



Paradigmas de Programação

Trabalho 1 – Características das linguagens descritivas

Kellyton Brito
kellyton.professor@gmail.com
21/07/2017



Aulas Passadas

- 1ª parte do curso: Características das linguagens descritivas
 - Variáveis
 - Tipos de dados
 - Expressões, atribuições
 - Estruturas de controle
 - Subprogramas
 - Tipos abstratos de dados e encapsulamento



Objetivos

- Implementar a solução de problemas através do uso das linguagens de programação
- Implementar, discutir e experimentar na prática como os aspectos estudados na disciplina são implementados nas linguagens descritivas
- Discutir e experimentar na prática como escolhas de design podem impactar no desempenho da aplicação



Proposta

1. Solucionar o problema através de uma implementação
 - Escolha livre de linguagem do paradigma descritivo ou derivados
2. Demonstrar na implementação cada um dos tópicos estudados: *Variáveis, Tipos de dados, Expressões, atribuições, Estruturas de controle, Subprogramas, Tipos abstratos de dados e encapsulamento*, além dos critérios estudados no início da disciplina (critérios de seleção/avaliação de linguagens)



Proposta

3. Testar, na prática, diferenças de performance:

Diferentes escolhas de:

- Tipos de dados
- Estruturas de controle
- Formas de encapsulamento
- Passagem de parâmetros
- Etc

Alteram o tempo de execução e/ou a quantidade de memória gasta?



Proposta

Projeto prático:

Solucionador de mãos de poker

Dadas duas mãos (dois conjuntos) de cartas,
identificar qual a vencedora.



Solucionador de Mãos de Poker

- 5 cartas em cada mão, cada carta tem um naipe
- Os jogos são valorados nessa ordem:
 - **Carta alta:** Carta com maior valor.
 - **Um par:** duas cartas com o mesmo valor.
 - **Dois pares:** dois pares diferentes.
 - **Trinca:** três cartas com o mesmo valor.
 - **Seqüência (Straight):** Todas as cinco cartas com valor consecutivo.
 - **Flush:** Todas as cartas do mesmo naipe.
 - **Full House:** uma trinca e um par
 - **Quadra:** quatro cartas do mesmo valor
 - **Straight Flush:** uma sequência com o mesmo naipe.
 - **Royal Flash:** 10, Valete, Dama, Rei e Ás do mesmo naipe.
- As cartas são valoradas na seguinte ordem: 2, 3, 4, 5, 6, 7, 8, 9, **Ten**, **Jack**, **Queen**, **King**, e **Ace**.
- As cartas podem ser de 4 naipes: H, C, S, D



Solucionador de Mãos de Poker

- 5 cartas em cada mão, cada carta tem um naipe
- Os jogos são valorados nessa ordem:

- **Carta alta:** Carta mais alta
- **Um par:** duas cartas iguais
- **Dois pares:** dois pares de cartas iguais
- **Trinca:** três cartas iguais
- **Seqüência (S):** cartas em sequência
- **Flush:** Todas as cartas do mesmo naipe
- **Full House:** um par e uma trinca
- **Quadra:** quatro cartas iguais
- **Straight Flush:** cartas em sequência do mesmo naipe
- **Royal Flush:** 10, valeto, dama, rei e As do mesmo naipe.

Vence o jogador com
melhor jogo

- As cartas são valoradas na seguinte ordem: 2, 3, 4, 5, 6, 7, 8, 9, **Ten**, **Jack**, **Queen**, **King**, e **Ace**.
- As cartas podem ser de 4 naipes: H, C, S, D



Descrição do problema

- Jogos diferentes: Vence o jogador com melhor jogo
- 2 jogadores com o mesmo tipo de jogo, ganha o jogo com a maior carta. Exemplo: 1 par de 8 é maior do que 1 par de 5
- 2 jogadores com o mesmo tipo de jogo e mesma maior carta no jogo (exemplo, ambos com par de rainha), é comparada a maior carta de cada mão, se necessário a segunda maior, e assim por diante.



