Relatório - Estrutura de Dados – Calculadora

Felipe Melchior de Britto – 18200

Gabrielle da Silva Barbosa – 18183

Objetivo:

Desenvolver uma calculadora que calcule expressões matemáticas mais complexas, utilizando de prioridades e parêntesis, em C# Visual Studio utilizando pilhas e filas. Criar métodos de conversão para as sequências matemáticas de forma que deixasse de ser “infixas” e se tornassem “posfixas”.

Desenvolvimento

18/04/19

* Criação do projeto no Visual Studio e elaboração do design da calculadora com seus respectivos botões e suas denominações.

30/04/19

* Desenvolvemos as funções dos botões da calculadora, os operadores e operandos eram escritos no txtVisor e os operandos eram separados e colocados em um vetor.
* Atribuímos uma letra para cada número do vetor descrito acima, para que as utilizássemos para a elaboração da conversão para posfixa.
* Com as letras nos lugares dos números, uma sequência infixa é gerada, sendo esta, a sequência que será utilizada para a realização da conversão.
* Desenvolvemos o método “DeveDesempilhar” que fazia a comparação entre operadores de uma expressão, existe método verifica entre dois operadores (um no topo da pilha e outro lido), qual possui prioridade de ser realizado com antecedência e portanto deveria ser desempilhado e escrito primeiramente na sequência posfixa.
* Começamos a desenvolver o método para conversão de sequência infixa para posfixa com base nos métodos apresentados a nós durante as aulas, fazendo pequenas mudanças conforme víamos a necessidade de uma melhor adaptação ao nosso projeto.

05/05/2019

* Finalizamos o desenvolvimento do método que realiza a conversão. Tivemos uma boa quantidade de problemas e erros inesperados, mas conseguimos resolvê-los de forma eficiente ao depurar o método linha por linha encontrando suas deficiências e consertando-as.
* Durante os testes, diversos outros erros foram encontrados no método “DeveDesempilhar”. Constatamos que não estamos verificando as prioridades de forma correta e, portanto, reelaboramos nossa tabela de prioridade dos operadores e depois passamos as adaptações ao programa.
* Depois da realização dos testes e tudo estar funcionando de acordo, passamos os resultados das sequências posfixas e infixas para seus respectivos “labels” para que assim o usuário do programa possa visualizá-las.
* Ao fim das conversões, começamos o desenvolvimento do método de cálculo da expressão matemática com base em uma sequência posfixa. Sendo este, o método o qual tivemos menos problemas em programar e menos necessidade de adaptação ao nosso código. Exibimos o resultado do cálculo no text box txtResultados como solicitado no enunciado.
* Para que o usuário pudesse digitar através do teclado, não apenas dos botões, adaptamos os métodos para verificar se não havia caracteres que não deveriam estar numa expressão e também utilizamos do evento TextChanged para que, quando digitado '=', fosse para o mesmo evento do click do botão.