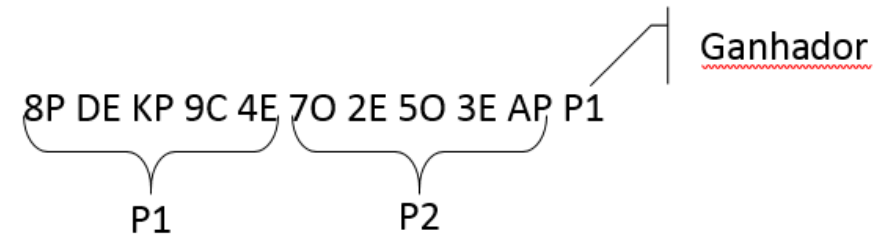


Implantação de auditoria de mão de poker na linguagem Scala

Antonio Ricardo Andrade Castelo Branco

Verônica Maria Costa dos Santo

Introdução



- Problema: Leitura de 03 arquivos contendo linhas com mãos de Poker.
- Linguagem: Java ⇔ Scala

Linguagem Scala

- Surgiu em 2001 – Martin Odersky
- Linguagem multiparadigma
 - Orientada a objetos – functional
- Tudo é tratado como objetos
- Funções podem ser passadas como parametros
- Facilmente integrada a Java
- Aplicada atualmente pelo Twitter e FourSquare
- Desvantagem:
 - Escasso material de consulta (comunidades, livros, fóruns)

Comparação do programa (Leitura do arquivo)

Java

```
private static int lerArquivo(String
nomeArquivo) throws IOException{
    String jogada = null;
    int conflito = 0;
    FileReader reader = new
    FileReader(nomeArquivo);
    BufferedReader buffer = new
    BufferedReader(reader);
    while((jogada = buffer.readLine())
    != null) {
        (...)
    }
    reader.close();
    buffer.close();
    return conflito;
}
```

Scala

```
def lerArquivo(): Array[String] = {
    scala.io.Source.fromFile("/Users/Ric
ardo/workspace/paradigma_scala/poker
M.txt").getLines().buffered.toArray
    }
def percorreArquivo(array:
Array[String]): Int = {
    if(array.length > 1){
        Main.percorreArquivo(array.tail) +
        Main.verConflito(array.head)
    }else
        Main.verConflito(array.head)
    }
}
```

Comparação do programa (Gravar arquivo)

Java

```
private static void  
gravarArquivo(String  
nomeArquivo, int conflitos) throws  
IOException {  
  
    FileWriter writer = new  
        FileWriter(nomeArquivo);  
  
    PrintWriter printer = new  
        PrintWriter(writer);  
  
    long tempoExecucao = 0;  
  
    tempoExecucao =  
        System.currentTimeMillis() -  
        inicio;  
  
    printer.printf("%d\n%d\n", conflitos,  
        tempoExecucao);  
  
    writer.close();  
}
```

Scala

```
def main(args:  
    Array[String]){  
  
    val pw = new  
        PrintWriter(new  
            FileWriter("saida.txt"));  
  
    val time =  
        System.currentTimeMillis();  
  
    pw.println(Main.percorreArq  
        uivo(Main.lerArquivo()))  
  
    pw.println(System.currentTi  
        meMillis() - time)  
  
    pw.close()  
}
```

Comparação do programa (Quicksort)

Java

```
private static char[]  
sort(char[] matriz) {  
    int tamanho =  
        matriz.length;  
  
    matriz =  
        quicksort(matriz, 0, tama  
            nho-1);  
    return matriz;  
}
```

Scala

```
def sort(array: Array[Int]):  
    Array[Int] = {  
        if(array.length <= 1)  
            array  
        else{  
            val pivot =  
                array(array.length/2)  
  
            Array.concat(Main.sort(array.  
                filter(pivot >)),  
                array.filter(pivot  
                    ==),  
                Main.sort(array.filter(pivot  
                    <)))  
        }  
    }
```

Comparação do programa (Quicksort)

Java

```
private static char[] quicksort(char[]  
matriz, int inicio, int fim) {  
  
    char pivo = matriz[inicio + (fim-  
inicio)/2];  
  
    int i = inicio;  
    int j = fim;  
  
    while(i <= j) {  
        while(valorCarta(matriz[i]) <  
            valorCarta(pivo)) { i++;}  
        while(valorCarta(matriz[j]) >  
            valorCarta(pivo)) { j--;}  
  
        if(inicio <= fim) {matriz = troca(matriz,  
i,j); i++; j--;}  
    }  
  
    if(j > inicio) {quicksort(matriz, inicio, j);}  
  
    if(i < fim) {quicksort(matriz, i, fim);}  
  
    return matriz; }  
}
```

Scala

```
def sort(array: Array[Int]):  
Array[Int] = {  
    if(array.length <= 1)  
        array  
    else{  
        val pivot =  
            array(array.length/2)  
  
        Array.concat(Main.sort(array.  
            filter(pivot >)),  
                    array.filter(pivot  
                        ==),  
                    Main.sort(array.filter(pivot  
                        <)))  
    }  
}
```

Comparação do programa (Quicksort)

Java

```
private static char[]  
troca(char[] matriz,  
int i, int j) {  
    char temp = matriz[i];  
    matriz[i] = matriz[j];  
    matriz[j] = temp;  
    return matriz;  
}
```

