

Programação Orientada a Objetos

Plano de Ensino

Rodrigo Bonifácio

14 de agosto de 2014

1 Identificação

- Professor: Rodrigo Bonifácio
- Semestre: 2/2014
- Horário: TER a QUI (08:00 — 09:50)
- Salas: PJC BT 101

2 Objetivo

O objetivo da disciplina é fazer com que os alunos conheçam técnicas para o desenho adequado (reusável, flexível) de um software usando recursos presentes na Orientação a Objetos (polimorfismo, herança, agregação). Para atingir esse objetivo, a disciplina apresenta técnicas e boas práticas de programação comumente usadas em OO, como a adoção de testes unitários, refatoração de código e padrões de projeto. Aulas expositivas serão intercaladas com exercícios práticos e acompanhamento de projetos, nos quais os alunos devem se familiarizar com a linguagem Java (ou alguma outra linguagem OO como C++, Scala ou Smalltalk).

3 Ementa

Esta disciplina objetiva cobrir aspectos conceituais e práticos associados à Orientação a Objetos. O conteúdo abordado inclui conceitos como Classes e Objetos, Herança e Polimorfismo (tanto por subtipos quanto paramétrico), Sobre carga de Operadores, Padrões de Projeto, Refatoração de Código OO, Testes Unitários e Desenvolvimento Dirigido a Testes, Mapeamento Objeto-Relacional.

4 Programa Básico

O programa da disciplina está organizado nos seguintes módulos:

- M1 Conceitos essenciais da programação OO, como classes, objetos, herança, polimorfismo e controle de exceções.
- M2 Aspectos pragmáticos do desenho de software OO, como o uso de notações UML, Desenvolvimento Dirigido a Testes e mapeamento objeto-relacional.
- M3 Qualidade de software OO e boas práticas de desenvolvimento, cobrindo métricas, padrões de projeto e refatoração de código.

5 Avaliação

- P1 Prova individual escrita com o conteúdo do módulo M1. A prova será realizada em uma data ainda a ser definida.
- P2 Prova individual escrita com o conteúdo dos módulos M2 e M3. A prova será realizada em uma data ainda a ser definida.
- P3 Primeira entrega do projeto da disciplina, associada às atividades de implementação do primeiro release do software a ser definido pelos alunos. Essa entrega deverá ser realizada no dia 03/10/2014.
- P4 Segunda entrega do projeto da disciplina, associada às atividades de implementação do segundo release do software a ser definido pelos alunos.. Essa entrega deverá ser realizada no dia 30/11/2014.

$$NF = \left(\frac{P1 + P2}{2} \times 0.6 \right) + \left(\frac{P3 + P4}{2} \times 0.4 \right)$$

Observação. Entregas intermediárias dos projetos podem ser solicitadas, com o intuito de permitir um melhor acompanhamento do projeto. Tais entregas intermediárias influenciam na nota da segunda entrega do projeto. Ambas as entregas estarão sujeitas a apresentação e a avaliação é individual.

6 Regras para o projeto

Os alunos devem escolher um dos projetos sugeridos na disciplina. Tal escolha, junto com a visão geral do sistema que será construído (destacando as funcionalidades, interface com usuário, interface com outros sistemas, mecanismo de persistência, etc.), precisa ser feita até o dia **25/08/2014**. A penalização, por dia de atraso no envio da visão geral do sistema, é de 0.5 pontos na nota P3.

É essencial discutir com antecedência as funcionalidades que serão implementadas nos projetos, com o intuito de evitar a construção de uma solução que não exercite as técnicas de desenvolvimento OO apresentadas no curso.

O documento de visão servirá como contrato em relação ao que será desenvolvido. Aspectos de qualidade do desenvolvimento, como decisões de projeto e automação dos testes, serão considerados na avaliação (individual) do projeto que pode ser realizado em grupo de até três pessoas.

7 Projetos sugeridos

MM UnB: aplicação multimídia usada para o ensino do desenvolvimento na plataforma Android. Muita coisa pode ser feita nessa aplicação, e um projeto da disciplina deve implementar melhorias de usabilidade e algumas features pendentes (como localização).

Linhas de produtos de jogos para Android: desenvolvimento de jogos simples, no estilo *classic arcade game* mas que possuam um claro objetivo abrangendo fases com diferentes níveis de dificuldade. Os jogos, conforme mencionado, podem ser simples no sentido de recursos gráficos e interação, mas precisam ser motivantes e completos.

Cliente de email: a ser usado como estudo de caso para raciocinar sobre interação entre features. A idéia é implementarmos funcionalidades típicas encontradas em aplicações que servem como *clientes* para servidores de e-mail. Tais funcionalidades incluem envio e recebimento de mensagens; gerenciamento de contatos; recursos de criptografia para comunicação mais segura; etc. Podem ser implementados para diferentes plataformas (desktop, Android, IOS, ...)

TG CIC: envolve funcionalidades para que os professores ofertem trabalhos de graduação de seu interesse, bem como funcionalidades para que os alunos notifiquem seu interesse em participar de um determinado projeto. Esse projeto já foi iniciado por um dos monitores da disciplina, mas requer a evolução em termos de funcionalidades e interface com o usuário.

Referências

- [1] Seleção de artigos publicados em bibliotecas digitais (como as da ACM e IEEE). Sempre que pertinente, serão sugeridas leituras de artigos apresentados em conferências de impacto. Links para esses artigos serão disponibilizados no site da disciplina.
- [2] Kent Beck. *Test Driven Development: By Example*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
- [3] Bruce Eckel. *Thinking in C++*. Prentice-Hall, Inc., 1995.
- [4] Martin Fowler. *Refactoring: improving the design of existing code*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

- [5] Erich Gamma, Richard Helm, Ralph E. Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [6] Cay Horstmann and Gary Cornell. *Core Java(TM) Volume 1 / 2*. Prentice Hall, nona edição edition, 2013.
- [7] Scott Meyers. *Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition)*. Addison-Wesley Professional, 2005.
- [8] Stephen Prata. *C++ Primer Plus (Fourth Edition)*. Sams, Indianapolis, IN, USA, 4th edition, 2001.