MC302 – Programação Orientada a Objetos
Primeiro semestre de 2017 – Turmas E e F

Professor responsável:
Fábio Luiz Usberti (fusberti@ic.unicamp.br) – sala 15 (IC1).

Monitores:
Rafael Kendy Arakaki (PED) – rafaelkendyarakaki@gmail.com
Natanael Ramos (PED)
Bleno Humberto Claus (PAD)

1 Página da Disciplina

Página da Disciplina (Sistema Moodle):
http://www.ggte.unicamp.br/ea/

2 Horário das Aulas

Dia	Horário	Sala
Terça	21 – 23	-
Quinta	19 – 21	-
Sábado	10 – 12	CC02 e CC03

3 Ementa

Conceitos básicos e avançados de programação orientada a objetos. Aplicação dos conceitos usando uma linguagem orientada a objetos.

4 Programa

- **Conceitos básicos do modelo de objetos**: objeto, interface pública, estado, mensagens, encapsulamento de dados e operações, comportamento.
- **Modelagem orientada a objetos e UML**
- **Introdução a uma linguagem de programação orientada a objetos**: tipos básicos, estruturas de dados e de programação.
- **Abstração de dados, classes e tipos abstratos de dados**: declaração de classes, métodos, variáveis, modificadores de acesso, instanciação de objetos e construtores.
- **Visibilidade de atributos e métodos**: modificadores de visibilidade privado, público, protegido e de pacote.
- **Modularização e pacotes**
- **Métodos estáticos e atributos estáticos**: atributos de instância versus atributos estáticos; métodos de instância versus métodos estáticos.
- **Vetores estáticos e dinâmicos**: declaração, inicialização e manipulação de arrays unidimensionais e multidimensionais; lista de argumentos de tamanho variável.

- **Relacionamentos entre classes**: hierarquias de generalização/especialização, hierarquias de agregação/decomposição e associação.
- **Enumerações**
- **Classes Internas**: classes internas e anônimas.
- **Herança Simples e Múltipla**: relacionamento "é um"; subclasses e superclasses; sobreposição e ocultamento; métodos e classes finais; construtores em subclasses.
- **Polimorfismo e Acoplamento Dinâmico**: polimorfismo de inclusão, polimorfismo paramétrico, sobrecarga (overloading), redefinição de operações (overriding), acoplamento estático versus dinâmico, métodos finais, classes finais, operações polimórficas, classes genéricas, parametrização de tipos.
- **Classes abstratas**
- **Interfaces**
- **Tratamento de exceções**: erros e exceções; hierarquia de exceções; captura ou declaração de exceções; lançamento de exceções; retrocesso de pilha; encadeamento de exceções; pré-condições e pós-condições; assertivas.
- **Arquivos**: E/S em arquivos textuais e binários; serialização e persistência de objetos.
- **Coleções genéricas**: Interfaces e implementações concretas de estruturas de dados genéricas.
- **Conceitos introdutórios em Padrões de Projeto**

5 Linguagem de Programação

A linguagem de programação utilizada na disciplina sera a Linguagem Java.

6 Critério de Avaliação

A avaliação da disciplina será composta por provas, laboratórios e projetos computacionais. Qualquer tentativa de fraude implicará em **média final zero** no semestre para todos os envolvidos.

6.1 Provas

Serão realizadas duas provas cobrindo todos os tópicos abordados até a data das mesmas. A nota P de provas será calculada como:

$$P = \frac{P_1 + P_2}{2}$$

Onde $P_1 \in [0, 10]$ e $P_2 \in [0, 10]$ são as notas das provas 1 e 2, respectivamente.

6.2 Atividades Práticas

As atividades práticas (laboratórios e trabalhos computacionais) realizados durante a disciplina deverão ser submetidos pelo sistema Moodle na area correspondente à disciplina. Na correção das atividades práticas, serão considerados os seguintes critérios:

- aderência ao enunciado

- algoritmos utilizados e sua implementação
- organização e legibilidade do código
- corretude da solução

Laboratórios: A nota L das atividades de laboratório será calculada da seguinte forma:

$$L = \frac{L_1 + \ldots + L_m}{m}$$

Onde $L_i \in [0, 10]$ é a nota do i-ésimo laboratório e m é o número total de laboratórios aplicados ao longo do semestre.

Trabalhos Computacionais: Durante o semestre serão cobrados dois trabalhos computacionais para aplicação dos conceitos de programação orientada a objetos apresentados ao longo do semestre.

Avaliação: A nota A das atividades práticas será calculada da seguinte forma:

$$A = 0,25L + 0,25T_1 + 0,5T_2$$

Onde $L \in [0, 10]$ é a nota das atividades de laboratório e $T_1, T_2 \in [0, 10]$ são as notas dos trabalhos computacionais 1 e 2, respectivamente.

6.3 Médias Parcial e Final

A média parcial MP do semestre será calculada como:

$$MP = \begin{cases} \dfrac{5P + 5A}{10} & \text{se} \quad \min(P, A) \geqslant 5 \\ \\ \min(P, A) & \text{caso contrário} \end{cases}$$

Alunos com $MP \geqslant 5$ são aprovados. Alunos com $MP < 2.5$ são reprovados sem direito a exame. Alunos com $2.5 \leqslant MP < 5$ e frequência às aulas maior ou igual a 75% podem realizar o exame.

Dado que EX é a nota do exame, a média final MF será calculada como:

$$MF = \begin{cases} \dfrac{MP + EX}{2} & \text{se o aluno fez exame} \\ \\ MP & \text{caso contrário} \end{cases} \tag{1}$$

7 Datas das Avaliações

- **Trabalho Computacional 1**: 18 de abril.
- **Prova 1**: 25 de abril.
- **Trabalho Computacional 2**: 13 de junho.
- **Prova 2**: 22 de junho.
- **Exame**: 11 de julho.

8 Atendimento

As aulas de laboratório poderão ser utilizadas para esclarecimentos de dúvidas com os monitores da disciplina. Após as aulas teóricas, o professor também estará disponível para esclarecimento de dúvidas. Para atendimento extra-classe, entre em contato com o professor pelo ensino aberto para agendar um horário.

9 Bibliografia

1. Java: Como Programar, Paul Deitel & Heivey Deitel; Pearson; 7a. Ed. (no. chamada IMECC – 05.133 D368j)
2. Java in a Nutshell, David Flanagan, O'Reilley, 2005. Disponível on-line pela rede da Unicamp (http://proquest.safaribooksonline.com/?uiCode=unicamp&xmlId=0596007736).
3. The Java Programming Language, Ken Arnold, James Gosling, & David Holmes; Prentice Hall, 4th edition (2005) (no. chamada IMECC – 005.133 Ar64j 3.ed.)
4. Thinking in Java, Bruce Eckel; Prentice Hall, 2th edition (2000) (no. chamada IMECC – 005.133 Ec53t 2.ed.)
5. Data Structures and Algorithms with Object Oriented Design Patterns in Java, Bruno Preiss; (http://www.brpreiss.com/books/opus6/)
6. The Java Tutorials (Oracle) (http://docs.oracle.com/javase/tutorial/)
7. Guia do Usuário UML, Grady Booch et. al.; Campus(1999)
8. Java Pocket Guide - Robert Liguori & Patricia Liguori; O'Reilley, 2008.