

## **Fundamentos de Programação**

Atividade Avaliativa II – 4º Bimestre

Prof. Bruno E. G. Gomes

Data: 03/04/2013

### **Objetivos**

Proporcionar a pesquisa e desenvolvimento em programação orientada a objetos.

### **Grupos**

- De até 3 alunos;
- O grupo deve escolher 1 dentre os temas propostos.

### **Orientação**

- Os trabalhos serão acompanhados e orientados pelo professor nos horários de CA e em horários extras combinados com a turma.

### **Critérios de avaliação**

- Serão avaliados a apresentação, relatório técnico e códigos desenvolvidos;
- Apresentação, com nota individual, terá peso de 50%;
- Relatório técnico e códigos serão parte da nota do grupo, com peso de 50%;
- Nota do aluno = nota individual + nota do grupo;
- O trabalho terá peso de 50% na nota do bimestre, sendo os outros 40% a nota da primeira avaliação escrita e 10% da postura do aluno em sala (exercícios, participação, comportamento, etc.)

### **Apresentação:**

- Critérios avaliados para a nota da apresentação seguem em arquivo em anexo;
- Recursos disponíveis: quadro negro, projetor, software de apresentação *BrOffice 3.4* e *MS Power Pointer 2010*;
- 15 a 20 minutos por grupo.

### **Relatório:**

- Mínimo de 7 páginas – Introdução, seções de desenvolvimento e considerações finais;
- Capa, contracapa e demais elementos pré-textuais, sumário e referências não contam para o mínimo de 7 páginas;
- Deve seguir o modelo enviado em anexo.

### **Datas**

**05/abr** – Definição dos grupos e temas. Faça uma descrição resumida do programa ou dos assuntos que serão abordados no tema de trabalho teórico.

**10/abr** – Esboço inicial do relatório, com uma primeira versão da descrição do sistema/tecnologia estudada. Incluir diagrama de classes, no caso de um programa.

**17/abr** – Relatório com a descrição detalhada do sistema ou, no caso de trabalho teórico, com uma versão detalhada de todas as seções explicando a tecnologia abordada. Sorteio do dia e ordem das apresentações.

**24/abr** – Apresentações dos trabalhos e entrega dos relatórios finais.

**25/abr** – Apresentações dos trabalhos e entrega dos relatórios finais.

**26/abr** – Avaliação final (recuperação).

## **Temas**

1. Implementação em C++ de um pequeno sistema ou Jogo
2. POO em outras linguagens (*Java*, *PHP*, *C#*, *Python* ou *Ruby*) – apenas uma linguagem por grupo
3. Padrões de Projeto Orientado a Objetos GRASP
4. Desenvolvimento de interface gráfica com o usuário
  - a) C++ - Ferramentas *Qt creator* (linux) ou *Microsoft Visual C++* (Windows)
  - b) Object Pascal – Ferramenta *Delphi*

## **Detalhamento dos Temas**

**Observação:** Em todos os temas abaixo, além do relatório e da apresentação, o grupo deve entregar ao professor todos códigos desenvolvidos.

### **Tema 1 – *Implementação em C++ de um pequeno sistema ou Jogo***

1. O programa desenvolvido deve exercitar todos os conceitos de orientação a objetos vistos em sala. Deve-se também procurar utilizar o máximo de recursos da linguagem C++ que abordamos nos dois últimos bimestres.
2. Os dados utilizados pelo programa devem ser lidos e exibidos na saída padrão (monitor) em interface de modo texto. Não é obrigado o uso de interface gráfica, não implicando necessariamente em uma nota maior para o grupo que o fizer.
3. O relatório técnico deve conter, obrigatoriamente, a descrição geral do sistema desenvolvido, uma especificação básica do sistema, em texto, descrevendo as suas funcionalidades e o diagrama de classes da aplicação.

### **Tema 2 – *POO em outras linguagens (Java, PHP, C#, Python ou Ruby)***

1. Deve ser feita pesquisa e descrição no relatório técnico de como a linguagem escolhida, dentre as opções acima, implementa a programação orientada a objetos.
2. O relatório deve conter, obrigatoriamente, uma apresentação geral da linguagem e seus recursos básicos. Devem constar seções descrevendo os tópicos de POO e como eles são implementados naquela linguagem (encapsulamento, membros de classe e de objeto, herança, polimorfismo e demais conceitos particulares à linguagem).
3. O relatório e a apresentação também deve possuir exemplos de código na linguagem.

### **Tema 3 – *Padrões de atribuição de responsabilidade (GRASP)***

1. Conjunto de padrões para melhor distribuição de responsabilidade de objetos em um sistema. Neste tema o grupo deve descrever cada padrão GRASP, acompanhado de um diagrama de classes para cada padrão;
2. Deve-se utilizar como referência básica o capítulo sobre o assunto do livro *Aplicando UML e Padrões*, do *Craig Larman* (tem na biblioteca). Outras bibliografias na internet ou em livros também devem ser consultadas.
3. O grupo deve fazer pequenas implementações em uma linguagem OO para exemplificar cada padrão.
4. O relatório deve conter a explicação detalhada de cada padrão de responsabilidade, com diagramas e exemplos.

### **Tema 4 – *Desenvolvimento de Interface gráfica com o usuário***

1. Neste tema, o objetivo é apresentar os recursos para desenvolvimento de

interface gráfica para aplicações *desktop* utilizando um *software* apropriado.

2. As ferramentas permitidas são *Qt creator* ou *Qt designer* (no linux, para C++), *Delphi*, ou MS *Visual C++*.
3. Devem ser desenvolvidas pequenas exemplos (ou uma interface maior) para exercitar os recursos de interface, tais como botões, caixas de textos, caixas de seleção, etc. O tratamento de eventos/ações ao se clicar em um elementos de interface também deve ser detalhado.
4. Um pequeno programa demonstrando a conexão da interface com as classes de algum sistema também pode ser utilizado para exemplificar os recursos da interface.
5. O relatório deve conter, obrigatoriamente, a apresentação da ferramenta e dos recursos e exemplos que o grupo utilizou.