Trabalho teórico - 03

29 de agosto de 2021

1. Parte

(a) Resolução:

```
1   public static boolean busca(int[] array, int elemento){
        for(int i=0; i<array.length; i++){
3           if(array[i] == elemento)
                return true;
5       }
        return false;
7   }
```

(b) Resolução:

```
1   public static boolean buscaBinaria(int[] array, int elemento){
        int meio;
3       int limiteS = array.length-1;
        int limiteI = 0;
5
        while(limiteI <= limiteS){
7           meio = (limiteS+limiteI)/2;
            if(elemento == array[meio])
9               return true;
            if(elemento < array[meio])
11              limiteS = meio - 1;
            else
13              limiteI = meio + 1;
        }
15      return false;
    }
```

(c) Resolução:

```
    public static void maiorMenor(int[] array){
2       int menor = array[0];
        int maior = menor;
4
        for(int i=0; i<array.length; i++){
6           if(array[i] > maior)
                maior = array[i];
8           if(array[i] < menor)
                menor = array[i];
10      }
        MyIO.println("Maior: " + maior);
12      MyIO.println("Menor: " + menor);
    }
```

(d) Resolução:

Já faz o mínimo de comparações possíveis.

(e) Resolução:

O código testa o caracter passado por parâmetro para ver se o mesmo é vogal ou não.

(f) Resolução:

O segundo está mais intuitivo.

(g) Resolução:

Acredito que os nomes das variáveis estão muito grandes, poderiam estar menores.

(h) Resolução:

No primeiro, irá retornar o valor de i e depois decrementá-la em 1, já no segundo, primeiro irá decrementá-la em 1 e depois retornar o valor da variável i.

(i) Resolução:

A variável b será uma progressão de 0 até 1, a s de 0 à 32767, a i de 0 à 2147483647 e, finalmente, a l de 0 à 2147483647.

2. Parte

(a) Resolução:

```
1   public static String conteudoArq(String path) throws IOException {          //
        Retorna o conteudo presente no arquivo
        BufferedReader Arq = new BufferedReader(new FileReader(path));
3
        String conteudo = "";
5       String aux = "";
        while (aux != null) {
7     conteudo += aux;
            aux = Arq.readLine();
9           if(aux!=null)
                aux+='\n';
11      }
        Arq.close();
13
        return conteudo;
15  }

17  public static void copiaArquivo(String pathOriginal, String pathCopia) throws
        IOException{          //Escreve o conte do do arquivo do primeiro
        parametro no arquivo do segundo
        BufferedWriter Arq = new BufferedWriter(new FileWriter(pathCopia));
19
        Arq.append(conteudoArq(pathOriginal));
21      Arq.close();
    }
```

(b) Resolução:

```
        public static String copiarParaString(String path) throws IOException{
                                                          //Retorna o conteudo
            original do arquivo
2       BufferedReader arq = new BufferedReader(new FileReader(path));

4       String conteudo = "";
        String aux;
6
        while(true){
8           aux = arq.readLine();
            if(aux == null)
```

```
10                  break;
                else{
12                  conteudo += aux;
                    conteudo += '\n';
14              }
            }
16
            arq.close();
18          return conteudo;
        }
20      public static void substituiParaOriginal(String path, String conteudo) throws
            IOException{                            //Exclui todo o conteudo do arquivo e
             o transforma na string passada por parametro
            BufferedWriter arq = new BufferedWriter(new FileWriter(path));
22
            arq.append(conteudo);
24
            arq.close();
26      }
        public static void inserePilha(String path, String frase, String
            conteudoExistente) throws IOException{
28          BufferedWriter arqF = new BufferedWriter(new FileWriter(path));

30          frase += '\n';
            frase += conteudoExistente;
32
            arqF.append(frase);
34
            MyIO.println(conteudoArq(path));
36          arqF.close();
        }
38      public static void deletar(String path) throws IOException{
                                                                    //M todo para
            dele   o de elementos - Se descomentar os elementes da classe
            BufferedWriter funciona, por m est  com bug
            BufferedReader arq = new BufferedReader(new FileReader(path));
40          //BufferedWriter arqW = new BufferedWriter(new FileWriter(path));

42          boolean primeiraRep = true;
            String aux;
44          String strFinal = "";

46          while(true){
                aux = arq.readLine();
48              System.out.println(aux);
                if(aux == null)
50                  break;
                else{
52                  if(primeiraRep)
                        primeiraRep = false;
54                  else{
                    strFinal += aux;
56                  strFinal += '\n';
                    }
58              }
            }
60          //arqW.append(strFinal);

62          //arqW.close();
            arq.close();
64      }

66      public static String conteudoArq(String path) throws IOException {
                                                                    //Retorna o
            conte do presente em um arquivo -/- 2
            BufferedReader Arq = new BufferedReader(new FileReader(path));
68
            String conteudo = "";
```

```
70          String aux = "";
            while (aux != null) {
72        conteudo += aux;
                aux = Arq.readLine();
74              if(aux!=null)
                    aux+='\n';
76          }
            Arq.close();
78
            return conteudo;
80      }

82      public static void main(String[] args) throws IOException{
        String path = "pilha.txt";
84      String conteudoInicial = copiarParaString(path);


86
        String aux;
88      String frase;
        int opcao;
90
        do{
92
            MyIO.println("\nInsira uma opcao\n1. Inserir\n2. Remover\n3. Listar\n4.
                Sair");
94          opcao = MyIO.readInt();
            if(!(opcao<1 || opcao>4)){
96              switch(opcao){
                    case 1:
98                  frase = MyIO.readLine();
                    aux = conteudoArq(path);
100                 inserePilha(path, frase, aux);
                    break;
102
                    case 2:
104                 //deletar(path);          //Bug na fun   o
                    System.out.println("M todo indispon vel, bug na hora de
                        representar a fun   o, escolha outro!");
106                 break;

108                 case 3:
                    MyIO.println(conteudoArq(path));
110                 break;

112                 default:
                    MyIO.println("ERROR, INSIRA OUTRA OPCAO");
114             }
            }
116         else
                MyIO.println("\nOpcao invalida, insira outra.\n");
118
        }while(opcao!=4);
120     MyIO.println(conteudoArq(path));
        substituiParaOriginal(path, conteudoInicial);
122
        //arqW.close();
124     }
```

