída adicional a saída já existente:

3 pertence à list1? true

3 pertence à list1? false

Método `removeElem` na classe `ListaInts`:

```
public void removeElem(Object elem){  
    this.lista.remove(elem);  
}
```

Código adicionado ao arquivo `TestaLista.java`:

```
System.out.println("3 pertence à list1? " + list1.elem(3));  
list1.removeElem(3);  
System.out.println("3 pertence à list1? " + list1.elem(3));
```

Item 4)

Código de `MinhaMatriz.java`:

```
package br.unicamp.ic.mc302.minhaMatriz;  
import java.util.ArrayList;  
  
public class MinhaMatriz {  
    ArrayList<ArrayList<Integer>> linhas;  
    public MinhaMatriz(){  
        linhas = new ArrayList< ArrayList<Integer> >();  
    }  
  
    public void adicionaLinha(){  
        ArrayList<Integer> l = new ArrayList<Integer>();  
        for(int i = 0; i < numColunas(); i++) // Adiciona as colunas já existentes  
            l.add(0);  
        linhas.add(l);  
    }  
  
    public void adicionaColuna(){  
        for( ArrayList<Integer> x: linhas){  
            x.add(0);  
        }  
    }  
}
```

```

public void setElemento(int linha,int coluna, int elem){
    ArrayList<Integer> l = linhas.get(linha);
    l.set(coluna, elem);
}

public int getElemento(int linha, int coluna){
    return linhas.get(linha).get(coluna);
}

public int numLinhas(){
    return linhas.size();
}

public int numColunas(){
    if(linhas.size() > 0)
        return linhas.get(0).size();
    else
        return 0;
}

public MinhaMatriz multiplica(MinhaMatriz m2){
    MinhaMatriz r = new MinhaMatriz();
    for( int l = 0; l < this.numLinhas(); l ++){
        r.adicionaLinha();
        for(int c = 0; c < m2.numColunas(); c++){
            r.adicionaColuna();
            for( int l = 0; l < this.numLinhas(); l ++){
                for(int c = 0; c < m2.numColunas(); c++){
                    Integer soma = 0;
                    for (int i = 0; i < this.numColunas(); i++) {
                        soma += this.getElemento(l, i) * m2.getElemento(i, c);
                    }
                    r.setElemento(l, c, soma);
                }
            }
        }
        return r;
    }
}

public void imprime(){
    for(int i = 0; i < numLinhas(); i++){
        for(int c = 0; c < numColunas(); c++){
            System.out.print(getElemento(i, c) + " ");
        }
        System.out.print("\n");
    }
}
}

```

Código de TestaMinhaMatriz.java:

```
package br.unicamp.ic.mc302.minhaMatriz;
```

```
public class TestaMinhaMatriz {  
    public static void main (String [] args) {  
  
        MinhaMatriz m1 = new MinhaMatriz();  
        m1.adicionaLinha();  
        m1.adicionaLinha();  
        m1.adicionaColuna();  
        m1.adicionaColuna();  
        m1.setElemento(0, 0, 1);  
        m1.setElemento(0, 1, 2);  
        m1.setElemento(1, 0, 3);  
        m1.setElemento(1, 1, 4);  
        System.out.println("M1");  
        m1.imprime();  
  
        MinhaMatriz m2 = new MinhaMatriz();  
        m2.adicionaLinha();  
        m2.adicionaLinha();  
        m2.adicionaColuna();  
        m2.adicionaColuna();  
        m2.setElemento(0, 0, 3);  
        m2.setElemento(1, 0, 4);  
        m2.setElemento(0, 1, 6);  
        m2.setElemento(1, 1, 9);  
        System.out.println("M2");  
        m2.imprime();  
  
        MinhaMatriz m3 = m1.multiplica(m2);  
        System.out.println("M3");  
        m3.imprime();  
  
    }  
}
```

Saída:

```
M1  
1 2  
1 2  
M2  
1 1  
2 2  
M3  
5 5  
5 5
```