

Unidade Curricular:	Programação Orientada a Objetos
Ano letivo:	2014-02
Carga horária:	72 horas
Professor:	Tulio Alberton Ribeiro – tulio.alberton@ifsc.edu.br

1 Objetivos

1.1 Objetivo Geral

- Ter conhecimento sobre programação orientada a objetos e os conceitos sobre UML.

1.2 Objetivos específicos

- Compreender os conceitos da programação orientada a objetos.
- Ter noção sobre a linguagem de programação Java¹.
- Conhecer as estruturas de dados básicas.
- Conhecer o Ambiente Integrado para Desenvolvimento de software (IDE).
- Ser capaz de desenvolver aplicações explorando os conceitos da programação orientada a objetos.

2 Ementa

Introdução ao paradigma da orientação a objetos: Classes, objeto, associações entre classes, herança. Introdução à linguagem de modelagem unificada (UML): Diagramas de caso de uso, classes, sequência. Introdução a linguagem de programação Java: Tipos de dados primitivos, estruturas de controle, vetores; concepção de projeto orientado a objetos, herança, polimorfismo; interfaces gráficas amigáveis.

3 Conteúdo programático

1. Introdução ao paradigma da orientação a objetos
 - (a) Paradigmas de programação: sequencial e estruturada
 - (b) Classes, objetos, encapsulamento, troca de mensagens
2. Introdução a linguagem de programação Java
 - (a) História e princípios da linguagem Java
 - (b) Ambiente de programação Java
 - (c) Desenvolvimento de aplicativos Java
 - i. Tipos de dados primitivos, comandos de entrada e saída
 - ii. Estruturas de controle, de repetição e vetores
 - iii. Classes, objetos, métodos e atributos
 - (d) Desenvolvimento de interfaces gráficas amigáveis através do SWING

¹<http://java.oracle.com>

3. Outros conceitos de orientação a objetos em Java
 - (a) Associação entre classes: agregação e composição
 - (b) Herança e Polimorfismo
 - (c) Classes abstratas e interfaces
4. Uso de APIs Java
 - (a) Tratamento de exceções, Threads, Coleções Java
5. Introdução à linguagem de modelagem unificada (UML)
 - (a) Diagramas de caso de uso, classes e de sequência

4 Metodologia

O conteúdo da disciplina será apresentado através de aulas expositivas e através de aulas práticas no laboratório. Nas aulas práticas serão desenvolvidos exercícios e trabalhos individuais.

5 Avaliação

Os alunos serão avaliados da seguinte forma:

LE – Listas de exercícios sobre todo o conteúdo apresentado em sala e laboratório;

A1 – Avaliação sobre o conteúdo apresentado nos itens 1 e 2 da seção 3 (individual);

A2 – Avaliação sobre o conteúdo apresentado nos itens 2 e 3 da seção 3 (individual);

P1 – Projeto prático sobre o conteúdo apresentado (em dupla) ².

- Para a aprovação é necessário obter no mínimo o conceito “C” nas **avaliações A1 e A2** e no **projeto P1**. É necessário possuir no mínimo 75% de presença para ser considerado aprovado.
- Haverá avaliação de recuperação individual para aqueles que não obtiverem no mínimo o conceito “C” na avaliação A1 ou A2. O conteúdo da prova de recuperação será o mesmo abordado na prova em que o aluno não alcançou o conceito desejado.

6 Cronograma

Aula	Data	Aulas	Atividade
1	30/07/2014	2	Apresentação e Revisão dos Paradigmas de programação: sequencial e estruturada
2	31/07/2014	2	História e princípios da linguagem Java
3	06/08/2014	2	Introdução ao ambiente de programação Java
4	07/08/2014	2	Estrutura de controle, repetição e vetores
5	13/08/2014	2	Introdução ao paradigma da orientação a objetos
6	14/08/2014	2	Introdução ao Java: Classes, objetos, métodos e atributos
7	20/08/2014	2	Introdução ao Java: Classe String
8	21/08/2014	2	Apresentação do ambiente de desenvolvimento integrado NetBeans
9	27/08/2014	2	Lista de exercícios 1
10	28/08/2014	2	Sobrecarga de métodos e palavras reservadas: final , this , static
11	03/09/2014	2	Associação entre classes
12	04/09/2014	2	Associação entre classes

²O projeto P1 ficará a critério do professor cobrá-lo ou não

Aula	Data	Aulas	Atividade
13	10/09/2014	2	Lista de exercícios 2
14	11/09/2014	2	Avaliação 1
15	17/09/2014	2	Revisão sobre a avaliação 1
16	18/09/2014	2	Herança, classes abstratas e interfaces
17	24/09/2014	2	Herança, classes abstratas e interfaces
18	25/09/2014	2	Polimorfismo
19	01/10/2014	2	Lista de exercícios 3
20	02/10/2014	2	Uso de APIs Java: Tratamento de exceções
21	08/10/2014	2	Uso de APIs Java: Tratamento de exceções
22	09/10/2014	2	Uso de APIs Java: Sockets
23	15/10/2014	2	Uso de APIs Java: Sockets
24	16/10/2014	2	Uso de APIs Java: Threads
25	22/10/2014	2	Uso de APIs Java: Threads
26	23/10/2014	2	Lista de exercícios 4
27	29/10/2014	2	Processo de Desenvolvimento de software - Casos de uso, regras de negócio, ciclo de vida de um projeto
28	30/10/2014	2	Processo de Desenvolvimento de software - Casos de uso, regras de negócio, ciclo de vida de um projeto
29	05/11/2014	2	Introdução a UML: Diagrama de classes
30	06/11/2014	2	Introdução a UML: Caso de uso e diagrama de caso de uso
31	12/11/2014	2	Introdução a UML: Diagrama de sequência
32	13/11/2014	2	Lista de exercícios 5
33	19/11/2014	2	Avaliação 2
34	20/11/2014	2	Revisão sobre a avaliação 2
35	26/11/2014	2	Aula dedicada ao desenvolvimento do projeto final
36	27/11/2014	2	Aula dedicada ao desenvolvimento do projeto final
37	03/12/2014	2	Aula dedicada ao desenvolvimento do projeto final
38	04/12/2014	2	Aula dedicada ao desenvolvimento do projeto final
39	10/12/2014	2	Apresentação do projeto final
40	11/12/2014	2	Avaliação de recuperação

Bibliografia

- [1] Eduardo Bezerra. *Princípios de análise e projeto de sistemas com UML*. Campus, 2002.
- [2] Caelum. Fj-11 – java e orientação a objetos. Technical report, Caelum Ensino e Soluções em Java, 2008. <http://www.sj.ifsc.edu.br/~mello/livros/java/apostila-caelum-java-objetos-fj11.pdf>.
- [3] H.M. Deitel and P.J. Deitel. *Java Como Programar*. Prentice Hall, 4 edition, 2003.
- [4] Cay S. Horstmann and Gary Cornell. *Core Java – Volume I – Fundamentos*. Pearson, 8 edition, 2010.
- [5] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Overgaard. *Object-oriented software engineering: a use case driven approach*. Addison-Wesley, 1992.