Scala

```
def sort(array: Array[Int]):  
Array[Int] = {  
    if(array.length <= 1)  
        array  
    else{  
        val pivot =  
            array(array.length/2)  
  
        Array.concat(Main.sort(array.  
            filter(pivot >)),  
                    array.filter(pivot  
            ==),  
            Main.sort(array.filter(pivot  
            <)))  
    }  
}
```


Comparação do programa (Flush)

Java

```
private static boolean  
flush(char[] naipe) {  
    if(naipe[0] == naipe[1]  
    && naipe[0] == naipe[2]  
    &&  
    naipe[0] == naipe[3] &&  
    naipe[0] == naipe[4]) {  
        return true;  
    }  
    return false;  
}
```

Scala

```
def flush(naipes:  
Array[Int]): Boolean =  
{  
    naipes.forall(p =>  
p == naipes.head)  
}
```

Comparação do programa (Straight)

Java

```
private static boolean  
straight(String  
sequencia) {  
  
    String naipes =  
        "23456789DQJKA";  
  
    if(naipes.contains(sequ  
encia)) {  
  
        return true;  
  
    }  
  
    return false;  
  
}
```

Scala

```
def  
straight(valorCartas:  
Array[Int]): Boolean = {  
  
    if(valorCartas.length !=  
        2){  
  
        Main.straight(valorCarta  
s.tail)  
        }  
  
        valorCartas.head + 1  
        == valorCartas.tail.head  
        }  
}
```

Resultados (Tempo)

Java

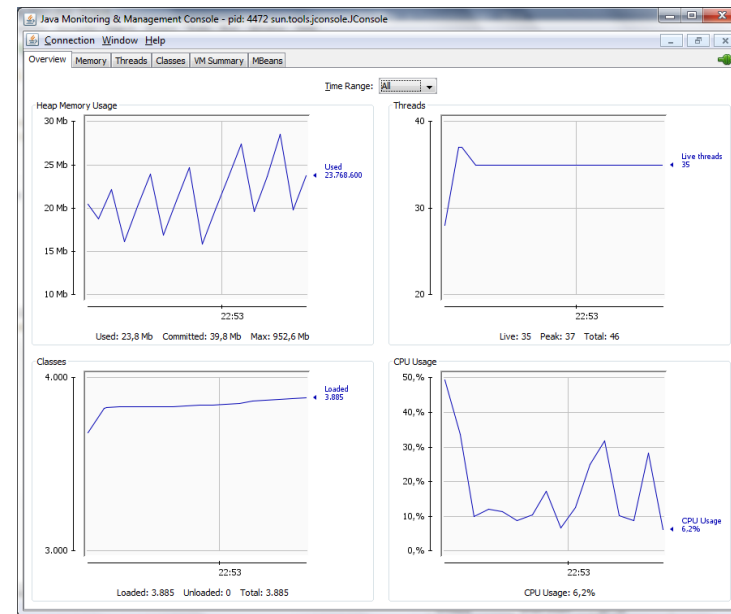
Java (Tempo em milissegundos)	pokerK	pokerM
Teste 1	115	12089
Teste 2	66	11216
Teste 3	74	10843
Média	85	11382

Scala

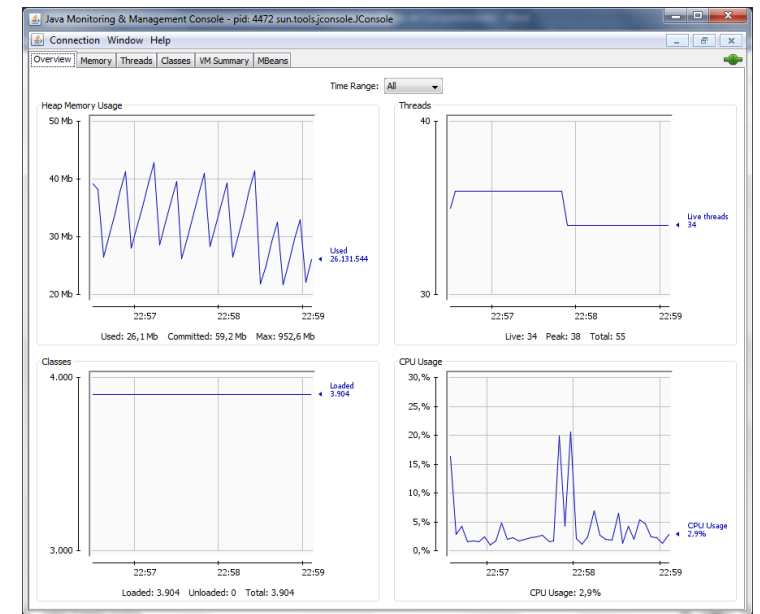
Scala (Tempo em milissegundos)	pokerK	pokerM
Teste 1	2445	Java Heap Space
Teste 2	2925	
Teste 3	4245	
Média	3205	

Resultados (Memória)

Java



Scala



Conclusões

- Uso excessivo de memória
- Simplicidade na codificação
- Multiplicidade na utilização de funções

Referências

- Schinz, Michael and Haller, Philipp (2008) “Um tutorial de Scala para programadores em Java”, Translate by Marcelo Castelliani and Thiago Rocha, Programming Methods Laboratory, Switzerland.
- Odersky, Martin (2010) “Scala através de exemplos”, Translate by Vinicius Miana and Antonio Basile, Programming Methods Laboratory, Switzerland.
- Wikipedia (2018) “Scala (Linguagem de Programação)”
[https://pt.wikipedia.org/wiki/Scala_\(linguagem_de_programação\)](https://pt.wikipedia.org/wiki/Scala_(linguagem_de_programação))
, January.