



Universidade Federal do Ceará
Campus de Sobral
Curso de Engenharia da Computação
Técnicas de Programação – TD02 08/04/2016
Prof. Iális Cavalcante (ialis@sobral.ufc.br)

Prazo de Entrega: 09/04/2016

Questão 01

pacote: td02.aleatorio

Associe as classes *Scanner* e *ArrayList* para a implementação do seguinte programa:

- Leia um vetor de 10 (dez) valores inteiros com a classe *Scanner*;
- Multiplique cada valor lido a um número aleatório (***Math.random()***) que varie de 0 a 10000;
- Insira os 04 (quatro) maiores valores do novo vetor em um objeto da classe *ArrayList*;
- Escreva na saída o valor deste novo objeto.

Questão 02

pacote: td02.interfaceMath

Pesquise sobre o uso da classe **Math** e implemente um programa Java que tenha a entrada de dois números inteiros, **numA** e **numB**, e escreva a saída em valor numérico real:

- Método **floor** aplicado em $numA/numB$;
- Método **ceil** aplicado em $numB/numA$;
- Método **pow** aplicado a $numA^{numB}$ e $numB^{numA}$;
- Método **tan** aplicado aos valores de $numA$, $numB$, $numA/numB$ e $numB/numA$.

Questão 03

pacote: td02.craps

Implemente o jogo de azar: “Craps”. Seu funcionamento é definido da seguinte forma:

- O jogador lança dois dados (cada um com seis faces, com 1, 2, 3, 4, 5 e 6 pontos respectivamente);
- Calcula-se a soma das faces expostas:
 - ✓ Se soma = 7 ou 11 -> jogador venceu;
 - ✓ Se soma = 2, 3 ou 12 -> jogador perdeu;

- ✓ Senão, soma torna-se a “pontuação” do jogador.
- Os dados são lançados novamente:
 - ✓ soma = pontuação -> jogador venceu;
 - ✓ soma = 7 -> jogador perdeu;
 - ✓ Senão, lança os dados novamente.

Questão 04

pacote: td02.testePerformance

Implemente uma classe que apresente um objeto de *ArrayList* e teste o desempenho por tempo da inserção e da consulta de 30000, 50000 e 100000 elementos. A inserção pode ser aplicada com o método *add* e a pesquisa pelo valor dentre os elementos já inseridos, com o método *contains*. Qual método possui melhor desempenho? Descreva a resposta em uma instrução de escrita!

Algumas instruções podem ser utilizadas para contagem do tempo de execução, como exemplo:

```
long inicio = System.currentTimeMillis();  
(...)  
long fim = System.currentTimeMillis();  
long tempo = fim - inicio;
```

A variável *tempo* descreva, em milissegundos, o tempo de execução dos comandos que estiverem entre a inicialização da variável *inicio* e da variável *fim*.

Nota: Esta lista deve ser entregue via SIGAA com todo o projeto criado compactado em arquivo JAR identificando os nomes dos alunos.

Bom Trabalho!
