



INSTITUTO DO EMPREGO
E FORMAÇÃO PROFISSIONAL

**Curso: PROGRAMADOR/A DE INFORMÁTICA
/ NST PROG 15/**

PROGRAMAÇÃO EM JAVA

**PROJECTO 1
EFEITOS ESPECIAIS E COORDENADAS EXCEL**

Data: 15/10/2023

Formando: Olga Ignatenko

Centro de Formação de Alcântara

1. Introdução e objetivos

Este projeto tem como principal objetivo a consolidação dos conhecimentos em programação Java, adquiridos até o momento, por meio da implementação de dois programas distintos.

Na primeira parte deste projeto, propõe-se a implementação de um conjunto de algoritmos destinados a produzir "efeitos especiais" em texto. Esses efeitos visam enriquecer a apresentação e a formatação de textos de maneira criativa e envolvente.

Já na segunda parte do projeto, o foco volta para o desenvolvimento de um programa capaz de converter coordenadas no formato Excel para coordenadas lineares em uma matriz. Esse programa tem o potencial de simplificar a manipulação de dados e facilitar a integração com outras aplicações.

A realização deste projeto tem como objetivo não só fortalecer os fundamentos, mas também expandir nossas habilidades e compreensão de sistemas complexos.

2. Desenho e Estrutura

Esta secção contém os algoritmos para os Efeitos 5 e 6 em pseudo-código, utilizando Portugol. Esse pseudo-código foi criado com base no código fornecido no programa.

No final da secção é apresentado o pseudo-código que descreve a conversão das colunas do formato "letras" para o formato numérico. Essa lógica é utilizada na implementação do segundo exercício do projeto.

Pseudo-código para Efeito 5

```
// Inicializa as variáveis de controle
j <- 0
x <- Comprimento de str * 2 - 1
// Loop externo controlado por i
Para i de 0 Até Comprimento de str - 1 Passo 1
    // Loop interno controlado por spacesCount
    Para spacesCount de 0 Até Comprimento de str * 2 -
1 Passo 1
        Se (j for igual a spacesCount) ou (x for igual
a spacesCount) Então
            // Imprime o caractere em maiúsculas na
posição i de str
            Escreva(str[i] em maiúsculas)
        Senão
            // Imprime um espaço em branco
            Escreva(" ")
        Fim Se
    Fim Para

    // Imprime uma nova linha
    EscrevaLinha()

    // Atualiza as variáveis de controle
    j <- j + 1
    x <- x - 1
Fim Para
Fim Algoritmo
```

Pseudo-código para Efeito 6

```
// Define o comprimento desejado da palavra
word_length <- 40

// Concatena os elementos da lista "strings" com um
espaço e converte para maiúsculas
s <- Junte os elementos de "strings" com um espaço
s <- Converter s para maiúsculas

// Adiciona espaços em branco no final da string, se
necessário
Se (word_length - Comprimento de s > 0) Então
    EspacosEmBranco <-
RepitaEspacoEmBranco(word_length - Comprimento de s)
    s <- s + EspacosEmBranco
Fim Se

// Converte a string em um array de caracteres
myCharArray <- ConverterStringParaArrayDeCaracteres(s)

// Loop infinito
Enquanto (Verdadeiro) Faça
    // Imprime o array de caracteres
    EscrevaLinha(myCharArray)

    // Obtém o primeiro caractere
    primeiro <- myCharArray[0]

    // Move os caracteres para a esquerda
    Para i de 1 Até Comprimento de myCharArray - 1
        Passo 1
            myCharArray[i - 1] <- myCharArray[i]
```

```

        Fim Para

        // Coloca o primeiro caractere no final
        myCharArray[Comprimento de myCharArray - 1] <-
primeiro

        // Espera 1 segundo
        Esperar(1000) // Aguarda 1 segundo (1000
milissegundos)

        Fim Enquanto
Fim Algoritmo

Função RepitaEspacoEmBranco(n)
    // Retorna uma string contendo n espaços em branco
    Espacos <- ""
    Para i de 1 Até n Passo 1
        Espacos <- Espacos + " "
    Fim Para
    Retorne Espacos
Fim Função

Função ConverterStringParaArrayDeCaracteres(s)
    // Converte uma string em um array de caracteres
    Caracteres <- Novo Array de Caracteres com Comprimento
de s
    Para i de 0 Até Comprimento de s - 1 Passo 1
        Caracteres[i] <- s[i]
    Fim Para
    Retorne Caracteres
Fim Função

