

# Programação com Objectos/Projecto de Programação com Objectos/Avaliação do Projecto (Época Normal)

From Wiki\*\*3

< Programação com Objectos | Projecto de Programação com Objectos

## AVISOS - Avaliação em Época Normal

[Collapse]

Esclarecimento de dúvidas:

- Consultar sempre o corpo docente atempadamente: presencialmente ou através do endereço oficial da disciplina [1].
- Não utilizar fontes de informação não oficialmente associadas ao corpo docente (podem colocar em causa a aprovação à disciplina).
- Não são aceites justificações para violações destes conselhos: quaisquer consequências nefastas são da responsabilidade do aluno.

Requisitos para desenvolvimento, material de apoio e actualizações do enunciado (ver informação completa em Projecto de Programação com Objectos):

- O material de apoio é de uso obrigatório e não pode ser alterado.
- Verificar atempadamente (mínimo de 48 horas antes do final de cada prazo) os requisitos exigidos pelo processo de desenvolvimento.

Processo de avaliação (ver informação completa em **Avaliação do Projecto**):

- Datas: **2023/09/29 12:00** (inicial); **2023/10/13 12:00** (intercalar); **2023/10/27 12:00** (final); **2023/10/27 (early bird) 2023/10/30 (normal)** (teste prático).
- **Todas as entregas são cruciais para o bom desenvolvimento do projecto, sendo obrigatórias: a não realização de uma entrega implica a exclusão da avaliação do projecto e, por consequência, da avaliação da disciplina.**
- Verificar atempadamente (até 48 horas antes do final de cada prazo) os requisitos exigidos pelo processo de avaliação, incluindo a capacidade de acesso ao repositório.
- **Apenas se consideram para avaliação os projectos existentes no repositório oficial. Apenas se considera para avaliação o ramo 'main'.**
- Trabalhos não presentes no repositório no final do prazo têm classificação 0 (zero) (não são aceites outras formas de entrega). Não são admitidas justificações para atrasos em sincronizações do repositório. A indisponibilidade temporária do repositório ou de outros materiais, desde que inferior a 24 horas, não justifica atrasos na submissão de um trabalho.
- A avaliação do projecto pressupõe o compromisso de honra de que o trabalho correspondente foi realizado pelos alunos correspondentes ao grupo de avaliação.
- **Fraudes na execução do projecto terão como resultado a exclusão dos alunos implicados do processo de avaliação.**

## Material de Uso Obrigatório

[Collapse]

As bibliotecas **po-uilib** e o conteúdo inicial do repositório GIT são de **uso obrigatório**:

- po-uilib (classes de base) po-uilib-202308010000.tar.bz2 (não pode ser alterada) - javadoc
- xxl-core (classes do "core") (via GIT) (deve ser completada -- os nomes das classes fornecidas não podem ser alterados)
- xxl-app (classes de interacção) (via GIT) (deve ser completada -- os nomes das classes fornecidas não podem ser alterados)

A máquina virtual, fornecida para desenvolvimento do projecto, já contém todo o material de apoio.

## Uso Obrigatório: Repositório GIT

**Apenas se consideram para avaliação os projectos existentes no repositório GIT oficial.**

Trabalhos não presentes no repositório no final do prazo têm classificação 0 (zero) (não são aceites outras formas de entrega). Não são admitidas justificações para atrasos em sincronizações do repositório. A indisponibilidade temporária do repositório, desde que inferior a 24 horas, não justifica atrasos na submissão de um trabalho.

## Contents

- 1 Método de Avaliação do Projecto
- 2 Cálculo da Nota do Projecto e Condições de Aprovação
- 3 Entrega do Modelo UML
- 4 Entrega Intermédia
  - 4.1 Funcionalidade a implementar
  - 4.2 Critérios de avaliação (não automática)
- 5 Entrega Final
- 6 Teste Prático
- 7 Considerações Adicionais sobre a Avaliação do Projecto
- 8 Compromisso de Honra

## Método de Avaliação do Projecto

A avaliação relativa à componente do Projecto processa-se em várias fases:

- Entrega do modelo UML (**obrigatória**: ver condições abaixo)
- Entrega intermédia (**obrigatória**: ver condições abaixo)
- Entrega final (**obrigatória**: ver condições abaixo)
- Teste prático (**obrigatório**: ver condições abaixo)

O Projecto (trabalho conducente às entregas acima mencionadas e abaixo descritas) é realizado em grupos de 2 elementos ou individualmente durante o período estabelecido.

