Apresentação da Disciplina & Introdução a OO

Instituto Metrópole Digital

Disciplina: IMD0040 - Linguagem de Programação II

Docente: Emerson Alencar

Professor

Emerson Alencar - emerson@imd.ufrn.br

Atendimento: Terça e Quinta, 16:00 às 16:50 - Sala A213

Orientações

- Uso do Celular em Sala
- Assiduidade
- Frequência
- Monitoria
- Atividades
- Outras Disciplinas

A Disciplina:

Carga horária: 60h – 36 encontros

Quantidade de avaliações: 3

Objetivo:

- Conhecimento sólido dos princípios de programação orientada a objetos;
- Capacidade de avaliar a qualidade de um (pequeno) sistema de software;
- Capacidade de implementar um pequeno sistema de software em Java utilizando boas práticas de programação e padrões de projetos.

Metodologia:

- Aulas teóricas expositivas;
- Aulas práticas em Laboratório;
- Atividades extra sala.

Avaliações

- Unidade I:
  - Atividades (20%)
  - Avaliação (80%)
- Unidade II:
  - Trabalho (20%)
  - Avaliação (80%)
- Unidade III
  - Projeto (80% ou 100%)
  - Avaliação ??

Plágio não será tolerado!!!!

Cronograma de Aulas (Unidade I)

Aula 01	26/07/2016	Aula01 - Apresentação da disciplina + Introdução a OO (objetos e classes)	Introdução a orientação a objetos, Conceitos: Objetos, Classes, Métodos, Parâmetros, Instância, Introdução a ferramenta BlueJ
Aula 02	28/07/2016	Aula02 - BlueJ + Definindo classes + Interação entre objetos + Exercícios	Campos, Construtores, Métodos, Parâmetros, Abstração, modularização, Chamadas de método, Diagramas de classe/objeto
Aula 03	02/08/2016	Aula03 - Interação entre objetos + Exercícios	
Aula 04	04/08/2016	Aula04 - Agrupando objetos	Coleções, Utilizando classes de bibliotecas, Lendo a documentação e outros tipos de coleções
Aula 05	09/08/2016	Aula05 - Agrupando objetos (prática)	
Aula 06	11/08/2016	Aula06 - Comportamento mais sofisticado	Testes, depuração e Automação de Testes
Aula 07	16/08/2016	Aula07 - Objetos bem-comportados (teste de software) (prática)	
Aula 08	18/08/2016	Aula08 - Design de Classes	Design baseado na responsabilidade, Acoplamento, coesão e refatoração
Aula 09	23/08/2016	Aula09 - Exercícios e Revisão	
Aula 10	25/08/2016	Aula10 - Avaliação I	

Cronograma de Aulas (Unidade II)

Aula 11	30/08/2016	Aula11 - Eclipse + Aperfeiçoando estruturas com o uso de herança	Herança, subtipagem, variáveis polimórficas
Aula 12	01/09/2016	Aula12 - Discussão Prova 01 + Vista	
Aula 13	06/09/2016	Aula13 - Herança	Polimorfismo de método, sobrescrição, tipo estático e dinâmico
Aula 14	08/09/2016	Aula14 - Mais sobre herança (Polimorfismo)	
Aula 15	13/09/2016	Aula15 - Prática: Herança e Polimorfismo	
Aula 16	15/09/2016	Aula16 - Interfaces e classes abstratas	
Aula 17	20/09/2016	Aula17 - Interfaces e classes abstratas (prática)	
Aula 18	22/09/2016	Aula18 - Tratando erros	
Aula 19	27/09/2016	Aula19 - Tratando erros (pratica)	
Aula 20	29/09/2016	Aula20 - Design de Classes	
Aula 21	04/10/2016	Aula21 - Organização de Código Java	
Aula 22	06/10/2016	Aula22 - Exercícios e Revisão	
Aula 23	11/10/2016	Aula23 - Implementação Árvore Binária de Busca	
Aula 24	13/10/2016	Aula24 - Design de Classes (prática)	
Aula 25	18/10/2016	Aula25 - Avaliação II	

