

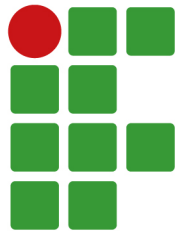
Programação Orientada a Objetos - POOS3¹

Tecnologia em Análise e Desenvolvimento de Sistemas

Exercícios de Fixação

Caderno de exercícios da disciplina.

2º semestre de 2018



INSTITUTO FEDERAL

São Paulo

Câmpus Araraquara



Exercício 1

- Implemente um sistema orientado a objetos que armazene informações de cinco alunos.
- Para cada aluno sabe-se que os dados são prontuário, nome, nota prova 1, nota prova 2, nota dos exercícios e nota do projeto. Cada nota possui um peso idêntico ao da disciplina POOS3.
- O sistema deve informar:
 - Quantidade de alunos aprovados;
 - Quantidade de alunos reprovados;
 - Média geral da turma.
- Utilize na implementação JOptionPane para entrada e saída de dados.

Exercício 2

- Implemente um sistema para uma calculadora de inteiros.
- A calculadora possui, além do construtor, os métodos:
 - `private int soma(int n);`
 - `private int subtrai(int n);`
 - `private int multiplica(int n);`
 - `private int divisao(int n);`
 - `public int calcular(int operando, char operador);`
 - `public void limparMemoria();`
- Implemente vários casos de testes para utilizar a calculadora.

Exercício 3

- Implemente um sistema orientado a objetos em Java que crie uma prova, permita a resposta do aluno e a correção pelo professor.
- A prova pode ter questões dissertativas e múltipla escolha. As questões dissertativas não podem ser corrigidas automaticamente, assim, devem permitir que o professor corrija a questão. Existem dois tipos de questões de múltipla escolha (Verdadeiro e Falso / 5 alternativas). As questões de múltipla escolha possuem correção automática. A quantidade de questões da prova é definida pelo usuário no momento que iniciar o sistema.
- Após criar a prova, permita que um aluno responda a prova. As respostas são armazenadas na própria prova para posterior correção.
- Ao final das respostas do aluno (toda a prova deve ser respondida) inicie o processo de correção. As questões de múltipla escolha são corrigidas automaticamente e as questões dissertativas são apresentadas ao professor. O professor deve ler a resposta correta da questão e a resposta dada pelo aluno e assim decidir se a resposta está correta ou não.
- Ao final da correção apresentar a pontuação do aluno.
- É obrigatório o uso correto dos conceitos: Programação Orientada a Objetos, Herança, Polimorfismo, Classes Abstratas e/ou Interface, boas práticas de programação.
-

Exercício 4

- Devido às constantes estiagens que aconteceram nos últimos tempos em algumas regiões do Brasil, o governo federal criou um órgão para a avaliação do consumo destas regiões com finalidade de verificar o comportamento da população na época de racionamento. Este órgão responsável irá pegar algumas cidades (por amostragem) e verificará como está sendo o consumo de cada uma das pessoas da cidade e o consumo médio de cada cidade por habitante.
- Entrada
 - A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro N ($1 \leq N \leq 1 \cdot 10^6$), indicando a quantidade de imóveis. As N linhas contém um par de valores X ($1 \leq X \leq 10$) e Y ($1 \leq Y \leq 200$), indicando a quantidade de moradores de cada imóvel e o respectivo consumo total de cada imóvel (em m3). Com certeza, nenhuma residência consome mais do que 200 m3 por mês. O final da entrada é representado pelo número zero.
- Saída
 - Para cada entrada, deve-se apresentar a mensagem “Cidade# n:”, onde n é o número da cidade seguindo a sequência (1, 2, 3, ...) e em seguida deve-se listar, por ordem ascendente de consumo, a quantidade de pessoas seguido de um hífen e o consumo destas pessoas, arredondando o valor para baixo. Na terceira linha da saída deve-se mostrar o consumo médio por pessoa da cidade, com 2 casas decimais sem arredondamento, considerando o consumo real total. Imprimir uma linha em branco entre dois casos de teste consecutivos. No fim da saída não deve haver uma linha em branco.

Utilize ArrayList
na solução deste
problema.

Exemplo de Entrada	Exemplo de Saída
3 3 22 2 11 3 39 5 1 25 2 20 3 31 2 40 6 70 0	Cidade# 1: 2-5 3-7 3-13 Consumo medio: 9.00 m3. Cidade# 2: 5-10 6-11 2-20 1-25 Consumo medio: 13.28 m3.

Exercício 5

- Implemente um sistema de cadastro de usuários.
- Cada usuário possui username (único), nome completo e senha (criptografada).
- O sistema deve manter os dados em um List de forma que fiquem classificados pelo username. Não pode-se permitir o cadastro de dois usuários com o mesmo username.
- O sistema deve possuir uma atividade para autenticar o usuário (apenas exibir mensagem se o usuário foi autenticado ou não).
- O sistema deve possuir interface gráfica com o usuários por meio de JFrame.