Colégio Técnico de Campinas - COTUCA
1º Ano - Informática

Relatório:
Projeto I

TI121 - Técnicas de Programação
Prof. Francisco Da Fonseca Rodrigues

Alunos:

Matheus Ferreira Fagundes - 23149
Yasmin Victória Lopes da Silva - 23582

1. Introdução

Este projeto foi desenvolvido como parte das atividades da disciplina de Técnicas de Programação, com o objetivo de aplicar os conhecimentos ensinados em sala por meio da criação de um programa com interface textual em Python. O principal objetivo foi a implementação de métodos numéricos, que envolvem repetição de ações e tratamento de dados, desenvolvendo o raciocínio lógico em equipe e a prática da programação orientada a objetos.

Existem quatro funcionalidades no programa, onde o usuário acessa por meio de um seletor de opções: 1o - identificação de números primos em um intervalo; 2o - cálculo da raiz quadrada por aproximação; 3o - geração de termos da sequência de Fibonacci; 4o - processamento de dados armazenados em arquivos texto. Para isso, foram utilizadas e adaptadas classes específicas que utilizam alguns conceitos como somatórios e produtórios.

Utilizamos o Github para salvar as versões do projeto:cc23149/projeto1_tp: Projeto1 de TP

2. Desenvolvimento

2.1 Interface com o usuário e suas funcionalidades

A interface é o meio de comunicação entre o programa e o usuário sendo totalmente necessária para o recebimento dos dados e a devolução das respostas esperadas que são a base do funcionamento do programa.
A interface com usuário é feita praticamente toda pelo terminal do Visual Studio, com exceção da 4º funcionalidade que é usado uma caixa de diálogo para que seja possível que o usuário escolha o arquivo com os dados a serem lidos como entrada, e de saída gera um arquivo HTML. Ao iniciar o programa é exibido ao usuário o seletor das opções disponíveis para a execução com a numeração correspondente a cada uma e é pedido para que seja digitado o número correspondente a funcionalidade desejada. Assim que recebe a informação esperada, o terminal é limpo e é exibido a funcionalidade escolhida e o pedido dos dados necessários para chegar nos resultados desejados, e logo após os resultados são exibidos e seguido do aviso para apertar ENTER e voltar ao seletor onde pode escolher outra funcionalidade ou encerrar o programa.

2.2 Tempo de Desenvolvimento

- 02/04 (quarta-feira) das 13:30 às 16:40
- 03/04 (quinta-feira) das 13:30 às 17:30
- 11/04 (sexta-feira) das 15:00 às 16:40
- 14/04 (segunda-feira) das 9:00 às 10:00 -12:40 às 14:15
- 15/04 (terça-feira) das 12:30 às 15:45
- 17/04 (quinta-feira) das 21:00 às 22:40
- 18/04 (sexta-feira) das 20:20 às 22:30

Tempo total: 18 horas e 30 minutos

2.3 Erros encontrados e soluções aplicadas

Durante o desenvolvimento do projeto, diversas versões intermediárias foram criadas até se chegar a um código funcional, modularizado e mais limpo. Abaixo listamos os principais erros encontrados ao longo do processo e como foram resolvidos:

O primeiro erro que nos deparamos foi no método fazPrimos() na qual iniciamos da variável de contagem do comando de repetição (for) repetição com 3 e incrementamos de 2 em 2 fazendo com que o método considere alguns múltiplos de 4 primos, como solução ajustamos o valor inicial da variável para 2 e passamos a incrementar de 1 em 1. Logo após, ao configurarmos as linhas deparamos com o segundo erro onde a vírgula aparecia após o último número exibido e para tratar colocamos uma verificação, para caso for o último do intervalo, seja exibido apenas a variável de contagem sem a concatenação.
O terceiro erro foi encontrado quando ao executar o método fazRaizQuadrada() e notamos que todas a raízes resultaram em 1, para resolver passamos a instanciar o objeto da classe RaizQuadrada dentro do comando de repetição (while) assim, a cada incremento um objeto que chamava o método da classe que calcula a raiz do valor do contador a cada incremento era instanciado.
O quarto erro foi percebido durante os testes no método fazFibonacci(), pois o escolhermos ver apenas o primeiro número da sequência dava erro, já que na classe Fibonacci atribuímos a segunda posição do vetor com N posições o valor fixo 1 que no vetor criado não existia, para corrigirmos colocamos uma verificação para que fosse atribuído o valor a segunda posição apenas quando N fosse maior que 1.
No quinto erro não tínhamos colocado o comando de repetição para ler as linhas e o método estava tentando converter em float o caracter que usamos para iniciar a variável linha e para compor acrescentamos o comando de repetição e ajustamos o método.