3. Parte

(a) Resolução:

```
        public static float media(int[] vetor){                         //Retorna a
             media dos elementos do vetor
2       float mediaE = 0;
```

```
        for(int i=0; i<vetor.length; i++)
4       mediaE+=vetor[i];

6       return mediaE/vetor.length;
    }
8
    public static void main(String[] args){
10      int n = MyIO.readInt();
        int[] vetor = new int[n];
12
        for(int i=0; i<n; i++)
14      vetor[i] = MyIO.readInt();

16      float mediaElementos = media(vetor);

18      for(int i=0; i<n; i++){
            if(vetor[i] > mediaElementos)
20              MyIO.println(vetor[i]);
        }
22  }
```

4. Parte

(a) Resolução:

A recursividade está presente na linha 6 dos dois códigos. No primeiro, é uma função que calcula o fatorial, sendo assim, a função é chamada subtraindo em 1 a variável n até que a mesma atinga 1.

Já no segundo código, ele chama a função duas vezes, somando-as, sendo a primeira subtraindo em 1 a variável n e na segunda subtraindo em 2 a variável n. Com isso, é a realizada a soma das duas funções, formando uma soma de Fibonacci.

(b) Resolução:

Porque primeiro ele imprime a variável i e chama a função novamente, isso até atingir a condição de parada e, logo quando a atinge, imprime o valor da variável i correspondente às outras chamadas de função.

(c) Resolução:

```
    public static int multiplicacao(int a, int b){
2       int soma = 0;
        if(b>0)
4           soma = a + multiplicacao(a, b-1);

6       return soma;
    }
```

(d) Resolução:

```
        public static int maiorElemento(int[] vetor, int tamanho, int maior){
                                        //Retorna o maior elemento
2       if(tamanho == 0)
        return maior;
4
        else{
6           if(vetor[tamanho] > maior)
            maior = vetor[tamanho];
8           return maiorElemento(vetor, --tamanho, maior);
        }
10  }
```

```
12
    public static void main(String [] args){
14      int n = MyIO.readInt();
        int[] vetor = new int[n];
16
        for(int i=0; i<n; i++)
18      vetor[i] = MyIO.readInt();

20      MyIO.println(maiorElemento(vetor, n-1, vetor[0]));
    }
```

(e) Resolução:

```
1       public static boolean isPalindromo(char[] vect, int pInicial, boolean boo){
                            //Retorna um valor boolean para palindromo ou nao
        if(pInicial >= vect.length/2)
3           return boo;
        else{
5           if(vect[pInicial] == vect[(vect.length-1)-pInicial])
                return isPalindromo(vect, ++pInicial, boo);
7           else
                return isPalindromo(vect, vect.length-1, false);
9       }
    }
```

(f) Resolução:

```
        public static int quantidadeVogais(char[] frase, int tamanho, int vogais){
                            //Retorna um valor indicando a quantidade de vogais na
             frase
2       if(isVogal(frase[tamanho]))
                                                                    //O metodo
            isVogal retorna um valor do tipo boolean indicando se o char    vogal
            ++vogais;
4
        if(tamanho == 0)
6           return vogais;
        else
8           return quantidadeVogais(frase, --tamanho, vogais);
    }
```

