

Programa do Minicurso

1 Identificação

Disciplina: Minicurso Java

Carga Horária: 18 horas-aula Teóricas: 8 Práticas: 12

Período: Março a Maio de 2017.

2 Cursos (público alvo)

Ciências da Computação;

Sistemas de Informação.

3 Requisitos

Nenhum.

4 Ementa

Aritmética em programação; Estruturas Condicionais e de Seleção; Estruturas de Repetição; Métodos, Procedimentos e Funções; Encapsulamento de variáveis; Interfaces; Tipos definidos por usuário; Classes abstratas; Padrões de Projeto; Diagramas UML (*Unified Modeling Language*); Classes e Casos de Uso; Componentes básicos AWT (*Abstract Window Toolkit*); Modelo Observador-Observável; Modelo MVC; *Callback*; Reuso; Boas Práticas de Programação (padrões de nomenclatura/casing, etc.); Funções *lambda*; *Cache friendliness*; *Generics*; Classe Anônima; Mecanismo de alocação de memória da JVM; Introdução à ferramenta de desenvolvimento Eclipse; Depuração; Análise de Algoritmos e Complexidade Assintótica; *HashMap*; Serialização;

5 Objetivos

Geral: Auxiliar na aprendizagem das disciplinas de Programação Orientada a Objetos 1 e 2, produzindo um projeto em console Java, seguido de sua

adaptação para interface gráfica, mostrando normas e passos para um código bem feito e reutilizável.

Específico: São objetivos do minicurso:

- Apresentar as especificidades básicas da linguagem Java, dentre elas:
 - Classe principal e método `main`;
 - Formas de declarar/instanciar variáveis e vetores;
 - Ciclo de vida das variáveis;
 - Referência e cópia;
 - Funcionamento do método construtor.
- Criar um projeto simples para mostrar o uso das estruturas básicas (comparação, seleção, repetição, funções...) da programação imperativa;
- Melhorar o desempenho e entendimento dos alunos nas disciplinas de Programação Orientada a Objetos 1 e 2;
- Instruir os participantes a respeito das boas práticas de programação, a fim de que façam códigos bem estruturados, reutilizáveis e legíveis, aplicando conceitos de código-limpo e que façam escolhas adequadas de padrões de projetos;
- Introduzir noções de algoritmos e resolução de problemas computacionalmente;
- Criar um projeto com enfoque no interfaceamento gráfico e uso de diferentes recursos de programação orientada a objetos (interfaces, classes, classes abstratas...);
- Expor formas de melhorar a manutenibilidade de software.

6 Conteúdo Programático

1. Introdução a Java (2 horas):
 - (a) Características principais da linguagem;
 - (b) Processo de compilação e *Bytecode*;
 - (c) Processo de execução da JVM (*Java Virtual Machine*);
 - (d) Introdução ao paradigma Imperativo;
 - (e) Tipos primitivos;
 - (f) Estruturas condicionais.
2. Coleções (2 horas):
 - (a) Laços de repetição;
 - (b) *Raw-arrays*;

- (c) Tempo de vida de variáveis: Escopo;
 - (d) Matrizes;
 - (e) *Cache-friendliness*.
3. Tipos definidos por usuário (2 horas):
- (a) Classes e objetos;
 - (b) Atributos;
 - (c) Alocação de memória na JVM;
 - (d) Garbage-Collection;
4. Introdução a Algoritmos (2 horas):
- (a) Busca linear e *Binary-Search*;
 - (b) Complexidade de Algoritmos;
 - (c) Notação assintótica;
 - (d) *Binary-Sort*.

7 Cronograma

Referências

- [1] Oracle. Java Platform SE 8 Documentation.
<https://docs.oracle.com/javase/8/docs/api/index.html>.