

Curso: Bacharelado em Ciência da Computação
Disciplina: Linguagens Comerciais de Programação – LCP
Professora: Dra Simone das Graças Domingues Prado

20/março/2020

Aula Prática 03

1. (Instrução auxiliada por computador) O uso de computadores na educação é referido como instrução auxiliada por computador (Computer-Assited Instruction – CAI). Escreva um programa que ajudará um aluno da escola elementar a aprender multiplicação. Utilize um objeto Random para produzir dois inteiros positivos de um algarismo.

O programa deve então fazer ao usuário uma pergunta, como: *"how much is 6 times 7?"* O aluno insere então a resposta. Em seguida, o programa verifica a resposta do aluno. Se estiver correta, exibe a mensagem *"Very good!"* e faz outra pergunta de multiplicação. Se a resposta estiver errada, exibe a mensagem *"No. Please try again"* e deixa que o aluno tente a mesma pergunta repetidamente até o aluno por fim responder corretamente.

Um método separado deve ser utilizado para gerar cada nova pergunta. Esse método deve ser chamado uma vez quando a aplicação inicia a execução e toda vez que o usuário responde a pergunta corretamente. (Deitel&Deitel, 2010, pag 188) **Obs: Use as frases em português.**

2. (Instrução auxiliada por computador: reduzindo a fadiga do aluno) Um problema em ambientes CAI é a fadiga do aluno. Isso pode ser reduzido variando-se as respostas do computador para prender a atenção do aluno. Modifique o programa do exercício anterior para que vários comentários sejam exibidos para cada resposta como mostrado a seguir.

Respostas possíveis para resposta correta:

<i>Very good!</i>	[Muito bom!]
<i>Excellent!</i>	[Excelente!]
<i>Nice work!</i>	[Bom trabalho!]
<i>Congratulation!</i>	[Parabéns!]

Respostas possíveis para uma resposta incorreta:

<i>No. Please try again.</i>	[Não. Tente de novo.]
<i>Wrong. Try once more.</i>	[Errado. Tente mais uma vez.]
<i>Don't give up!</i>	[Não desista!]
<i>No. Keep trying.</i>	[Não. Continue tentando.]

Utilize a geração de números aleatórios para escolher um número de 1 a 4 que será utilizado para selecionar uma de quatro respostas adequadas a cada resposta correta e incorreta. Utilize uma instrução switch para emitir as repostas. (Deitel&Deitel, 2010, pag 188). **Obs: Use as frases em português.**

3. (Instrução auxiliada por computador: níveis de dificuldade) Os exercícios 1 e 2 desenvolveram um programa de instrução auxiliada pro computador para ajudar a ensinar multiplicação a um aluno do ensino básico. Modifique o programa para permitir que o usuário insira um nível de dificuldade. Em nível de dificuldade 1, o programa deve utilizar apenas números de um único dígito nos problemas. Em nível de dificuldade 2, os números com dois dígitos, e assim por diante. Faça até o nível 4. (Deitel&Deitel, 2010, pag 188).

4. Observe a Interface *FuncionarioInterface.java* e as classe *Funcionario.java*, *Professor.java*, *Reitor.java* e *TesteFunc.java* vistas na Aula02 (de 13/março). Crie uma classe de *ProfessorAssistente.java* onde seu valor por hora/aula é de R\$20,00. Adapte a classe *TesteFunc.java* para que o Professor Assistente apareça no relatório gerado.

```
public interface FuncionarioInterface {
    void setNome(String nome);
    String getNome();
    void setSalario(double salario);
    double getSalario();
    double getGastos();
    String getInfo();}
```

5. Observe a classe abstrata *EmployeeAbstract.java* e as classes *SalariedEmployee.java* e *HourlyEmployee.java* vistas na Aula02 (de 13/março). Crie uma classe *ComissionEmployee.java* que representa um funcionário comissionado que possui duas variáveis: *VendasSemanais* (verifique se o valor é maior que zero) e *Comissao* (verifica se a comissão está entre 0 e 1, por que representa porcentagem). Se os valores das variáveis não estiverem corretos, elas receberão o valor zero. O seu salário/ganho é o valor de *VendasSemanais* vezes *Comissao*. Adapte a classe *EmployeeAbstractTest.java* para testar essa classe criada.

```
public abstract class EmployeeAbstract{
    private String firstName;
    private String lastName;
    private String socialSecurityNumber;
    public EmployeeAbstract(String firstName, String lastName, String SSN){
        this.firstName=firstName;
        this.lastName=lastName;
        socialSecurityNumber=SSN;}
    public void setFirstName(String Name){ firstName=Name;}
    public String getFirstName(){ return firstName;}
    public void setLastName(String Name){ lastName=Name;}
    public String getLastName(){ return lastName;}
    public void setSocialSecurityNumber(String Number){
        socialSecurityNumber=Number;}
    public String getSocialSecurityNumber(){
        return socialSecurityNumber;}
    public String toString(){
        return String.format("%s %s\n Social Security Number:%s",
            getFirstName(),getLastName(),getSocialSecurityNumber());}
    public abstract double earnings(); }
```