

Slide B.

1) Faça um método que receba um array de inteiros e um número inteiro x e retorne um valor booleano informando se o elemento x está contido no array.

class Qb

{

  public static boolean comp(int[] s , int num)

  {

    boolean resp=false;

    for(int i=0 ;i<s.length ;i++)

    {

      if(num==s[i])

        resp=true;

    }

    return resp;

  }

  public static void main(String[] args)

  {

    int[] iniciaValores = {12,32,54,6,8,89,64,64,6};

    int numero=6;

    boolean resp;

    resp=comp(iniciaValores,numero);

    MyIO.println(resp);

}

}

2)Repita o exercício acima considerando que os elementos do array estão ordenados de forma crescente. Uma sugestão seria começar a pesquisa pelo meio do array.

class Qc

{

public static boolean comp(int[] s , int num)

{

boolean resp=false;

int control = s.length;

for(int i=0 ;i<control ;i++)

{

if(num==s[control/2])

resp=true;

else

{

if(num>=s[control/2])

{

i=control/2;

else

{

if(num>=s[control/2])

control=control=2;

}

}

}

}

return resp;

}

public static void main(String[] args)

{

int[] iniciaValores = {12,32,54,6,8,89,64,64,6};

int numero=6;

boolean resp;

resp=comp(iniciaValores,numero);

MyIO.println(resp);

}

}

3)Faça um método que receba um array de inteiros e mostre na tela o maior e o menor elementos do array.

class Qc

{

public static String comp(int[] vetor , int num)

{

String resp = " " ;

int aux =0;

for (int i = 0; i < vetor.length; i++)

{

for (int j = 0; j < vetor.length; j++)

{

if (vetor[i] < vetor[j])

{

aux = vetor[i];

vetor[i] = vetor[j];

vetor[j] = aux;

}

}

}

for (int i = 0; i < vetor.length; i++)

resp = vetor[i];

return resp;

}

public static void main(String[] args)

{

int[] iniciaValores = {12,32,54,6,8,89,64,64,6};

int numero=6;

String resp = "";

resp=comp(iniciaValores,numero);

MyIO.println(resp[0] + " "+ resp[length]);

}

}

        resp=true;

    }

    return resp;

  }

  public static void main(String[] args)

  {

    int[] iniciaValores = {12,32,54,6,8,89,64,64,6};

    int numero=6;

    boolean resp;

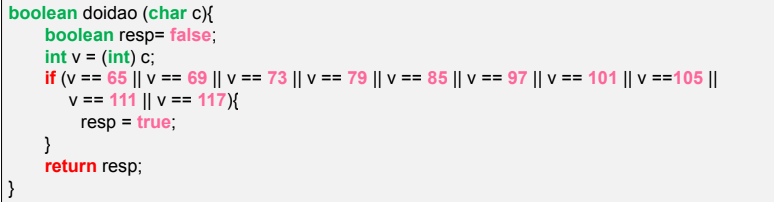
    resp=comp(iniciaValores,numero);

    MyIO.println(resp);

}

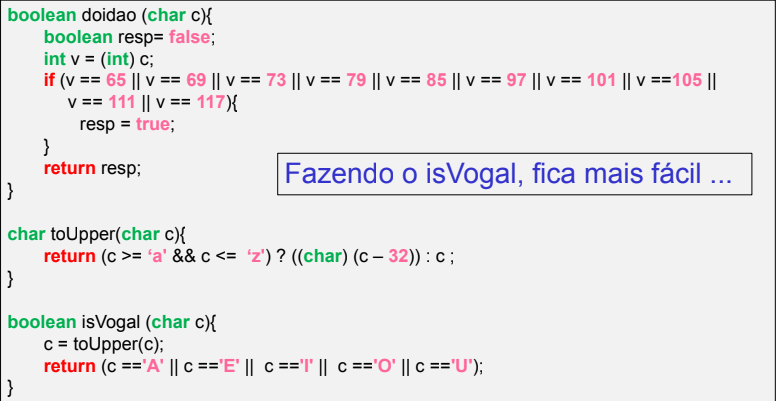
}

O que o Método a baixo faz?



O método booleano doidao recebe um char, transforma este char em um inteiro da tabelas Ascii, e verifica se esse número e uma vogal. Caso a resposta for verdadeira ele retorna true, caso contrario retorna false.

