

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

13/março/2020

Aula Prática 02

1. (Calculadora de frequência cardíaca-alvo) Ao realizar exercícios físicos, você pode utilizar um monitor de frequência cardíaca para verificar se sua frequência permanece dentro de um intervalo seguro, sugerido pelos seus treinadores e médicos. Segundo a American Heart Association (AHA) (www.heart.org), a fórmula para calcular a frequência cardíaca máxima por minuto é 220 menos a idade. Sua frequência cardíaca alvo é o intervalo entre 50-85% da frequência cardíaca máxima.

- Crie uma classe chamada HeartRates. Sua classe deve ter um construtor que recebe esses dados como parâmetros.
- Os atributos da classe devem incluir o nome, sobrenome e data de nascimento da pessoa (consistindo em atributos separados para o mês, dia e ano de nascimento). Para cada atributo forneça métodos set e get.
- Os métodos da classe:
 - um que calcula e retorna a idade da pessoa (em anos),
 - um para calcular a frequência cardíaca máxima da pessoa e
 - um que calcula e retorna a frequência cardíaca-alvo da pessoa.

Escreva um programa Java que solicite as informações da pessoa, instancie um objeto da classe HeartRates; imprime as informações a partir desse objeto (incluindo o nome, sobrenome e data de nascimento da pessoa), calcule e imprima a idade da pessoa (em anos), intervalo de frequência cardíaca máxima e frequência cardíaca-alvo. (Deitel&Deitel, 2010, pag 80).

2. (Computadorização dos registros de saúde) Uma questão relacionada à assistência médica discutida ultimamente nos veículos de comunicação é a computadorização dos registros de saúde. Essa possibilidade está sendo abordada cautelosamente por causa de preocupações quanto a privacidade e segurança de dados sigilosos, entre outros. A computadorização dos registros de saúde pode tornar mais fácil que pacientes compartilhem seus perfis e históricos de saúde entre vários profissionais de saúde. Isso pode aprimorar a qualidade de assistência médica e ajudar a evitar conflitos de medicamentos e prescrições de medicamentos errados, reduzir custos em ambulatórios e poderia salvar vidas.

Nesse exercício, você projetará uma classe HealthProfile para uma pessoa.

- Os atributos da classe devem incluir o nome, sobrenome, sexo, data de nascimento (consistindo em atributos separados para o mês, dia e ano de nascimento), altura e peso da pessoa. Para cada atributo forneça métodos set e get.
- Sua classe deve ter um construtor que recebe esses dados.
- A classe também deve incluir métodos que calcula e retorna a idade da pessoa (em anos), intervalo de frequência cardíaca máxima e frequência cardíaca-alvo e índice de massa corporal (IMC).

Escreva um programa Java que solicite as informações da pessoa, instancie um objeto da classe HealthProfile e imprime as informações a partir desse objeto (incluindo o nome, sobrenome e data de nascimento, altura e peso da pessoa) e, então, calcule e imprima a idade da pessoa (em anos), IMC, intervalo de frequência cardíaca máxima e frequência cardíaca-alvo. (Deitel&Deitel, 2010, pag 80)

OBS: nesse exercício você deve usar o exercício 1 e expandi-lo para aceitar o cálculo do IMC (use herança)

3. Sabe-se de interface que tem um único método chamado “calculaArea()”. Crie classes que implementem essa interface para Círculos, Retângulos, Quadrados e Triângulos. Crie uma classe para testar as outras classes. As classes precisam de variáveis que representem suas medidas e de um construtor.

```
interface AreaCalculavel{  
    double calculaArea();  
}
```

4. (Agora vamos resolver o mesmo problema apresentado na questão 03 usando classe Abstrata) Seja a classe abstrata “AreaCalculavelAbstract” abaixo. Crie as classes Círculos, Retângulos, Quadrados e Triângulos, além de uma para fazer testes.

```
abstract class AreaCalculavelAbstract{  
    private double tamanho1;  
    public AreaCalculavelAbstract(double tamanho1){  
        setTamanho1(tamanho1);  
    }  
    public void setTamanho1(Double t){ tamanho1=t;}  
    public double getTamanho1(){ return tamanho1;}  
    abstract double calculaArea();  
}
```