Cronograma de Aulas

Aula 26	20/10/2016	Aula26 - Padrões de projeto + Introdução tema projeto
Aula 27	25/10/2016	Aula27 - Padrões de projeto + Discussão prova + Vista
Aula 28	27/10/2016	Aula28 - padrões de projeto + Acompanhamento de Projeto
Aula 29	01/11/2016	Aula29 - Acompanhamento Projeto
Aula 30	03/11/2016	Aula30 - Acompanhamento Projeto
Aula 31	08/11/2016	Aula31 - Acompanhamento Projeto
Aula 32	10/11/2016	Aula32 - Acompanhamento Projeto
Aula 33	15/11/2016	Aula33 - Acompanhamento Projeto
Aula 34	17/11/2016	Aula34 - Acompanhamento Projeto
Aula 35	22/11/2016	Aula35 - Acompanhamento Projeto
Aula 36	24/11/2016	Aula36 - Avaliação III - Apresentação do Projeto

Lições aprendidas em outras turmas

- Os alunos deixam de se dedicar à disciplina durante a segunda unidade, porque obtiveram uma boa nota na primeira unidade
- Porque outras disciplinas começam a apertar
- Não seja passivo
- Aproveite as aulas práticas para praticar
- Faça os exercícios/trabalhos/projetos
- É melhor fazer exercício/trabalho/projeto do que prova
- Planeje seu semestre
- Estude regularmente
- Não deixe para estudar no último dia
- “Programação se aprende fazendo”, pratique

Frequência

O art. 47, § 3º, da Lei das Diretrizes e Bases da Educação Nacional (LDB) nº 9.394, de 20 de dezembro de 1996, dispõe que é obrigatória a freqüência de alunos e professores, salvo nos programas de educação a distância, que se regem por outras disposições. **Não existe legalmente abono de faltas**. É admitida, para a aprovação, a freqüência mínima de 75% da freqüência total às aulas e demais atividades escolares, em conformidade com o disposto na Resolução nº 4, de 16/9/86, do extinto Conselho Federal de Educação.

Ainda sobre frequência

- Carga horária: 60 horas (36 encontros)
- Cada dia de aula conta como 2 aulas
- Atenção ao horário das aulas
- Não existe previsão legal para “tolerância de X minutos”
- Se perdeu a chamada ou não assinou a lista, recebe falta
- A presença será atualizada semanalmente no SIGAA
- De acordo com o Art. 106 do regulamento da graduação, o aluno reprovado por falta não tem direito a avaliação de reposição
- Planeje suas faltas para ficar dentro dos 25% permitidos

Ferramentas

- Java SE JDK (Java Decelopment Kit) 8
- BlueJ (www.bluej.org)
- Qualquer editor de texto
- Eclipse (Unidade II)
- Scener Builder (Unidade II ou III)

Perguntas????

Introdução a Orientação a Objetos

Roteiro

- Introdução a orientação a objetos
- Conceitos
- Objetos
- Classes
- Métodos
- Parâmetros
- Instância
- BlueJ
- Introdução a ferramenta

Paradigmas de programação

- Um paradigma de programação fornece e determina a visão que o programador possui sobre a estruturação e execução do programa.
- Exemplo:
  - Programação estruturada(procedural)
  - Programação orientada a objetos

Paradigmas de programação

- Programação estruturada (procedural)
- Descreve a computação como ações, enunciados ou comandos que mudam o estado (variáveis) de um programa
- IF/FOR/WHILE
- Baseado no conceito de chamadas a procedimentos

Paradigmas de programação

- Programação orientada a objetos
- A orientação a objetos é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamada objetos.