```

Pseudo-código para coordenadas de Excel

Na Figura 1 está apresentado o código da função desenvolvida que transforma a coordenada do Excel para formato numérico.

```
public static int[] excelCoordinateToIndices(String excelCoordinate) {
    int column = 0;
    int row = 0;
    String rowString = "";

    for (int i = 0; i < excelCoordinate.length(); i++) {
        char c = excelCoordinate.charAt(i);
        if (Character.isLetter(c)) {
            column = column * 26 + (Character.toUpperCase(c) - 'A' + 1);
        } else if (Character.isDigit(c)) {
            rowString = rowString + Character.toString(c);
            row = Integer.parseInt(rowString);
        } else {
            return null;
        }
    }
}
```

Figura 1. Função excelCoordinateToIndices.

Na base dessa função, foi desenvolvido o seguinte pseudo-código de modo a explicar melhor a sua lógica:

```
// Inicializa a variável 'column' para armazenar o
valor da coluna

column <- 0

// Lê um caractere 'c' em maiúscula

Leia c

// Converte a letra em maiúscula para um valor de
coluna

// Supõe que 'c' seja uma letra de A a Z (maiúscula)

column <- column * 26 + (AsciiDeMaiuscula(c) -
AsciiDeMaiuscula('A') + 1)

// O resultado em 'column' é o valor da coluna
correspondente à letra 'c'

// Exibe o valor da coluna

Escreva "O valor da coluna é: ", column

Fim Algoritmo
```

3. Implementação

Durante a realização desse projeto, foi utilizado o IntelliJ como o principal ambiente de desenvolvimento. Deste modo, foi criado um projeto chamado “Project1” com a seguinte estrutura:

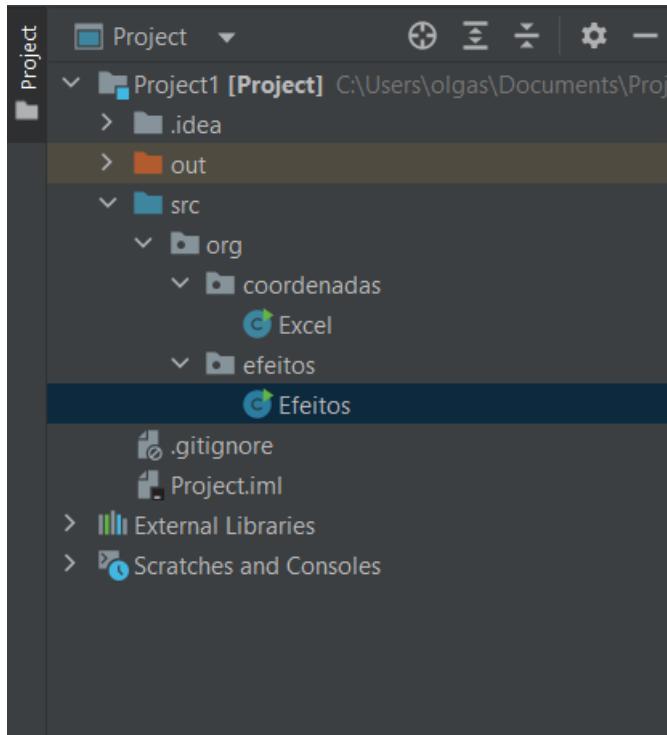


Figura 2. Estrutura do projeto.

As funções para os efeitos 1-6 estão apresentadas na class “Efeitos”, que se encontra no package *org.efeitos* (Figura 3).