**TODAS AS ENTREGAS SÃO REALIZADAS ATÉ ÀS 12:00 ("meio-dia") DAS RESPECTIVAS DATAS**

O Teste Prático é realizado individualmente, em data e local a agendar.

Considerando que todas as entregas são cruciais na concepção e realização do projecto, a não realização de qualquer entrega conduz automaticamente a uma classificação de 0 (zero) na componente de avaliação relativa ao Projecto e consequente exclusão da avaliação da disciplina no ano lectivo em curso.

## Cálculo da Nota do Projecto e Condições de Aprovação

Componentes de avaliação:

- **PRJ** - nota final do projecto
- **PU** - nota da entrega do diagrama UML
- **PI** - nota da entrega intermédia
- **PF** - nota da entrega final
- **TP** - nota do teste prático

A nota do teste prático condiciona a distância ao mínimo entre as notas do teste prático e a do projecto: desde um mínimo de 12.5% de acréscimo à menor das duas (abaixo de 7.5 valores no TP), até um máximo de 25% (para 20 valores no TP). O acréscimo é linear entre 7.5 e 20.

Ou seja:

- $PRJ = \min(PU+PI+PF, TP) + |TP - PU - PI - PF| * \max(12.5, TP+5) / 100$

Condições necessárias para aprovação à disciplina -- **necessárias todas**:

- **PRJ >= 9.5** (sem arredondamento)
- **PRJ != NA**
- **TP > 0**
- **TP != NA**

## Entrega do Modelo UML

A Entrega do Modelo UML avalia o estado do projecto relativamente à concepção do modelo.

Esta entrega tem uma classificação máxima de 3 valores.

**A não realização da Entrega do Modelo UML conduz automaticamente a uma classificação de 0 (zero) na componente do projecto.**

Devem ser entregues diagramas de classes correspondentes a todos os conceitos da aplicação, mesmo que esses conceitos venham a ser revistos/alterados futuramente. Podem ainda ser incluídas anotações nos diagramas, por forma a clarificar outros aspectos relevantes. Não é necessária a inclusão de getters/setters/equals/toString (e outros métodos já definidos no Java).

- Diagrama 1: diagrama do pacote **po-uilib** (material de apoio que não pode ser alterado): **UML-po-uilib.pdf** (não é necessário representar o conteúdo de **pt/tecnico/uilib/swing**)
- Diagrama 2: diagrama do pacote **xxl-app** (material do projecto completo -- pode receber adições de código em métodos **execute**, de atributos nas classes dos comandos e de classes auxiliares): **UML-xxl-app.pdf**
- Diagrama 3: diagrama do pacote **xxl-core** (material do projecto incompleto e que deve ser completado e desenvolvido -- ou seja, deve incluir adicionalmente partes que correspondem à interpretação do enunciado): **UML-xxl-core.pdf**. Este diagrama é de entrega **obrigatória** (grupos que não o entreguem estão excluídos da avaliação).

A entrega dos diagramas **obrigatoriamente manuscritos** é feita em documentos PDF na directoria "uml" do projecto no GIT. O formato é A4 (cada diagrama pode ter várias páginas) e realiza-se até à data limite: 6ª feira, 2023/09/29, 12:00 (meio-dia). Documentos ilegíveis ou noutros formatos serão sumariamente ignorados, mesmo que isso resulte em reprovação dos alunos envolvidos: é da responsabilidade dos alunos produzir material de boa qualidade nas condições de avaliação.

Apenas são aceites diagramas manuscritos e posteriormente digitalizados (usando uma aplicação como nas aulas laboratoriais): serão recusados quaisquer diagramas produzidos com ferramentas informáticas, mesmo que isso resulte em reprovação dos alunos envolvidos: é da responsabilidade dos alunos produzir material de material de boa qualidade nas condições de avaliação.

