

Instituto Politécnico de Santarém
Escola Superior de Gestão e Tecnologia
Licenciatura em Informática

Linguagens de programação

DivNum- O Jogo adivinha



Realizado por:

Sílvia Cabral 180100422@esg.ipsantarem.pt

Mamadou Saikou Diallo 170100363@esg.ipsantarem.pt

Índice

1. Introdução	3
2. Descrição do Sistema	4
3. Principais Funções do Sistema	Erro! Marcador não definido.
4. Diagrama de Casos de Utilização	Erro! Marcador não definido.
12. Conclusões.....	Erro! Marcador não definido.
13. Referências Bibliográficas	Erro! Marcador não definido.

1. Introdução

Com este trabalho pretende-se desenvolver um jogo chamado “DivNum”, o jogo tem principais funções acertar um número que na qual foi escolhido aleatoriamente pelo computador. Caso não acerte num número de 5 tentativas disponibilizamos uma opção de querer ver o número escolhido caso o jogador queira. Na elaboração deste trabalho foi utilizado o diagrama de uma árvore binária de busca que foi o essencial para a criação deste trabalho.

2. Elaboração do Jogo

Neste trabalho o primeiro passo que tivemos e foi o mais importante foi a criação de uma árvore binária de busca na qual os números estão todos ordenados em relação a raiz.

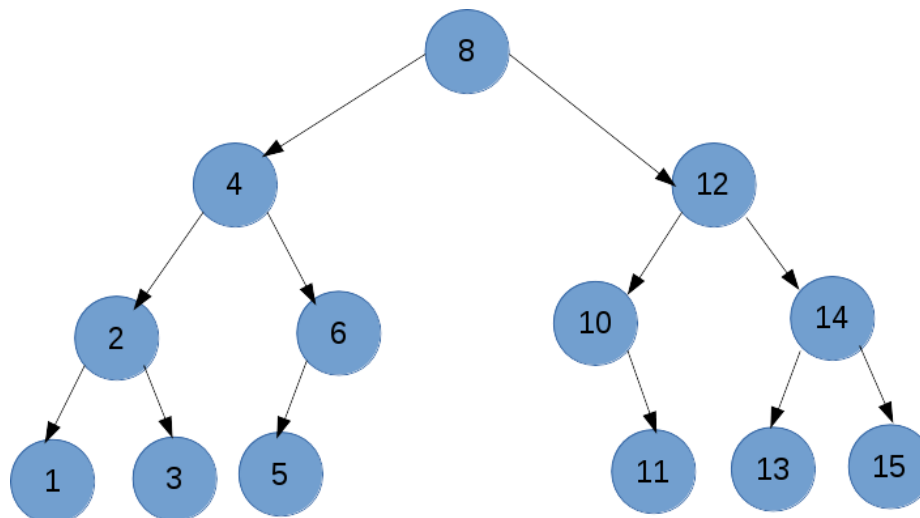


Figura 1.1- Árvore binária de busca

Como podemos ver na figura 1.1 , uma verdadeira árvore binária de busca, sabemos que a raiz é 8 e os números menores que a raiz se encontram no lado esquerdo e os maiores no lado direito, isto é apenas um exemplo para o melhor entendimento do nosso trabalho . No nosso trabalho fizemos uma árvore muito maior, e para a criação dela foram necessárias 3 classes:

- No;
- Árvore;
- Principal (Main);

A Classe No é formada da seguinte forma:

```
private int valor;
private No filhoEsquerda;
private No filhoDireita;

// Acessores
public int getValor() {
    return valor;
}
public void setValor(int valor) {
    this.valor = valor;
}
public No getFilhoEsquerda() {
    return filhoEsquerda;
}
public void setFilhoEsquerda(No filhoEsquerda) {
    this.filhoEsquerda = filhoEsquerda;
}
public No getFilhoDireita() {
    return filhoDireita;
}
public void setFilhoDireita(No filhoDireita) {
    this.filhoDireita = filhoDireita;
}

// Construtores
public No(int valor) {
    this.valor = valor;
    this.filhoEsquerda = null;
    this.filhoDireita = null;
}

// Override toString
@Override
```

Figura 1.2 Componentes da classe No

```
// Override toString
@Override
public String toString() {
    return + valor + ", filhoEsquerda=" + filhoEsquerda + ", filhoDireita=" + filhoDireita
    + "];"
}
```

Figura 1.3- Continuação da classe No

A Classe Arvore é formada da seguinte forma:

```
public class Arvore {
    private int i=0;

    private No raiz;

    public Arvore() {
        raiz = null;
    }

    public Arvore(No a) {
        raiz = a;
    }

    private boolean isEmpty() {
        if (raiz == null)
            return true;
        return false;
    }
}
```

Figura 1.4 -Classe Arvore na sua forma inicial

A nossa árvore contém 100 números (num intervalo de 0 a 100) gerados aleatoriamente pelo computador, tendo em conta que se é um jogo para adivinhar o número não podemos ter números iguais, para isto foram criados vários métodos na nossa classe árvore

