Curso Superior de Tecnologia em Sistemas para Internet

Terceiro Período

Disciplina: Orientação ao Objetos

Professor: Me. Jonas Pontes

Segunda avaliação da Nota 2

Você e seu time foram desafiados a construir um sistema computacional, seguindo o paradigma da orientação a objetos. O time deve pensar em um problema e a implementação proposta deve solucioná-lo. Faça o levantamento dos requisitos funcionais do sistema, os quais devem ser listados na entrega do sistema. Depois, implemente o protótipo funcional em um projeto Java no GitHub e compartilhe o link como resposta a esta atividade.

Requisitos do projeto:

1. A implementação deve seguir os pilares da orientação a objetos;
2. É necessário que o projeto tenha ao menos quatro classes do tipo modelo e uma de controle. A interação do usuário com o sistema pode ser feita por meio de interface gráfica (GUI) desktop, texto ou web.
3. Use herança, reescrita, polimorfismo. Ao menos uma classe abstrata ou interface deve ser usada.
4. Aos menos seis ações principais devem ser implementadas (itens de menu);

Entregáveis:

1. Projeto Java, via GitHub;
2. Documentação detalhada do problema e do sistema associada, incluindo os seus requisitos. Essa documentação deve ser preferencialmente o arquivo readme.md do GitHub.

Exemplo de um problema:

Você e seu time foram desafiados a desenvolverem um protótipo funcional de um sistema acadêmico. Esse ambiente abrange objetos do tipo aluno, funcionário, curso e turma. A descrição do ambiente e de suas tarefas estão definidas como segue:

1. Modele uma classe chamada Aluno para definir os objetos que representam os alunos da instituição. Essa classe deve declarar três atributos: o primeiro para o nome, o segundo para a matrícula e o terceiro para a data de nascimento dos alunos; todos com visibilidade privada. Modificações e acessos a atributos são feitos via métodos setters e getters, respectivamente. Todos os atributos são obrigatórios no ato da criação de um aluno. CPF não pode ser modificado posteriormente, mas nome e data de nascimento, sim.
2. Implemente, em Java, a classe Aluno como descrita em 1.
3. No ambiente aqui descrito, além dos alunos, temos os funcionários, que também precisam ser representados na aplicação. Modele a classe Funcionario, que deve conter três atributos: o primeiro para o nome, o segundo para o CPF e o terceiro para o salário; todos com visibilidade privada. Modificações, se necessárias, devem ser feitas via métodos setters, e acesso, por métodos getters. Ademais, no ato da instanciação de um novo funcionário, é necessário informar seu nome e seu salário. Com exceção do CPF, todos os atributos são editáveis posteriormente. Os funcionários são especializados como professor, coordenador, diretor e técnico. Um técnico tem os atributos e métodos básicos de funcionário, apenas. Um professor tem um atributo para representar a sua carga horária semanal, e há uma bonificação que se baseia nessa carga horária (a regra ou porcentagem fica a seu critério). Um coordenador é também um professor, e sua bonificação é baseada, além da carga horária de sala de aula, em um acréscimo fixo baseado no cargo. Um diretor tem sua bonificação baseada apenas no cargo.
4. Implemente, em Java, Funcionario e suas subclasses como descrita em 3.
5. Há vários cursos na instituição. Modele Curso, o qual contém atributos privados para representar o nome do curso e a sigla e a modalidade de ensino (graduação ou ensino técnico). Todos os atributos são obrigatórios e modificáveis.
6. Implemente, em Java, o que está descrito em 5.
7. Ainda, os alunos precisam ser divididos por turmas, que devem ser representadas dentro da aplicação. Modele uma classe chamada Turma que contenha dois atributos: o primeiro para o curso, o segundo para definir a série ou semestre. Todos os seus

atributos são privados e ações de modificação e acesso a atributos são feitos via métodos setters e getters, respectivamente. Todos os atributos são obrigatórios no ato da criação. O atributo que representa o curso ao qual a turma pertence não pode ser modificado após a criação.

8. Implemente, em Java, o que está descrito em 7.
9. O sistema permite fazer login, com CPF e senha (outro) para professor, coordenador e diretor. Alunos também fazem login, usando matrícula e senha. Necessariamente, use uma interface para definir um contrato para qualquer classe que deseje que seus objetos sejam autenticáveis. Na interação com o usuário, é preciso ter uma opção para fazer login e o resultado disso pode ser apenas uma simples mensagem indicando se o login foi feito ou não.
10. Implemente, em Java, o que foi descrito em 9.
11. Funcionario não pode ser concreta. Utilize classe abstrata ou interface para isso.
12. O protótipo do programa deve apresentar um menu a permitir as seguintes funcionalidades:
    a. Incluir um novo aluno;
    b. Incluir um novo funcionário;
    c. Incluir uma nova turma;
    d. Emitir relatório de um aluno escolhido pelo usuário;
    e. Emitir relatório de um funcionário escolhido pelo usuário;
    f. Listar todos os funcionários pelo nome;
    g. Mostrar o total de gastos com funcionários;
    h. Listar todas as turmas por sigla e nome;
    i. Mudar um aluno de turma;
    j. Sair do sistema.