Cada diagrama deve conter a seguinte declaração de autoria, assinada sob compromisso de honra (os diagramas entregues por grupos de 2 alunos devem conter duas declarações independentes com o mesmo texto e correspondente assinatura): **Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo de projecto. (assinatura)**

**Diagramas entregues sem declaração de autoria serão considerados indício de fraude e os alunos correspondentes excluídos da avaliação da disciplina, sendo comunicada a situação aos órgãos da Escola.**

Serão aplicados os seguintes critérios na avaliação da entrega UML.

Factores aditivos **positivos**:

- 0.80 - Conceitos
- 0.50 - Herança
- 0.70 - Associações
- 0.50 - Atributos
- 0.50 - Interface (métodos das classes, interfaces, etc.)

Factores aditivos **negativos**:

- 0.50 - Erros de notação

A compreensão clara dos aspectos acima deve ser adquirida antes da entrega.

## Entrega Intermédia

A Entrega Intermédia avalia o estado do projecto relativamente a um mínimo de funcionalidade.

Serão executados testes automáticos nesta entrega.

Esta entrega tem uma classificação máxima de 6 valores (2.5 valores para testes automáticos).

**A não realização da Entrega Intermédia conduz automaticamente a uma classificação de 0 (zero) na componente do projecto.**

## Funcionalidade a implementar

A funcionalidade necessária para a entrega intermédia, além da abertura de todos os submenus, é a seguinte:

- Leitura, interpretação e representação em memória do ficheiro textual indicado pela propriedade **import** (implica implementar algumas classes do "core" -- na entrega intermédia, o ficheiro de entrada textual não contém funções que aceitam intervalos)
- Menu principal: Salvaguarda do estado actual da aplicação: "Criar", "Abrir" e "Guardar" -- implementação completa
- Menu principal: Gestão e consulta de dados da aplicação: abertura dos vários submenus (comandos fornecidos já implementados).
- Menu de Edição -- implementar a operação "Visualizar".

Todos os outros comandos em todos os menus têm de estar implementados ("execute" pode estar vazio). Não é má prática implementar os outros comandos, pois poupa esforço para a entrega final, mas não serão avaliados nesta entrega.

A funcionalidade a implementar em **xxl-core** tem de ser suficiente para suportar os comandos indicados acima.

## Critérios de avaliação (não automática)

Serão aplicados os seguintes **critérios na avaliação da entrega intermédia** do projecto.

Factores aditivos **positivos**:

- 0.40 - Atributos não públicos
- 0.30 - Atributos e métodos não "static" (excepto onde autorizado)
- 0.20 - Serialização
- 0.75 - Utilização correcta de herança
- 0.25 - Utilização das estruturas de dados correctas (incluindo, sem limitação, o conceito de utilizador)
- 0.50 - Flexibilidade da estrutura de representação do armazenamento
- 0.60 - Separação app/core (pode haver descontos na parte automática)
- 0.30 - Qualidade do projecto
- 0.20 - Comentários Javadoc (a colocar necessariamente na classe **Calculator**)

Factores aditivos **negativos**:

- 0.15 - Violação de regras de codificação Java

Será aplicado um desconto até 1.00 pela existência de lixo no repositório GIT. Este "lixo" é apenas considerado se aparecer na cópia local durante um "checkout".

## Entrega Final

A Entrega Final pressupõe que todo o projecto foi completamente implementado.

Esta entrega tem uma classificação máxima de 11 valores:

- 5.50 valores para a qualidade do código
- 5.50 valores para testes automáticos

**A não realização da Entrega Final conduz automaticamente a uma classificação de 0 (zero) na componente do projecto.**

Os testes correspondem a uma série de entradas a processar pelo projecto de cada grupo e cuja execução deve corresponder a um conjunto de resultados padrão.

Serão aplicados os seguintes critérios na avaliação da entrega final do projecto.