- Método **PreencherArvore()**- Este método faz a criação de nós, e se a árvore estiver vazia este novo nó torna-se raiz e se não for ele chama a função existe () dentro de uma condição e se for verdade, logo a seguir chama o método adicionar() e adiciona o nó na árvore e chama logo o nosso método selecionarNumero();

```
public int PreencherArvore() {
    Random gerador = new Random();
    for (int i = 0; i < 100; i++) {
        No a = new No(gerador.nextInt(100));
        if (isEmpty()) {
            raiz = a;
        }
        else if (!existe(a.getValor())) {
            adicionar(raiz, a);
        }
    }
    return selecionarNumero();
}
```

Figura 1.5 Método preencher

- Método **adicionar (parametro1, parametro2)** – este método adiciona o nó à árvore;

```
private No adicionar(No raiz, No a) {
    if (isEmpty()) {
        raiz = a;
    }
    if (raiz.getValor() > a.getValor()) {
        // Seque pela esquerda
        if (raiz.getFilhoEsquerda() == null) {
            raiz.setFilhoEsquerda(a);
        } else {
            adicionar(raiz.getFilhoEsquerda(), a);
        }
    } else if (raiz.getValor() < a.getValor()) {
        // Seque pela direita
        if (raiz.getFilhoDireita() == null) {
            raiz.setFilhoDireita(a);
        } else {
            adicionar(raiz.getFilhoDireita(), a);
        }
    }
    return a;
}
```

Figura 1.6 Método adicionar

- Metodo **existe()**- este metodo recebe o valor de um nó e verifica se esta na arvore

```
public boolean existe(int valor) {  
    return existe(raiz, valor);  
}  
  
private boolean existe(No raiz, int valor) {  
    if (raiz == null)  
        return false;  
  
    if (raiz.getValor() == valor)  
        return true;  
  
    boolean res1 = existe(raiz.getFilhoEsquerda(), valor);  
    if (res1)  
        return true;  
  
    boolean res2 = existe(raiz.getFilhoDireita(), valor);  
  
    return res2;  
}
```

Figura 1.7 Metodo existe

- Metodo **selecaonumero()**- este metodo é chamado logo a seguir a criação da arvore e nele é selecionado o numero sorteado pelo computador e logo a seguir chama o metodo encontrar que leva o numero sorteado;

```
private int selecaonumero() {  
  
    System.out.println("=====");  
    System.out.println("=====DivNum=====");  
    System.out.println(" =====O Jogo Adivinha=====");  
    System.out.println(" =====Número=====");  
    Random gerador1=new Random();  
    int selecao=gerador1.nextInt(100);  
  
    return encontrar(selecao);  
}
```

Figura 1.8 Metodo selecao numero

- Metodo **encontrar(parametro)**- O metodo encontrar é chamado logo a seguir a execucao do metodo selecaonumero() o metodo recebe o numero e depois faz a comparacao com o numero que o usuario enviou e lá dará palpites sobre o número, e repete isso ate 7 vezes , quando as hipoteses acabarem ele faz o ultimo System.out.println e chama a funcao escolha(selecao);

```
private int encontrar(int selecao) {
    Scanner ler = new Scanner(System.in);

    while (i!=7) {
        System.out.println("Qual é o numero certo?");
        int certo=ler.nextInt();

        if(selecao==certo) {
            System.out.println("Parabéns ! Numero correcto");
            return certo;
        }

        if (selecao>certo) {
            System.out.println ("   Baixo");
        }else if (selecao < certo) {
            System.out.println("   Alto");
        }
        i++;
    }

    return encontrar(selecao);
}
```

Figura 1.9 Metodo Encontrar selecao

- Metodo **Escolha(selecao)**- Neste metodo foi opcional ele recebe o numero aleatorio , e ele só é chamado caso o utilizador perca,e aqui ele pode decidir se quer ver o resultado ou não , e logo a seguir ele recomeça o jogo;

```
private int escolha(int selecao) {

    Scanner ler = new Scanner(System.in);
    System.out.println("Deseja ver o numero certo (1 S || 0 N )?");
    int resposta=ler.nextInt();

    if (resposta ==1 ) {
        System.out.println(" Numero escolhido " + selecao);
    }
    System.out.println ("Vamos recomeçar o jogo");
    i=0;
    return PreencherArvore();
}
```

Figura 2- Metodo escolha

3. Resultados

```
=====DivNum=====
=====O Jogo Adivinha=====
=====Número=====
Qual é o numero certo?
50
    Alto
Qual é o numero certo?
45
    Alto
Qual é o numero certo?
30
    Alto
Qual é o numero certo?
20
    Alto
Qual é o numero certo?
15
    Alto
Qual é o numero certo?
10
    Alto
Qual é o numero certo?
5
    Alto
Esgotou o nivel maximo de tentativas, tente outra vez!
Deseja ver o numero certo (1 S || 0 N)?
1
| Numero escolhido 1
| Vamos recomeçar o jogo
```

Figura 2.1- O usuario perdeu

```
    Alto
Esgotou o nivel maximo de tentativas, tente outra vez!
Deseja ver o numero certo (1 S || 0 N)?
1
| Numero escolhido 1
| Vamos recomeçar o jogo
| =====DivNum=====
| =====O Jogo Adivinha=====
| =====Número=====
Qual é o numero certo?
50
    Alto
Qual é o numero certo?
20
    Baixo
Qual é o numero certo?
35
    Alto
Qual é o numero certo?
30
    Baixo
Qual é o numero certo?
32
    Baixo
Qual é o numero certo?
33
Parabéns ! Numero correcto
```

Figura 2.2 Usuario ganhou