Orientação a objetos

- Surge nos anos 60~70 com a linguagem Smalltalk
- Se espalhou para diversas linguagens: C++, Java, C#, etc.
- “Uma nova maneira de pensar os problemas utilizando conceitos do Mundo Real. O componente fundamental é o objeto que combina estrutura e comportamento em uma única entidade”

Raumbaugh

Objetos

Programação orientada a objetos

- O que é um objeto?
- **Abstração** para representação computacional de entidades
- Entidades que representam ‘coisas’ reais, ou de algum domínio de um problema (abstrato)
- Exemplo:
  - Carro (concreto)
  - Cão (concreto)
  - Transação Bancária (abstrato)

Programação orientada a objetos

- **… estrutura e comportamento ….**
- Estado (a estrutura)
  - Atributos do objeto (suas características)
- Operações (o comportamento)
  - Métodos
- **As operações mudam** os valores dos atributos de um objeto (**seu estado**)

Programação orientada a objetos

- Métodos e atributos

Atributos
Raça: Poodle
Nome: Rex
Peso: 5 quilos

Métodos
Latir
Comer
Dormir

Atributos
Potência: 500cc
Marca: Honda
Ano: 1998
Velocidade: 100km/h

Métodos
Acelerar
Frear
Abastecer

Programação orientada a objetos

- Como sabemos quais os atributos de um objetos?
- Como sabemos quais os métodos de um objetos?

Classes

Classe

- Abstração que descreve um objeto
  - Quais atributos o objeto tem
  - Quais mensagens ele entende
  - Como ele se comporta
- Uma forma para se criar objetos
  - Objetos são representações concretas de uma classe
  - Também chamados de instâncias de uma classe
- Conjunto de objetos
  - Características/comportamentos comum
  - Representam todos os objetos de um tipo

Programação orientada a objetos - Classe

Gato
Raça: Savannah
Nome: Gatuno
Peso: 2,5 quilos
Idade: 2 anos

Gato
Raça: Maine Moon
Nome: Listrado
Peso: 3 quilos
Idade: 5 anos

Gato
Raça: Siamês
Nome: Bichano
Peso: 4 quilos
Idade: 3 anos

Programação orientada a objetos

- Observações
  - Muitas instâncias podem ser criadas a partir de uma única classe
  - Um objeto tem atributos: valores armazenados em campos
  - A classe define quais campos um objeto tem
  - Mas cada objeto armazena seu próprio conjunto de valores (o estado do objeto)

Programação orientada a objetos

- Métodos são parecidos com funções
  - Recebem parâmetros
  - Possuem um valor de retorno

Atributos
Potência: 500cc
Marca: Honda
Ano: 1998
Velocidade: 100km/h

Métodos
Acelerar
Frear
Abastecer
Ver velocidade

Quantos litros colocar quando abastecendo?
Qual a velocidade atual?

Programação orientada a objetos

- Métodos são parecidos com funções
  - Recebem parâmetros
  - Possuem um valor de retorno

Atributos
Potência: 500cc
Marca: Honda
Ano: 1998
Velocidade: 100km/h

Métodos
Acelerar
Frear
Abastecer
Ver velocidade

Quantos litros colocar quando abastecendo? Parâmetro
Qual a velocidade atual? Retorno

A linguagem de programação Java

A linguagem de Programação Java

- Início em 1991, com um pequeno grupo de projeto da Sun MicroSystems
- Desenvolvimento de software para uma ampla variedade de dispositivos de rede e sistemas embutidos
- Decisão pela criação de uma nova linguagem de programação que fosse simples, portátil e fácil de ser programada
- Linguagem interpretada Oak (carvalho em inglês), que foi renomeada para Java devido a problemas de direitos autorais
- Oficialmente apresentada no SunWorld’95
- Java Developer’s Kit (JDK) 1.0

A linguagem de Programação Java

- Dezembro de 2011

A linguagem de Programação Java

Características:

- Simples
- Fácil aprendizado
- Sintaxe – similar a C++
- Paradigma – similar a Smalltalk
- Orientada a objetos
- Foco da disciplina
- Distribuída
- Bibliotecas para programação distribuída