Factores aditivos **positivos**:

- 0.50 - Utilização de estruturas de dados correctas
- 1.00 - Aplicação dos princípios de desenho aberto/fechado e programar para o supertipo
- 0.50 - Qualidade da solução utilizada para suportar procuras
- 1.25 - Qualidade da solução utilizada para suportar propagação de alterações
- 1.00 - Fugas de privacidade entre as entidades do core e a camada de interacção com o utilizador e separação de responsabilidades, incluindo serialização (core vs. app)
- 0.75 - Verificação de situações erróneas nos programas (e.g., não mascarar excepções, entre outros aspectos)
- 0.50 - Apreciação global

Factores aditivos **negativos**:

- até 1.00 - Violação de regras de codificação Java
- até 1.00 - Uso directo dos tipos Object (excepto onde inevitável), Exception (excepto como superclasse), Throwable, Error
- até 1.00 - Existência de lixo no repositório GIT (o "lixo" é apenas considerado se aparecer na cópia local durante um "checkout")
- até 0.50 - Atributos (qualidade e acesso)
- até 0.25 - Atributos e métodos "static" (excepto autorizados)

**Note-se que não é considerada a entrega final se o repositório à data da entrega final contiver o mesmo código da entrega intermédia. Isso implica também a reprovação imediata dos alunos do grupo em causa (todas as entregas do projecto são obrigatórias).**

## Teste Prático

O Teste Prático consiste num conjunto de questões (correspondentes a pequenas alterações/extensões ao enunciado) a resolver com base na implementação submetida para a avaliação correspondente à entrega final.

Este teste avalia o conhecimento do aluno relativamente ao projecto entregue, assim como a sua capacidade de realizar alterações ao código do projecto.

O teste prático é como uma discussão de projecto, pelo que não existem repescagens como nos testes escritos. Alunos que faltem ao teste prático estão automaticamente reprovados à disciplina.

**A não realização do Teste Prático conduz automaticamente a uma classificação de 0 (zero) na componente de avaliação relativa ao Projecto e consequente exclusão da avaliação da disciplina no ano lectivo em curso.**

**O teste prático contém uma prova eliminatória: o aluno deverá ser capaz de compilar e executar um pequeno programa em Java.**

## Considerações Adicionais sobre a Avaliação do Projecto

**O projecto deverá ser desenvolvido atempadamente ao longo do período lectivo.**

As versões intermédias registadas no GIT poderão ser testadas, pelo que deverão ser periodicamente actualizadas. O projecto é constituído por um projecto GIT designado pelo número do grupo que o executa. O repositório GIT disponibilizado já contém uma versão inicial do projecto. Apenas os ficheiros registados no projecto GIT serão considerados na avaliação.

**Não serão consideradas quaisquer alterações a alguns dos ficheiros disponibilizados: eventuais cópias desses ficheiros serão automaticamente substituídas durante a avaliação da funcionalidade do código submetido. Ver página sobre o repositório, para mais pormenores.**

A avaliação executa testes automáticos ao projecto a implementar: a nota reflectirá apenas os testes com sucesso. Chama-se especial atenção para o nome da classe que inicia a execução. As restantes componentes da nota são obtidas pela análise do código e resultado do teste prático (realizado individualmente). O código é avaliado quanto à sua correcção, simplicidade, extensibilidade e legibilidade: devem existir comentários das partes mais complexas, mas não devem ser excessivos, nem óbvios (diminuem a legibilidade), nem muito escassos (impedem a compreensão).

**Qualquer alteração à especificação providenciada no enunciado é penalizada, mesmo que possa ser entendida como um melhoramento.**

Notar que o facto de os testes automáticos terem sido superados não reflecte a qualidade do código, quer do ponto de vista de engenharia de software, quer do ponto de vista da correcta aplicação dos princípios leccionados na disciplina. A funcionalidade do projecto final é, no entanto, de extrema importância, pelo que **é preferível um projecto que realize, correctamente, apenas parte da funcionalidade a um quase completo mas que nem sequer compile ou que não processe nenhuma entrada correctamente.**

## Compromisso de Honra

A entrega de quaisquer trabalhos pressupõe o compromisso de honra que foram realizados pelos autores referenciados e que resultam do esforço por cumprir os requisitos da entrega. A quebra deste compromisso, nomeadamente, a tentativa de apropriação de trabalhos alheios, terá como consequência a reprovação de todos os alunos envolvidos (incluindo os que possibilitaram a ocorrência) no ano lectivo actual, sem prejuízo de acções disciplinares previstas pelo IST.

Categories: **Projecto de PO** **PO** **Ensino**