

Empréstimo de Bicicletas

Guilherme Sanches Cavazzotto () e Gustavo Lösch do Amaral (241065200)

A seguinte aplicação foi elaborada como trabalho final na disciplina de Fundamentos de Programação na PUCRS.

Link do Vídeo de Apresentação - (https://youtu.be/b3_F7KVp6P4)

Descrição:

A aplicação elaborada tem como objetivo **gerenciar um clube de empréstimo de bicicletas**, realizando diversos tipos de operações entre os objetos cadastrados. O projeto foi estruturado de maneira hierárquica, dessa forma, a complexidade de interpretação do código foi reduzida. A criação de 5 classes foi necessária para a realização do projeto, sendo uma delas a aplicação principal (Clube), dois objetos (Membro e Bicicleta) e duas classes complementares (CadastroMembro e CadastroBicicleta) que realizam operações com os objetos.

As operações presentes no software foram cuidadosamente elaboradas para que nenhum erro cometido pelo usuário, como inserir valores errados, afete o funcionamento do sistema. Para que isso fosse possível, o grupo utilizou métodos de tratamento de excessões, como estruturas de "try / catch" e estruturas "if". Segue abaixo um exemplo:

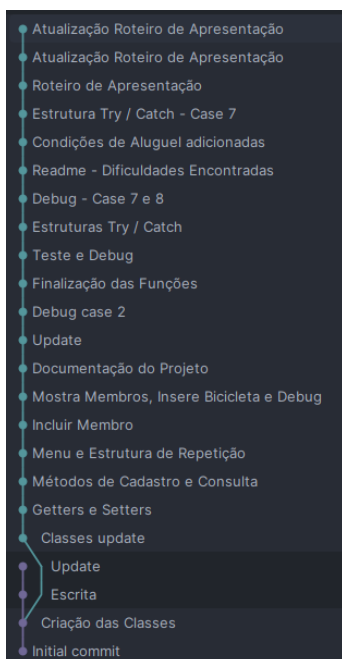
```
int buscar_codigo = t.nextInt();
try {
    cadastroBicicleta.buscaBicicletaPeloCodigo(buscar_codigo).imprimir();
}
catch (NullPointerException e) {
    System.out.println("Não existem bicicletas cadastradas com esse código.");
}
```

Segue abaixo a lista de operações possíveis de se fazer com a aplicação:

1. Inserir Membros
 2. Mostrar Membros
 3. Buscar Membro por Nome
 4. Inserir Bicicleta
 5. Mostar Bicicletas
 6. Buscar Bicicleta por Código
 7. Emprestar Bicicleta
 8. Devolver Bicicleta
 9. Quantidade de Bicicletas Disponíveis
 10. Encerrar
-

Lições Aprendidas:

Diversas lições foram aprendidas em múltiplos âmbitos durante a elaboração do trabalho. Logo no início do trabalho, o grupo entendeu que, por se tratar de uma aplicação relativamente grande, a organização era fundamental para se obter um bom trabalho em equipe e para que ninguém ficasse para trás. Aprendemos também, que a comunicação precisa, com o intuito de explicar o funcionamento de algo, entre os integrantes era essencial, uma vez que foi necessário explicar o funcionamento do que cada um fez. Como solução para a organização do projeto, decidimos utilizar o software de versionamento "git", além de comentar os pontos mais importantes do código.



Dificuldades Encontradas:

Encontramos diversos desafios durante o processo de desenvolvimento da aplicação. O primeiro problema encontrado refere-se a um erro **"NullPointerException"**, o qual significa, em nosso contexto, que foi tentado acessar as informações de um objeto que não existe. Em nosso caso, tanto o vetor de objetos de membros quanto o de bicicletas eram inicializados apenas com elementos nulos e, a medida que fosse sendo preenchido, os valores nulos deixariam de existir. Portanto, foi criada uma regra de excessão utilizando um if simples que somente realizaria operações se o objeto lido não fosse nulo.

Além disso, também nos deparamos com o erro **"InputMismatchException"**, o qual se refere a um erro de diferença entre o input recebido e o input esperado pelo Scanner. Para solucionar o problema, utilizamos o tratamento de excessão por **"try / catch"**. Alguns pequenos erros de lógica também foram encontrados, e para solucionarmos, desenhamos um fluxograma do programa que nos ajudou a entender como deveríamos agir.

Referências:

- Link do Repositório no GitHub - (<https://github.com/gustavo-losch/Emprestimo-de-Bicicletas>)
- Link do Fluxograma - (<https://www.tldraw.com/r/fmlxcLgRhRuJqkQuYGxa1?v=-472,-93,3637,1726&p=sc4ZYX7zrPZK9jlxCH1YJ>)
- Oracle Documentation | Erros - (<https://docs.oracle.com/javase/8/docs/api/java/util/InputMismatchException.html>)
- Stack Overflow | Erros - (<https://pt.stackoverflow.com/questions/63617/o-que-%C3%A9-a-nullpointerexception-e-quais-s%C3%A3o-suas-principais-causas>)
- DevMedia | Excessões - (<https://www.devmedia.com.br/blocos-try-catch/7339>)