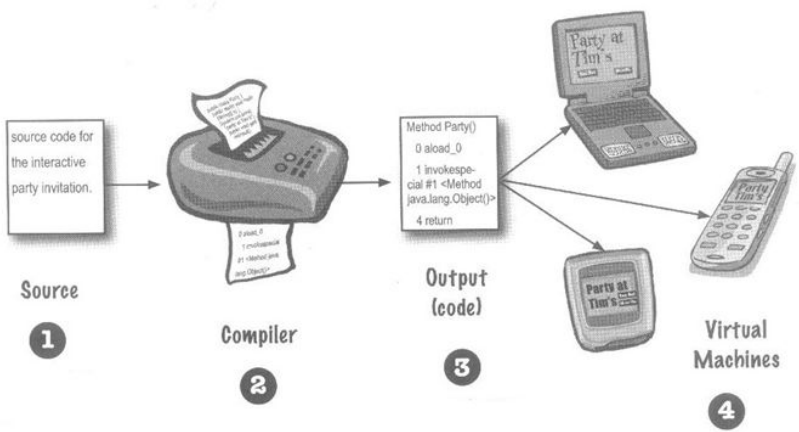
A linguagem de Programação Java

- Características
- Robusta e Segura
- Fortemente tipada
- “Sem ponteiros”
- Coleta de lixo (garbage collector)
- Portável
- Código transformado em bytecodes
- Executam em qualquer maquina com Java Virtual Machine (JVM)
- Independente de plataforma

A linguagem de Programação Java

Java é Portável

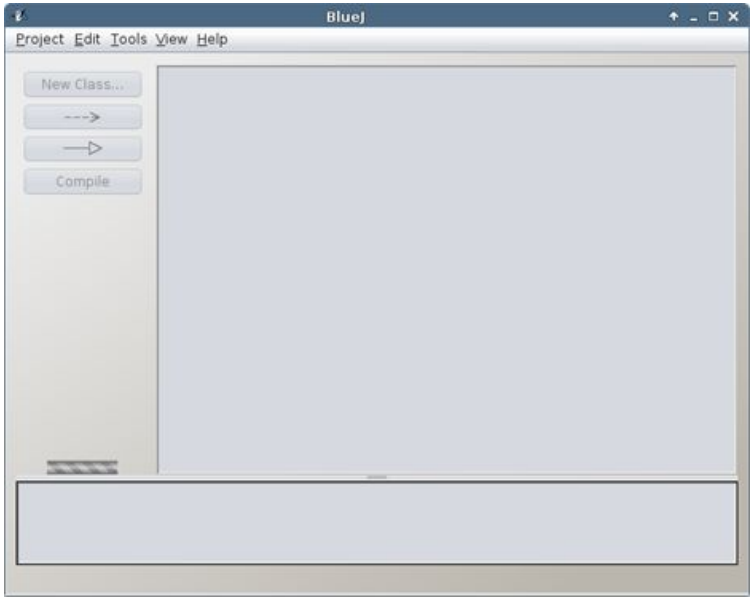
- Conversão para código intermediário (bytecodes)
- Bytecode pode ser executado em qualquer plataforma
- Basta que haja uma JVM para ela



BlueJ

BlueJ

- Ambiente interativo para Java
- Projetado para ensino de POO
- www.bluej.org
- Página oficial com diversos recursos
- Projetos de exemplo, documentação, fórum, ….



Livro: BlueJ

David J. Barnes & Michael Kölling

Programação orientada a objetos com Java:
Uma introdução prática usando BlueJ

4a. Edição,
Pearson Prentice Hall, 2009
ISBN 978-85-7605-187-9.

BlueJ

- Exercícios do Cap. 1 do livro
  - Para se acostumar com o ambiente
- Fazer todos os exercícios
  - Responder no papel quando for o caso

Por hoje é só