Descrição do problema

- Exemplos:

Hand	Player 1	Player 2	Winner
1	5H 5C 6S 7S KD Pair of Fives	2C 3S 8S 8D TD Pair of Eights	
2	5D 8C 9S JS AC Highest card Ace	2C 5C 7D 8S QH Highest card Queen	
3	2D 9C AS AH AC Three Aces	3D 6D 7D TD QD Flush with Diamonds	
4	4D 6S 9H QH QC Pair of Queens Highest card Nine	3D 6D 7H QD QS Pair of Queens Highest card Seven	
5	2H 2D 4C 4D 4S Full House With Three Fours	3C 3D 3S 9S 9D Full House with Three Threes	



Descrição do problema

- Exemplos:

Hand	Player 1	Player 2	Winner
1	5H 5C 6S 7S KD Pair of Fives	2C 3S 8S 8D TD Pair of Eights	Player 2
2	5D 8C 9S JS AC Highest card Ace	2C 5C 7D 8S QH Highest card Queen	Player 1
3	2D 9C AS AH AC Three Aces	3D 6D 7D TD QD Flush with Diamonds	Player 2
4	4D 6S 9H QH QC Pair of Queens Highest card Nine	3D 6D 7H QD QS Pair of Queens Highest card Seven	Player 1
5	2H 2D 4C 4D 4S Full House With Three Fours	3C 3D 3S 9S 9D Full House with Three Threes	Player 1



Teste da Solução

- Serão disponibilizados 3 arquivos no grupo da disciplina:
 - 1.000 mãos de jogos de poker
 - 1.000.000 de mãos de jogos de poker
 - 100.000.000 de mãos de jogos de poker
- Entrada: Arquivos na pasta corrente (pokerK.txt, pokerM.txt, poker100M.txt)
- Programa deve executar (sem interação com o usuário) e retornar duas linhas:
 - Em quantas dessas mãos o jogador 1 vence?
 - Tempo gasto para execução do programa, em milissegundos



Entrega

1. Solução desenvolvida
2. Apresentar como o problema foi resolvido
3. Relacionar o uso dos aspectos estudados (variáveis, tipos de dados, expressões e atribuições, estruturas de controle, subprogramas, tipos abstratos e encapsulamento, etc) na solução
4. Testes comparativos de performance



Entrega

- Relatório (no formato da Sociedade Brasileira de Computação) :
 - Solução desenvolvida
 - Link para download do código fonte e do executável
 - Instruções de como executar a aplicação (ambiente necessário, processo de instalação/execução, etc)
 - Apresentar como o problema foi resolvido
 - Explicar o código da aplicação e os principais trechos. Utilizar diagramas se necessário
 - Relacionar o uso dos aspectos estudados (variáveis, tipos, etc) na solução
 - Mostrar como cada conceito estudado foi implementado, relacionando com trechos do código fonte (com *printscreen*)
 - Testes comparativos de performance destacando:
 - Diferentes escolhas de tipos dados/estruturas de controle/formas de encapsulamento/etc alteram o tempo de execução e/ou a quantidade de memória gasta?
 - Ps.: Para isso deve-se ter mais de uma versão do programa (mínimo 3, ideal 5).



Considerações

- Pode ser feito na web.
- Para os testes de performance:
 - Tempo: pegar a hora do sistema quando começar, quando terminar e tirar a diferença.
 - Rodar 3 vezes, e tirar a média (colocar no relatório **todos** os tempos obtidos)
 - Espaço: utilize a ferramenta embutida no interpretador/máquina virtual da linguagem que estiver utilizando (se for o caso), OU o gerenciador de memória do sistema operacional.



Considerações

- Serão rodadas pelo menos 3 ferramentas de detecção de clones/plágio
 - Se detectado $> 50\%$ de clone (de outros trabalhos ou da net), será zerado.
- Mesmo trabalho do semestre anterior: será tolerado apenas auto-plágio.
- *Lembrem-se: eu também tenho google*



Entrega

- Este trabalho valerá 60% da 2VA
- Grupos de 0 a 4 pessoas
- Entrega até 08/08/2017
 - ***Firm deadline***
- Por e-mail, para kellyton.brito@gmail.com
 - Assunto: [Paradigmas] *<Nome dos integrantes da equipe>*
- Entregas posteriores com penalidade de 25% para cada 6h incompletas de atraso



Nossa Agenda

- 25/jul: Terça: Definição do trabalho
- 28/jul: Sexta: Execução do trabalho
- 01/ago: Terça: Tira dúvidas
- 04/ago: Sexta: Execução do trabalho
- **08/ago: Terça: Entrega trabalho**
- 11/ago: Sexta: Discussão 1a parte, definição 2a parte



Nossa Agenda

- 11/ago, Sexta: Discussão 1aP, definição 2aP
- 15/08, Terça: Tira dúvidas
- 18/08, Sexta: Tira dúvidas
- 22/08, Terça: Execução do trabalho
- 25/08, Sexta: Execução do trabalho
- **29/08, Terça: Entrega do trabalho**
- 01/09, Sexta: Discussão 2aP + revisão 3VA
- 05/09, Terça: 3a va
- 12/09, Terça: Prova final