



Trabalho Final

Leia **atentamente** este documento. Nele estão descritos todos os requisitos mínimos da aplicação a ser implementada, entregas e os critérios de avaliação.

Objetivo

O Trabalho Final objetiva exercitar as habilidades e conceitos de programação desenvolvidos ao longo do semestre através da implementação de uma aplicação orientada a objetos utilizando a linguagem de programação Java.

Orientações

Os alunos deverão formar grupos de dois ou três integrantes, no máximo, e definir que aplicação desejam implementar como trabalho final. Isto é, **a escolha do tema/problema a ser abordado através da aplicação a ser implementada é livre para cada grupo**. Você pode pensar em implementar um jogo, ou um sistema que esteja alinhado com alguma necessidade que você observa dentro da universidade, da sua área de atuação ou de sua experiência profissional. Seja criativo! Além da definição do problema, os alunos serão responsáveis pela interpretação, modelagem do problema e implementação da aplicação, de forma que estejam de acordo com os princípios de orientação a objetos vistos na disciplina.

Requisitos

- A implementação da aplicação deve ser realizada em Java. O grupo pode utilizar a IDE de sua preferência, desde que, ao final, sejam fornecidos corretamente os arquivos solicitados (vide item **"Entrega"**). A implementação da aplicação **precisa** adotar técnicas de orientação a objetos aprendidas em aula de forma correta; i.e., não basta que a aplicação funcione corretamente.
- A implementação deverá aplicar, **obrigatoriamente**, os conceitos de encapsulamento, classes, objetos, métodos, atributos, mensagens, herança, polimorfismo, classes abstratas e interfaces. **Todos** estes conceitos devem estar representados no modelo e código entregues.
- A aplicação deve envolver, ao menos, 8 (**oito**) entidades distintas, entre classes, classes abstratas e interfaces.
- Devem ser utilizados corretamente os modificadores de acesso, priorizando o uso de **private** a fim de garantir o encapsulamento. Ou seja, o uso de outros modificadores de acesso devem ser justificados por necessidade imposta pela aplicação, não

podendo ser escolhidos por critérios definidos pelo grupo como, por exemplo, a "praticidade na implementação".

- A aplicação deve incluir uma interface de comunicação com o usuário (gráfica ou não) que o oriente quanto ao uso da aplicação e permita realizar as operações disponíveis e visualizar informações, de acordo com o problema abordado. Atenção: **não é necessário sofisticação em sua interface**, o que será valorizado é a disponibilidade das operações e a visualização das informações.
- Ao implementar a aplicação, o aluno deve se preocupar com a **consistência e coerência dos dados e das operações** realizadas. Por exemplo, se a aplicação visar o gerenciamento de uma loja *online*, então só se deve efetuar a venda de itens com estoque disponível.
- A aplicação deve estar devidamente documentada. Isto inclui tanto o uso apropriado de comentários no código, documentação das classes e diagrama de classes (Perspectiva de Especificação) que represente a estrutura final do sistema (classes/interfaces e relacionamentos).

Desejáveis

- Criar e disponibilizar um arquivo executável **.jar** gerado a partir dos códigos (o **.jar** deve ser entregue juntamente com os códigos originais).
- Utilizar o conceito de pacotes.
- Criar uma interface gráfica para auxiliar o uso da aplicação. Atenção: o uso de uma interface (gráfica ou não) para guiar o usuário é **obrigatória**.
- Ao executarmos a aplicação, é interessante que a mesma já inicie com um estado interno pré-definido, ou seja, com uma série de dados internos representando informações já cadastradas e operações previamente realizadas de acordo com o contexto da aplicação, tal que seja possível o **teste direto** das suas funcionalidades. O grupo pode implementar isto através do uso de arquivos, realizando inicializações manualmente, ou da forma que julgar mais apropriado.

Entrega

O trabalho será entregue em **3 (três) etapas**:

1. Sinalização e entrega da descrição do problema a ser modelado: até dia **15 de Abril de 2021** às 23:55 horas pelo ambiente virtual Moodle.
 - O grupo deve **sinalizar** seus integrantes e uma breve descrição do problema (1-2 **frases**) que irá modelar e da aplicação que visa implementar. Tal sinalização será feita em fórum específico e de forma **pública** aos demais alunos da disciplina. Dois ou mais grupos **não** podem focar em um mesmo problema/solução. O controle de se o problema/solução já foi ou não selecionado(a) por outro grupo deve ser feito pelos próprios **grupos** (checar fórum indicado antes de sinalizar suas escolhas). **O grupo que sinalizar primeiro, publicamente e em fórum específico, o problema/solução que irá modelar tem preferência!**