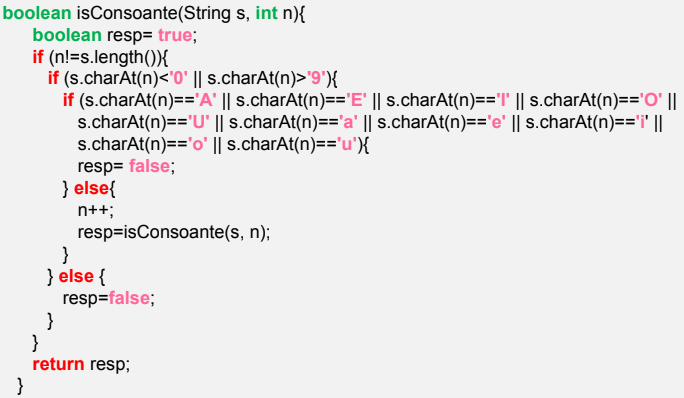
O que o Método a baixo faz?



O método booleano doidao recebe um char, transforma este char em um inteiro da tabela Ascii, e verifica se esse número e uma vogal. Caso a resposta for verdadeira ele retorna true, caso contrário retorna false.

Já o método isVogal confere apenas as vogais Maiúsculas pois ele chama o método toUpper para fazer a conversão.

Um aluno desenvolveu o código abaixo, corrija-o:



boolean isConsoante(String s, int n)

{

boolean resp= true;

if (n != s.length())

{

if (s.charAt(n)<'0' || s.charAt(n)>'9')

{

if (isVogal(s.charAt(n)) == true)

resp= false;

else

resp=isConsoante(s, n + 1);

}

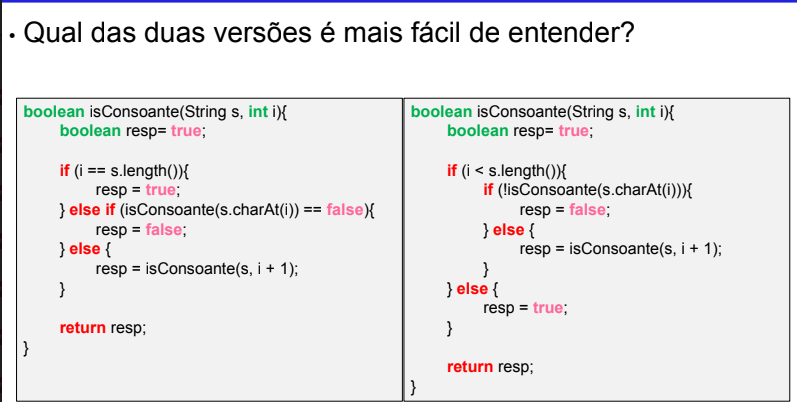
else

resp=false;

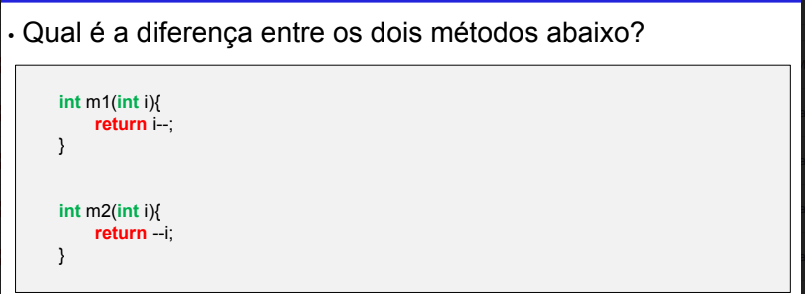
}

return resp;

}

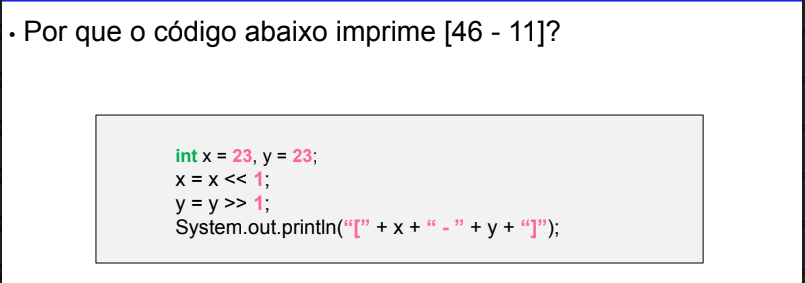


2- É mais fácil de entender pois ele esta identada.



O 1 retorna (i-1).

O 2 retorna (i).



Ele imprime 46 pois desloca um bit para esquerda dobrando seu valor

Ele imprime 11 pois desloca um bit para direita contando deu valor na metade -1.