(g) Resolução:

```
1       public static int quantidadeMinusculas(char[] frase, int tamanho, int qtd){
                            //Retorna a quantidade de minusculas na frase
        if(isMinuscula(frase[tamanho]))
                                                                //Este metodo
            retorna um valor boolean indicando se e minuscula
3       ++qtd;

5       if(tamanho == 0)
        return qtd;
7       else
        return quantidadeMinusculas(frase, --tamanho, qtd);
9   }
```

(h) Resolução:

```
        public static int ordenacao(int[] array, int pos){
                                                    //Ordena o array de forma
            crescente
        int menor = array[pos];
        int aux = pos;

        for(int i=pos+1; i<array.length; i++){
            if(array[i] < menor){
            menor = array[i];
            aux = i;
            }
        }
        array[aux] = array[pos];
        array[pos] = menor;

        if(pos == array.length-1)
        return 0;
        else
        return ordenacao(array, ++pos);
    }
```

(i) Resolução:

```
        public static int e2(int n) {   //Retorna a conta solicitada
        if (n == 0)
            return 1;
        else
            return (int) Math.pow((n - 1), 2);
    }
```

5. Parte

(a) Resolução:

Imprime na tela que são diferentes (b) -¿ Pois está comparando dois Objetos, não suas propriedades. Sendo assim, são distintos.

(b) Resolução:

i. Está errada, pois o segundo parâmetro é do tipo int e o terceito do tipo char.

ii. Está certa, pois o objeto c é um ponteiro para o objeto b, sendo assim, quando altera uma propriedade em c, altera consecutivamente em b.

iii. Está certa, pois existe um método para alterar este atributo.

iv. Está errada, pois ele é mutável.

6. Parte

(a) Resolução:

```
        Retangulo::Retangulo(double base, double altura){
            this->base = base;
            this->altura = altura;
        }
        double Retangulo::getArea(){
            double area = this->altura * this->base;
            return area;
        }
        double Retangulo::getPerimetro(){
            double perimetro = (this->altura*2) + (this->base*2);
            return perimetro;
```

```
        }
        double Retangulo::getDiagonal(){
            double diagonal = sqrt(pow(this->altura, 2) + pow(this->base, 2));
            return diagonal;
        }

        int main(){
            Retangulo *r1 = new Retangulo(10, 2);
            Retangulo *r2 = new Retangulo(20, 4);

            cout<<"\n"<<r2->getArea();
            cout<<"\n"<<r1->getArea();

            cout<<"\n"<<r2->getPerimetro();
            cout<<"\n"<<r1->getPerimetro();

            cout<<"\n"<<r2->getDiagonal();
            cout<<"\n"<<r1->getDiagonal();

            return 0;
        }
```

```
        class Retangulo{
            private:
            double base;
            double altura;

            public:
            Retangulo(double base, double altura);
            double getArea();
            double getPerimetro();
            double getDiagonal();
        };
```

(b) Resolução:

```
        #include <math.h>

class Ponto{
    private:
    double x;
    double y;
    int id;
    int static nextID = 0;

    public:
    Ponto();
    Ponto(double x, double y);
    void alterarId();
    int getId();
    static void alterarNextID();
    int getNextId();
    void setX(double valor);
    double getX();
    void setY(double valor);
    double getY();
    double dist(double ponto2);
    double dist(Ponto ponto);
    int isTriangulo(Ponto ponto1, Ponto ponto2, Ponto ponto3);
    double getAreaRetangulo(Ponto ponto);
};

    Ponto::Ponto(){
        this->x = 0;
```

```
29          this->y = 0;
    }
31  Ponto::Ponto(double x, double y){
        this->x = x;
33      this->y = y;
    }
35  void Ponto::alterarId(){
        this->id = this->nextID;
37      Ponto::alterarNextID();
    }
39  int Ponto::getId(){
        return this->id;
41  }
    void Ponto::alterarNextID(){
43      nextID++;
    }
45  int Ponto::getNextId(){
        return this->nextID;
47  }
    double Ponto::dist(double ponto2){
49      return this->x - ponto2;
    }
51  double Ponto::dist(Ponto ponto){
        return this->x - ponto.getX();
53  }
    int Ponto::isTriangulo(Ponto ponto1, Ponto ponto2, Ponto ponto3){
55      if(ponto1.getX() > abs(ponto2.getX() - ponto3.getX()) && ponto1.getX() <
            ponto2.getX() + ponto3.getX()){
            if (ponto2.getX() > abs(ponto1.getX() - ponto3.getX()) && ponto2.getX()
                 < ponto1.getX() + ponto3.getX()){
57              if(ponto3.getX() > abs(ponto1.getX() - ponto2.getX()) && ponto3.
                    getX() < ponto1.getX() + ponto2.getX())
                return 1;
59              else
                return 0;
61          }
            else
63          return 0;
        }
65      else
        return 0;
67  }
    double Ponto::getAreaRetangulo(Ponto ponto){
69      return ponto.getX()*ponto.getY();
    }
```

7. Parte

- Conteúdo encontra-se no arquivo "parte7.pdf"no arquivo .zip enviado!