- O grupo deve entregar um breve relatório (1-2 **parágrafos**) **em formato PDF**, com a descrição do problema que irá modelar e da aplicação que visa implementar. Nesta descrição já deverão constar as **entidades abstraídas** (classes e interfaces) a partir do contexto (entidades que modelam interação como usuário podem ser adicionadas em momento posterior). Se já houverem definições a respeito de funcionalidades e relacionamentos entre classes, estes também **podem** constar no documento.
 - O relatório será analisado pelo Professor e, se necessário, discutido com o grupo a fim de deixá-lo mais alinhado com as especificações dos requisitos mínimos. Detalhes sobre classes e funcionalidades podem ser mudados durante a implementação, mas os alunos deverão manter até o final do semestre a proposta de projeto (tema/aplicação definido(a)).
2. Entrega do código: até dia **20 de Maio de 2021** às 23:55 horas pelo ambiente virtual Moodle.
- *Upload* no ambiente Moodle usando um único **arquivo compactado (.zip)**. Este arquivo deve conter o **código completo e documentado** (arquivos .java são obrigatórios), bem como o **diagrama de classes final** em Perspectiva de Especificação (formato imagem ou **PDF**). O arquivo **.jar**, quando gerado, deve ser incluído neste arquivo compactado.
 - Inclua o nome dos autores no cabeçalho do arquivo que contém o método principal, em forma de comentário.
 - Você pode fazer *upload* de diferentes versões e depois aperfeiçoar a aplicação. Identifique adequadamente as versões, tal que a mais recente seja facilmente identificada. Faça o *upload* assim que tiver uma versão funcional, de modo a garantir a entrega e precaver-se de problemas com servidor, rede, etc.
3. Apresentação em vídeo: até dia **25 de Maio de 2021** às 23:55 horas pelo ambiente virtual Moodle.
- Cada grupo deve apresentar (a) o problema, (b) a modelagem e (c) as principais funcionalidades da aplicação desenvolvida. Tais funcionalidades podem ser exibidas na execução da aplicação. Idealmente, todos os alunos do grupo devem explicar, ao menos, um ponto ((a), (b) ou (c)) no vídeo. Minha sugestão é que cada um se encarregue de explicar um desses itens ((a), (b) ou (c)) e grave seu próprio vídeo. **A entrega, por grupo, deve conter um vídeo apenas!**
 - A apresentação, por grupo, deve ter no máximo **7 minutos** de duração. A elaboração de uma "apresentação" objetiva e clara, que se adeque a este tempo, faz parte do processo de avaliação.
 - O vídeo pode ser enviado via ambiente virtual Moodle (*upload*) ou hospedado no Youtube (recomenda-se manter o vídeo "*unlisted*"). Nesse último caso, a entrega nada mais é que um arquivo texto contendo o *link* para o vídeo. Após a data de entrega do código, nenhuma alteração referente ao desenvolvimento da aplicação será permitida.
 - A apresentação do trabalho conta nota na avaliação final.

A avaliação deste Trabalho Final possui um peso de **30%** no cálculo da média final do aluno, representando parte da nota concedida no quesito “Trabalho Final” (o restante se refere às notas das Atividades Teóricas (ATs) e Práticas (APs) desenvolvidas ao longo do semestre). Lembre-se que a não entrega desta atividade implica em Recuperação. **A aplicação modelada/desenvolvida deve atender a todos os requisitos listados neste enunciado, deve obrigatoriamente estar orientada a objetos, não deve apresentar erros de compilação e deve executar normalmente.** A aplicação desenvolvida deverá demonstrar os seguintes conteúdos que serão avaliados:

1. Habilidade em estruturar aplicações pela decomposição da realidade modelada em classes. Cada classe deve ser responsável pelo seu estado interno (atributos) e propor uma série de funcionalidades para alterar o estado interno (métodos). (3 pontos)
2. Documentação de programas (identação, utilização de nomes de variáveis, abstração dos procedimentos para obter maior clareza, uso de comentários no código). (1,5 ponto)
3. Domínio na utilização dos conceitos estudados: encapsulamento, herança, polimorfismo, classes abstratas, interface, sobrecarga (*overload*) e sobreescrita (*override*) de métodos, atributos de instância ou de classe, métodos de instância ou de classe, modificadores de acesso. (3 pontos)
4. Formatação e controle de entrada e saída, com construção de interfaces que orientem corretamente o usuário, sem instruções ou intervenção adicional do programador. Garantia da consistência e coerência dos dados e operações. (1,5 ponto)
5. Clareza, organização e demonstração de domínio da solução implementada avaliados na apresentação final do projeto (vídeo). (1 ponto)

Observações

Este trabalho deve refletir a solução individual do grupo para o problema proposto. Casos de plágio (mais de 90% de similaridade) serão tratados com severidade e resultarão em anulação da nota e, conseqüentemente, na reprovação dos alunos envolvidos (vide programa da disciplina). Para detectar e quantificar o plágio no código, será utilizado o [software MOSS](#).

Dicas

- Para auxiliar na elaboração da interface gráfica, caso o grupo deseje implementar uma, adote uma IDE. Explore a documentação das bibliotecas gráficas [Swing](#) (recomendada) ou [AWT](#). O NetBeans, por exemplo, tem um [tutorial](#) ilustrando como se pode projetar interfaces gráficas com *Swing*. [Tutoriais Oracle](#) sobre *Swing* também são boas referências públicas.
- A apresentação “*Material Extra: exportando um arquivo .jar*”, disponibilizada no ambiente virtual Moodle, explica como exportar o arquivo .jar via linha de comandos.
- Tire bom proveito do *Google* e da infinidade de informações que ele fornece!