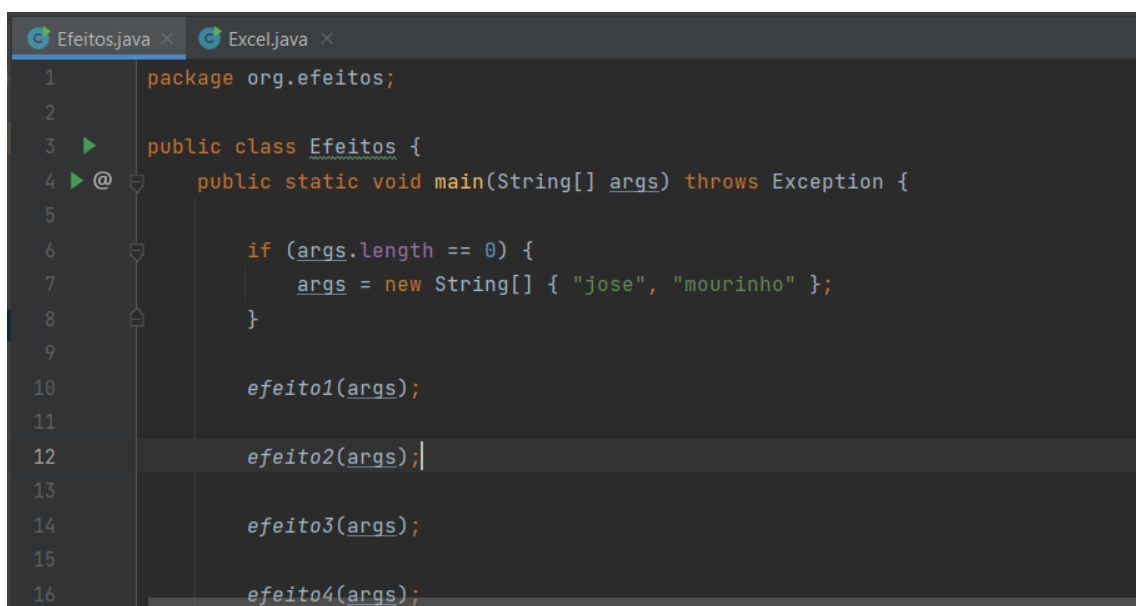
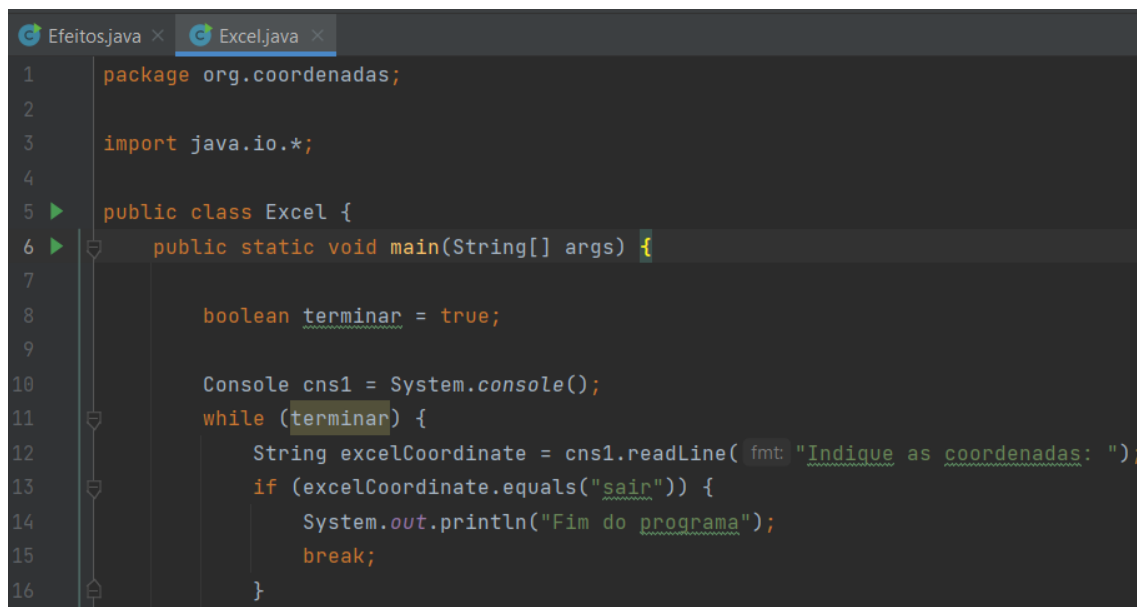


Figura 3. Class “Efeitos”

Essa classe está estruturada de seguinte forma:

- O `main()` invoca as funções responsáveis pelos efeitos 1-6. Caso não haver argumento no programa invocado, é utilizada a String “jose” “mourinho”
- As funções para os efeitos 1-6 estão descritas a seguir
- A função `changeName()` é utilizada na função `efeito6()` e é responsável pela alteração da ordem das letras na String introduzida de modo a replicar o efeito do texto deslizando

O exercício de coordenadas Excel encontra-se na class “Excel”, que por sua vez está inserida no package `org.coordenadas` (Figura 4).



```
1 package org.coordenadas;
2
3 import java.io.*;
4
5 public class Excel {
6     public static void main(String[] args) {
7
8         boolean terminar = true;
9
10        Console cns1 = System.console();
11        while (terminar) {
12            String excelCoordinate = cns1.readLine( fmt: "Indique as coordenadas: ");
13            if (excelCoordinate.equals("sair")) {
14                System.out.println("Fim do programa");
15                break;
16            }
17        }
18    }
19 }
```

Figura 4. Class Excel.