2.4 Imagens/telas de erros

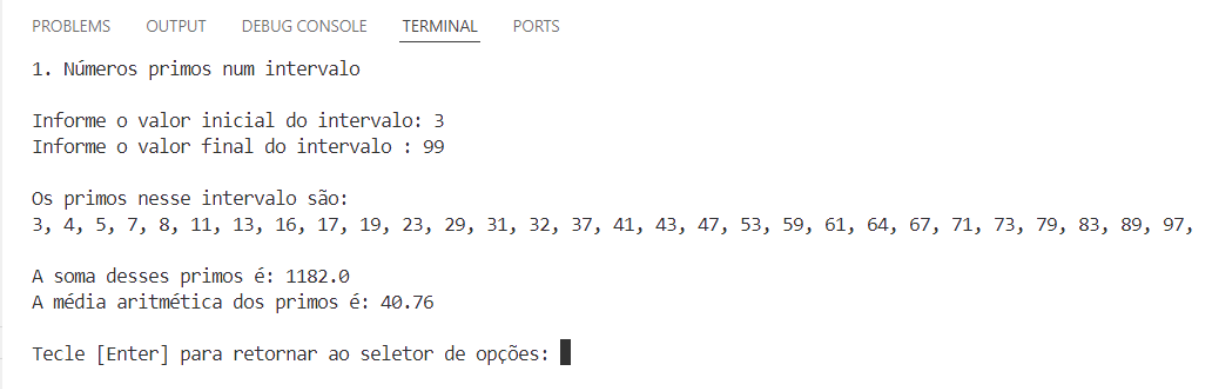

Imagem 1: Erro no método fazPrimos() em que estava considerando os múltiplos de 4 como primo (as linhas ainda não tinham sido configuradas).

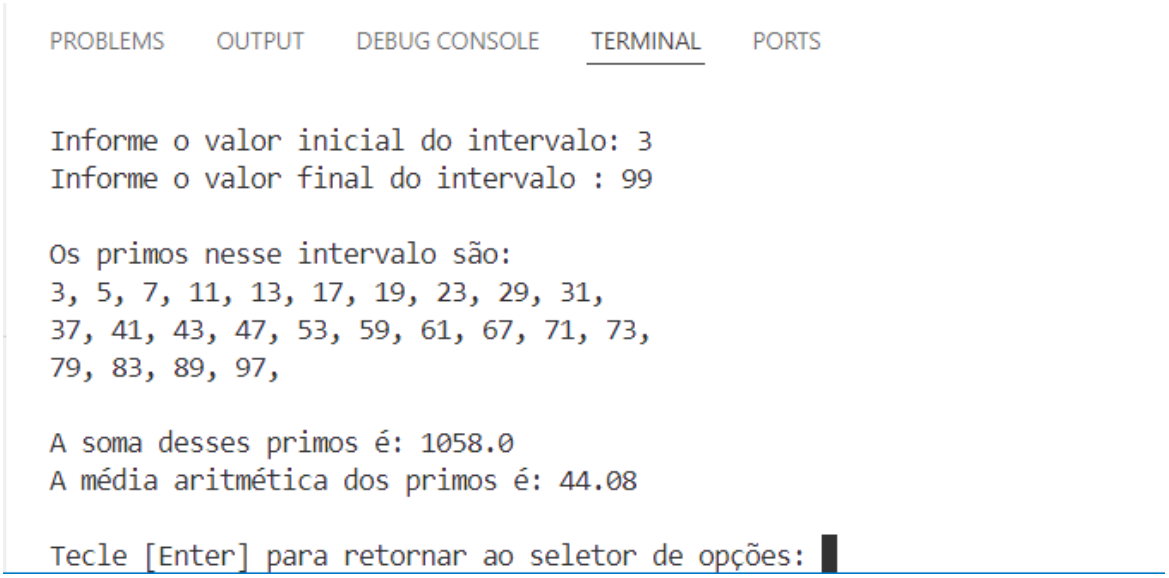

Imagem 2: Erro na configuração do print do método fazPrimos() em que a vírgula estava aparecendo depois do último número.

