

Projeto de Linguagem de Programação Orientada a Objetos

Título: Gestão Universitária

Professor Luis Gustavo Araujo

Objetivo: Desenvolver um sistema de gestão para o controle de dados de funcionários, professores e alunos da UNIFACS.

O sistema de gestão deverá ser útil para o cadastro e manutenção de dados de funcionários, professores e alunos da UNIFACS. O sistema deve utilizar, obrigatoriamente, os conceitos de POO como *Classes, Objetos, Métodos, Heranças, Polimorfismo, Classes Abstratas, Atributos e Métodos Estáticos, Associação e Agregação*. O estudante deve analisar o cenário descrito e propor uma solução, antes da codificação, ou seguir o modelo visto em sala (25/05). O sistema deve modelar o cenário, o máximo possível, quanto às classes (isolando as responsabilidades).

A interface Gráfico é opcional, assim como conexão com o banco de dados. O sistema deve, ao menos:

1. Possibilitar a exibição de dados via console (system.out.print). Ex: ao usar BancoDados.getEscolas(), deve aparecer, no console, uma lista de escola cadastradas;
2. Possibilitar o armazenando em memória os dados das Escolas, Curso, Disciplinas, Turmas, Setores e outros objetos (usando array, list ou outra estrutura multivalorada).

Manter administrativo/professores:

O sistema deverá permitir o cadastro/atualização de funcionários administrativos e professores, armazenando/modificando as seguintes informações: *matrícula, nome, endereço, telefone, data de admissão e salário*. Caso seja administrativo: *setor* (escolhido de uma lista de setores cadastrados). Caso seja professor: *escola* (escolhida de uma lista de escolas cadastradas). O sistema deve permitir a exclusão do cadastro através da matrícula.

Manter escolas:

O sistema deverá permitir o cadastro/atualização de escolas, armazenando/modificando as seguintes informações: *id, nome da escola*. Este módulo deverá permitir a exibição de uma lista com todos os professores que lecionam na escola. Ex.: TI, Saúde, RH. O sistema deve permitir, ainda, a exclusão do cadastro através do id.

Manter cursos:

O sistema deverá permitir o cadastro/atualização de cursos, armazenando/modificando as seguintes informações: *código, nome do curso e escola* a qual pertence (escolhida de uma lista de escolas cadastradas). O sistema deve permitir a exclusão do cadastro através do código.

Manter disciplinas:

O sistema deverá permitir o cadastro/atualização de disciplinas, armazenando/modificando as seguintes informações: *código, nome da disciplina, cursos** (escolhido de uma lista de cursos cadastrados).

* A disciplina pode ser de vários cursos (Ex: Algoritmos em Eng. Civil e Algoritmos em SI).

Manter turmas:

O sistema deverá permitir o cadastro/atualização de turmas, armazenando/modificando as seguintes informações: *código*, *nome*, *disciplina* (escolhida de uma lista), *professor* (escolhido de uma lista) e *alunos* (escolhidos de uma lista). O sistema deve permitir a exclusão do cadastro através do código.

Manter alunos:

O sistema deverá permitir o cadastro/atualização de alunos, armazenando/modificando as seguintes informações: *matrícula*, *nome*, *endereço*, *telefone*, *curso* (escolhido de uma lista de cursos cadastradas). O sistema deve permitir a exclusão do cadastro através da matrícula.

Manter setor:

O sistema deverá permitir o cadastro/atualização de aluno, armazenando/modificando as seguintes informações: *id*, *nome*. O sistema deve permitir a exclusão do cadastro através da matrícula.

Avaliação do projeto:

1. O desenvolvimento do grupo de classes do modelo apontado anteriormente valerá, ao todo, 6 pontos (será avaliado o uso correto dos objetos, atributos, métodos ou conceitos de POO necessários). São 8 itens: *Administrativo*, *Professor*, *Escolas*, *Cursos*, *Disciplinas*, *Turmas*, *Alunos* e *Setor*. Algumas classes podem depender de outras aqui não listadas, sendo esse item também avaliado.
2. A utilização correta dos conceitos obrigatórios valerá 3 pontos. São 10 itens: *Classes*, *Objetos*, *Métodos*, *Heranças*, *Polimorfismo*, *Classes Abstratas*, *Atributos* e *Métodos Estáticos*, *Associação* e *Agregação*.
3. O funcionamento correto do sistema valerá 1,0 (será realizado testes de inserção e deleção pelo professor).
4. O uso de interface gráfica será pontuado como *ponto extra*, valendo 1,0 ponto. O ponto extra só será computado até o estudante atingir a nota final de 10 pontos. Ex: *Nota do Trabalho = 9,5*, *Ponto Extra = 1,0*, *Total = 10,0*.
- 5.

Entrega:

1. A entrega dos trabalho deverá ser, obrigatoriamente, via e-mail (luisaraujo.ifba@gmail.com) até 17h do dia 15/06. Ou no dia 15/06, no horário da aula, em sala (através de um pen-drive ou similar);
2. Verifiquem, antes de enviar ou entregar o projeto, a integridade dos arquivos. Projetos com 100% dos arquivos corrompidos não serão avaliados, ficando o estudante como nota 0,0, na avaliação II.
3. Projetos que utilizam pacotes externos deve conter, na pasta do projeto, as classes necessárias ou arquivo .jar. Na impossibilidade de rodar o sistema, o item 3 da avaliação não será realizado.