

**Universidade Federal de Santa Catarina
Campus de Florianópolis**



**CALCULADORA JAVA
Relatório do Trabalho Final de Programação Orientada a Objetos II**

Mariany Ferreira da Silva

**2016
Florianópolis**

Introdução

Calculadoras são dispositivos para a realização de cálculos numéricos simples e/ou complexos. Essas são consideradas distintas dos computadores, no sentido de que a calculadora é um dispositivo criado para um fim específico e que não pode ser qualificada como uma Máquina de Turing.

A orientação a objetos é um modelo de análise, projeto e programação de *software* baseado na composição e interação entre unidades de *software* chamadas de objetos. Esse método de modelagem é tido como a melhor estratégia para se eliminar dificuldades no processo de modelar o mundo real do domínio do problema em um conjunto de componentes de *software*. Na programação orientada a objetos, implementa-se um conjunto de classes que definem os objetos e cada classe determina o comportamento (definido nos métodos) e estados de seus objetos, assim como o relacionamento entre objetos.

Tema escolhido

Calculadora Java

Equipe

Filipe Assunção dos Santos, Mariany Ferreira e Filipe Longarai Trisotto.

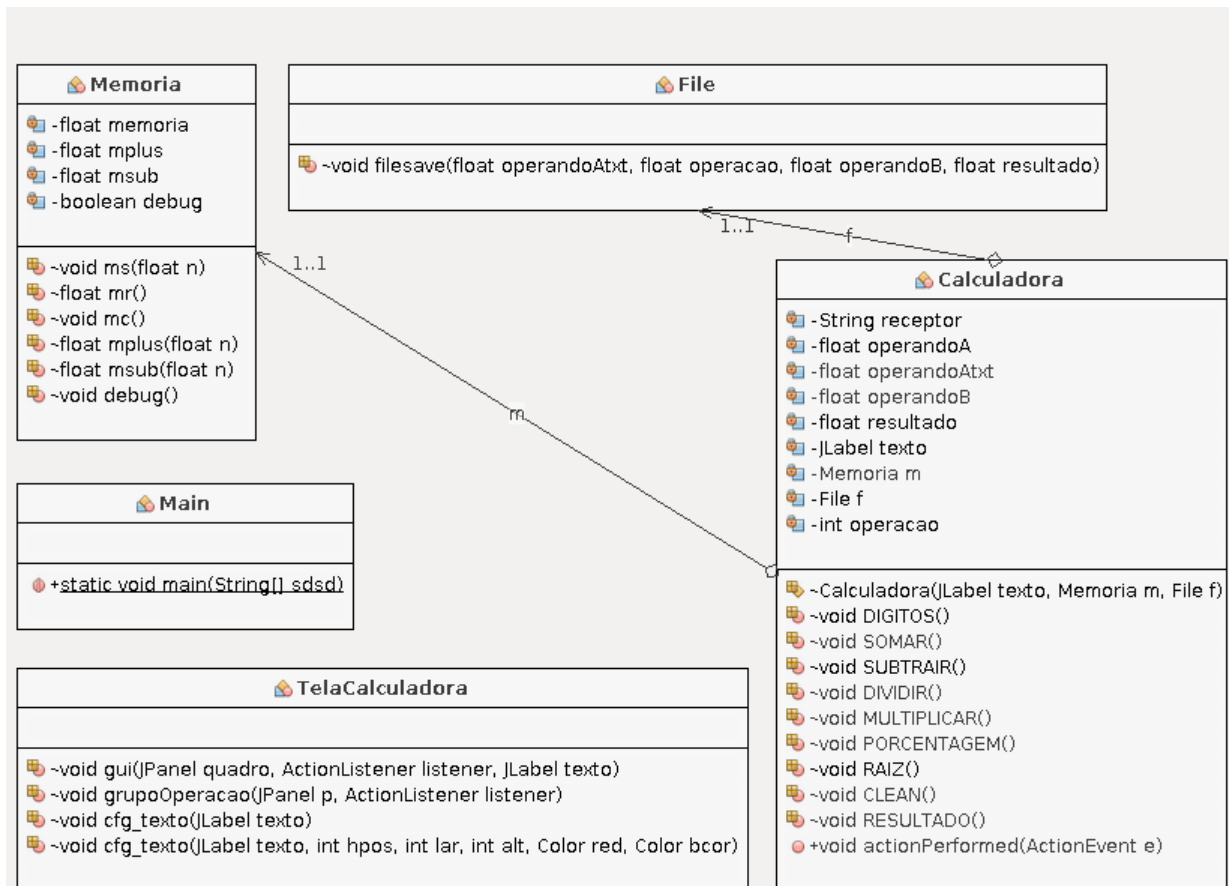
Uso do programa

A Calculadora implementada tem funcionalidades simples e possibilita o usuário de fazer operações aritméticas como adição, subtração, multiplicação, divisão, porcentagens, etc. também é possível que o usuário salve os cálculos feitos em um arquivo .txt para que use posteriormente caso necessário.

Análise do problema

O programa espera uma interação do usuário, aguardando a escolha de operandos e operações que o usuário deseja realizar, captura essas interações, realiza a operação desejada e retorna o resultado. Através do ActionListener e ActionCommand é possível capturar essas interações e utilizá-las para assim evocar os métodos correspondentes à operação selecionada para obtenção do resultado.

Arquitetura do projeto



Classes envolvidas

Main

A classe main é a classe evocada na hora da execução do programa e é ela quem realiza a chamada do frame e do panel, utilizados pela calculadora. No método main também são instanciados os objetos Calculadora, Memoria, TelaCalculadora, File, etc.