.

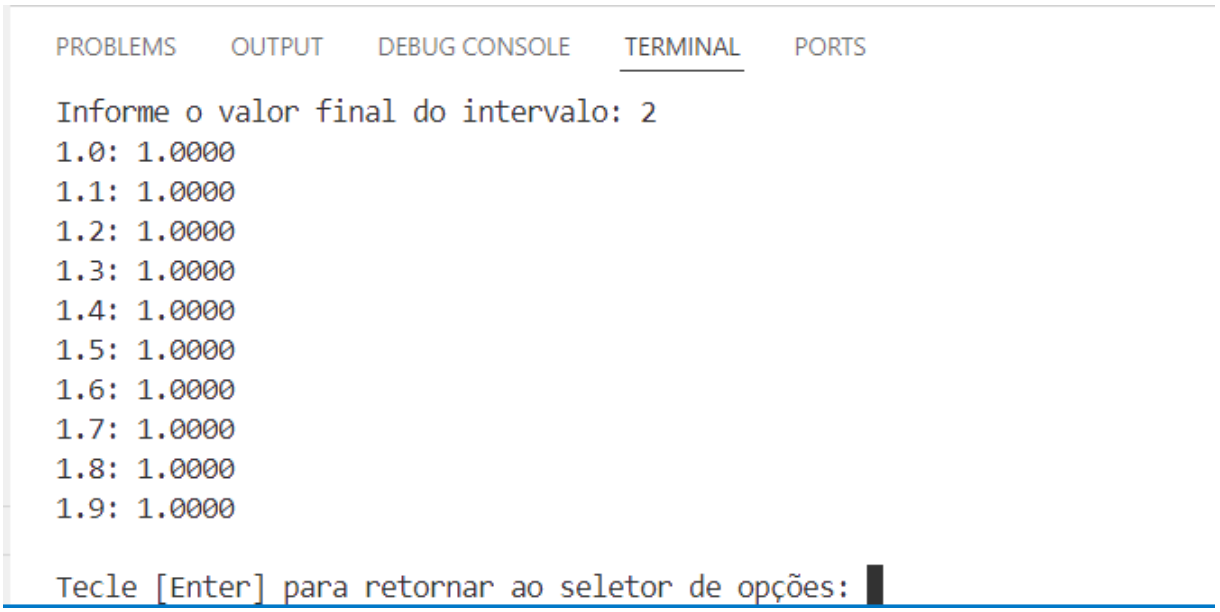

Imagem 3: Erro no método FazRaizQuadrada() em que a raiz de todos os números do intervalo estava resultando em 1.

Imagem 4: Erro na classe Fibonacci na qual quando era recebido o valor 1 como parâmetro o programa era abortado por ter um valor em uma posição não existente no vetor criado

Imagem 5:Erro no método fazProcessamentoDeDados() na qual o método não estava lendo os valores no arquivo, então estava tentando converter o valor inicial da variável linha (-) em float.

2.5 Dificuldades encontradas

Algumas dificuldades enfrentadas incluíram a separação da lógica entre classes e métodos, o uso de arquivos de texto e a geração correta do HTML. Também foi desafiador aplicar a lógica dos métodos numéricos em uma estrutura orientada a objetos.

2.6 Auxílio da monitoria

A monitoria foi de extrema importância para o desenvolvimento do projeto auxiliando no entendimento dos erros cometidos e a melhor maneira de consertá- los, as explicações e exemplos para conseguirmos implementar a terceira e quarta funcionalidades e para tirar dúvidas sobre a eficiência do código.

3. Conclusão

O projeto foi muito bom para colocarmos em prática os conhecimentos ensinados em aula como orientação a objetos, estruturas de repetição, manipulação de arquivos e o auto reconhecimento de nossas capacidades. Pudemos entender a importância da organização do código e a reutilização de componentes como a de Somatoria e Produrio. Além disso, o desafio de gerar uma saída em HTML, trouxe uma nova perspectiva de como podemos aplicar os conhecimentos das matérias Técnicas de Programação e Desenvolvimento para Internet em um projeto de forma estruturada.