

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DE
COMPUTAÇÃO

LEONARDO DALCIN
PAULO RICARDO RAMOS DA ROSA
PIETRO DEGRAZIA

Jogo Pedagógico em C++

Relatório apresentado como requisito parcial para
a obtenção de conceito na Disciplina de Modelos
de Linguagens de Programação

Prof. Dr. Lucas Mello Schnorr
Orientador

Porto Alegre
2018

SUMÁRIO

1 INTRODUÇÃO	3
1.1 Projeto.....	3
1.2 Sobre o jogo	3
2 C++17	4
3 IMPLEMENTAÇÃO	5
3.1 Orientação a Objetos	5

1 INTRODUÇÃO

Este trabalho tem como objetivo a implementação em C++ de um jogo pedagógico que visa ensinar diversos conceitos da cadeira de Modelos de Linguagens de Programação em diferentes níveis de profundidade. O programa permitirá que o usuário compartilhe seus resultados e confira respostas de outros colegas, assim como explicações e links para referências sobre o assunto.

1.1 Projeto

A proposta do projeto consta em, dada uma linguagem de programação escolhida pelo grupo dentre as pré-selecionadas pelo professor, desenvolver uma aplicação com duas implementações: uma puramente orientada a objetos e outra puramente funcional, contendo alguma forma de paralelismo. C++17 será a linguagem utilizada para ambas as versões.

1.2 Sobre o jogo

O projeto será um jogo de perguntas e respostas sobre a disciplina de Modelos de Linguagem de Programação, aos moldes dos programas "Show do Milhão" e "Quem quer ser um milionário?", com a objetivo de ajudar o estudante na disciplina. O jogo contará com explicações sobre o conteúdo das perguntas e poderá ser feito a consulta das respostas dos outros jogadores.

2 C++17

A linguagem C++ foi introduzida inicialmente em 1985 por Bjarne Stroustrup, herdando as características da linguagem C mas adicionando o conceito de Orientação a Objetos. Desde então, passou por diversas atualizações, estando hoje padronizada na versão ISO/IEC 14882:2017, informalmente conhecida como C++17. C++ possui 4 tipos de gerenciamento de memória:

- Armazenamento de duração estática: são criados antes da chamada `main()` (salvo exceções) e destruídos na ordem reversa de criação após a saída de `main()`.
- Armazenamento de duração de thread: Similar ao armazenamento estático, porém o objeto é criado com a thread e destruído com o `join` da thread.
- Armazenamento automático: Variáveis automáticas são criadas no ponto de declaração e destruídas na ordem reversa de criação de seu escopo. São alocadas automaticamente na pilha.
- Armazenamento dinâmico: São criadas com uma chamada `new` e destruídas com uma chamada `delete`.

Sendo uma linguagem com suporte a Orientação a Objetos, possui classes, objetos, herança, polimorfismo, encapsulamento, sobrecarga, sobrescrita e tratamento de exceções.

3 IMPLEMENTAÇÃO

3.1 Orientação a Objetos

A implementação inicial inclui duas classes bases:

- Jogador: Classe que será utilizada para criar informações sobre o jogador, como nome e pontuação.

Atributos: `std::string nome`, `int pontuacao`.

Métodos: Construtor, `getPontuacao`, `setPontuacao(int pontuacao)`.

- Pergunta: Classe que conterá uma pergunta com suas alternativas e resposta.

Atributos: `std::string enunciado`, `std::string alternativas[4]`, `int resposta`.

Métodos: Construtor, `showPergunta()`: mostra a pergunta e as alternativas, `responder(int alternativa)`: recebe a resposta do usuário, `perguntar()`: realiza a pergunta com os métodos `showPergunta()` e `responder(int alternativa)`.

Além dessas classes, uma classe `Database` está incluída para gravar dados do programa, essencialmente as perguntas, e uma classe `Main` para execução do programa.