Por meio dos import `java.awt.event.*` e `import javax.swing.*`; podemos tratar algumas ações do painel necessárias para o bom funcionamento da Calculadora.

TelaCalculadora

Essa classe utiliza vários imports relacionados a Painel, Design e construção de telas esses imports são utilizados para construir a interface com o usuário, conforme a imagem abaixo. Nessa classe definimos a matriz de botões, as cores, nomes dos botões, tamanho, organização, etc. Construindo assim o canal de comunicação com o usuário.



Calculadora

Essa, se assim podemos considerar, é a classe de maior impacto nesse projeto. Nela definimos todos os métodos necessários para realizar as operações numéricas que a calculadora se propoe a fazer.

Conforme exposto na documentação Oracle o `javax.swing`, utilizado nessa classe fornece um conjunto de componentes "leves" que, na medida do possível, trabalham o mesmo em todas as plataformas. A interface de ação, `Action`, proporciona uma extensão útil para a interface `ActionListener` nos casos essa funcionalidade pode ser acessada por vários controles.

O import `java.awt.event` também foi usado pois ele fornece interfaces e classes para lidar com diferentes tipos de eventos disparados por componentes AWT.

Foram utilizadas variáveis de recepção de dados e variáveis para operar com os dados recebidos. Cada operação tem sua função declarada e retorna um resultado conforme implementado no método `RESULTADO`.

O método `actionPerformed` é chamado quando uma ação ocorre e é nele que tratamos a chamada dos outros métodos quando necessário.

Memoria

Onde foram construídos todos os métodos que realizam as operações de salvamento de resultados, adição/subtração de memória imediata, recuperação de operandos/resultados salvos e pela limpeza/reset da calculadora.

File

Fazendo uso dos importes `java.io.FileWriter` e `java.io.PrintWriter` nessa classe definimos o método que realiza a issue memória de cálculo. A implementação do método guarda em um histórico as operações realizadas pelo usuário que foram marcadas para salvamento por meio de um botão. É possível encontrar esse histórico conforme o caminho mostrado pelo `System.out.print` evocado após a realização do salvamento do arquivo.

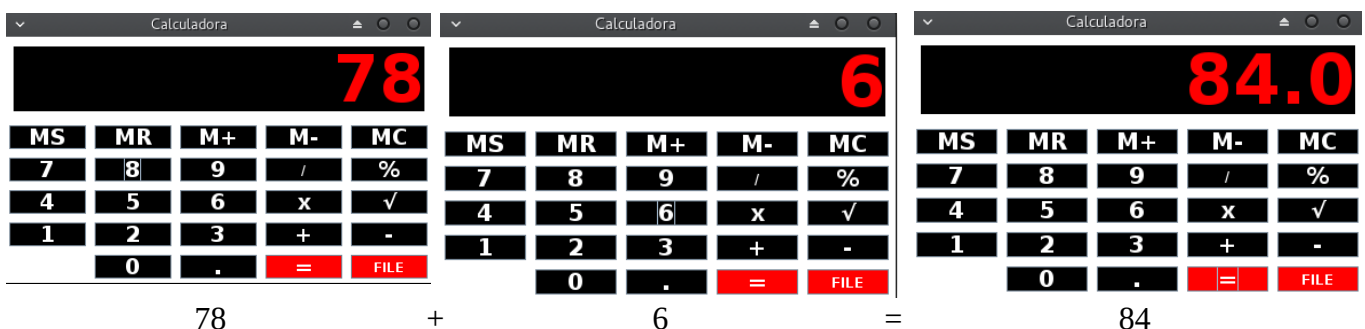
Para tratar possíveis exceções utilizamos o `import java.io.IOException` e uma mensagem é mostrada ao usuário, para isso também utilizamos o `import javax.swing.JOptionPane`.

Comportamento, cálculo e resultados

Ao abrir a Calculadora o usuário pode visualizar a interface de comunicação da Calculadora onde ele poderá escolher os operandos com quais deseja trabalhar e as operações que deseja fazer, a classe `TelaCalculadora` captura esses operandos utilizando o `ActionCommand` e os repassa para classe `Calculadora`, onde serão feitas as operações desejadas conforme a interação do usuário.

Após receber o primeiro operando a classe `Calculadora` aguarda que o usuário escolha a operação aritmética desejada e após receber a operação escolhida a `Calculadora` aguarda que o usuário escolha o segundo operando que deve ser usado no cálculo. Ao clicar no botão “=” o usuário pode visualizar na parte superior da calculadora o resultado da operação.

Todos os passos em que a calculadora recebe e envia dados para a `TelaCalculadora` é possível visualizar a interação feita na parte superior da tela.



Ao clicar em FILE, a Calculadora, que tem uma instancia da class FILE, utilizando essa instancia e chamando o metodo filisave(); passa como parametros os operandos, resultado e operação esse metodo salva a operação feita em um arquivo chamado historico.txt dentro da pasta do projeto salva o resultado, operação e operandos.

MS salva o valor escrito no Jlabel, pode ser um operando qualquer ou o resultado. Quando MS é selecionado a classe Calculadora, que tem uma instancia de Memoria chama o metodo MS(); da classe memoria, passando como parametro o texto atual do Jlabel.

MR retorna o valor salvo.

M+ faz uma adição imediata (valor salvo+ valor salvo);

M- identico só que uma subtração imediata;

MC limpeza,reset;

Conclusão

A realização desse projeto foi de grande importância para que os conhecimentos obtidos ao longo desse semestre nas aulas de Programação Orientada a Objetos II fossem utilizados para implementar de forma organizada, utilizando as ferramentas e metodologias estudadas utilizando a linguagem java.

Referencias de pesquisa

<https://docs.oracle.com/javase/7/docs/api/>