Esta class está dividida de seguinte forma:

- O `main` abre a linha dos comandos para o utilizador conseguir inserir as coordenadas pretendidas. As coordenadas entram como input na função “`excelCoordinateToIndices`” que tem como o output a linha e a coluna em formato numérico.
- A função “`excelCoordinateToIndices`” é responsável pela tradução das coordenadas “simbólicas” do Excel para as coordenadas lineares, utilizando a lógica explicada no pseudo-código na secção anterior.

De modo a compilar o programa desenvolvido, foi utilizada a função “Build” do IntelliJ, que cria uma pasta “out” com o código compilado (Figuras 5 e 6):

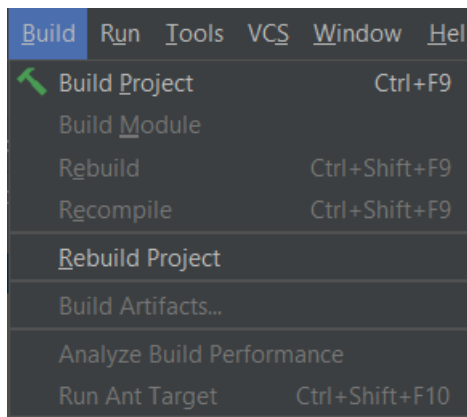


Figura 5. Realização do Build do projeto.

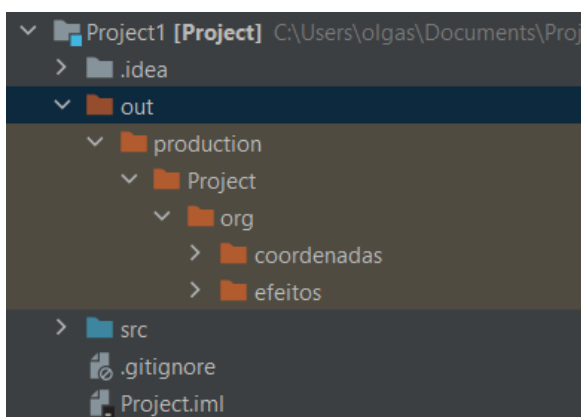


Figura 6. Estrutura da pasta "out".

Deste modo, para invocar o programa “Efeitos”, é preciso de entrar na pasta “Project” de seguinte forma:

```
C:\Users\olgas\Documents\Projeto\Project1> cd out/production/Project
C:\Users\olgas\Documents\Projeto\Project1\out\production\Project> java org.efeitos.Efeitos
```

Por outro lado, para invocar o programa “Excel”, é necessário realizar o seguinte comando:

```
PS C:\Users\olgas\Documents\Projeto\Project1\out\production\Project> java org.coordenadas.Excel
Indique as coordenadas:
```

De modo a prosseguir com a realização do programa, é necessário indicar as coordenadas do Excel no formato “AA 7”.

Os testes dos programas desenvolvidos foram realizados utilizando o terminal de IntelliJ. É de notar que para o programa de coordenadas de Excel foi utilizada a biblioteca “java.io.*” de modo a conseguir ler as coordenadas introduzidas pelo utilizador.

4. Conclusão

O desenvolvimento deste trabalho em Java permitiu atingir os objetivos estabelecidos no início deste projeto, ou seja, aplicar na prática os conhecimentos teóricos adquiridos durante o curso. Durante a implementação dos programas requeridos, foi possível aprender a importância da organização do código, boas práticas de programação, e o valor de manter o código limpo e documentado. Adicionalmente, foi possível obter um conhecimento mais profundo do Java e suas bibliotecas, o que permitiu criar um programa robusto e funcional.

É de notar que não foi possível realizar alguns pontos “extra” devido à falta do conhecimento em algumas áreas de desenvolvimento em Java.

Concluindo, a realização deste projeto Java foi uma experiência desafiante e enriquecedora, que permitiu crescer no desenvolvimento de Java.