

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa Nº1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.
- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.
- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.
- \_\_\_\_\_ O Acceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ O MOF não permite relações de herança.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.

2. Nas linhas de produtos de software. . .

- (a) usa-se aspect-oriented programming para modelar a variailidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

3. Admitindo uma abordagem de linhas de produtos de software. . .

- (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

4. Num modelo de domínio. . .

- (a) os *value objects* podem conter referências para *entities*.
- (b) os *value objects* não podem fazer parte de *aggregates*.
- (c) os *value objects* não podem ser parâmetros de *services*.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Meta-metamodelo	(a) OCL
_____ Linguagem de transformação <i>model-to-text</i>	(b) Xtext
_____ UML Structure Diagram	(c) ATL
_____ Ferramenta para DSL textuais	(d) Feature model
_____ UML Behavior Diagram	(e) Class diagram
_____ Linhas de produtos de software	(f) State diagram
_____ Linguagem para especificar constrangimentos	(g) Acceleo
_____ Linguagem de transformação <i>model-to-model</i>	(h) ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. MDE.
2. SPL - Software Product Line.

### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.
2. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
3. A associação \_\_\_\_\_ representa uma coleção.
4. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].
2. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].
3. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o \_\_\_\_\_ não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.
2. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.
3. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.
4. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

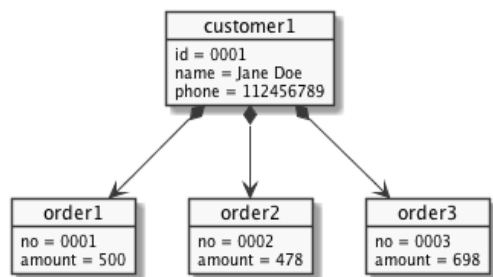
```

let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de um modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam A

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |       |  |
|-------|--|
| False | Um modelo de domínio é uma parte fundamental da camada UI.   |
| False | O OCL é uma exemplo de general-purpose language.   |
| False | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| False | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| True  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| True  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |
| False | A sintaxe abstrata apenas lida com a notação textual.  |
| True  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| True  | Num modelo de domínio os repositórios permitem persistir entidades.  |
| False | Em OCL o termo <b>this</b> é usado para se aceder à instância de contexto.   |
| True  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |
| False | O Acceleo é uma ferramenta de transformação entre modelos.   |
| False | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| True  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| True  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| False | Existe uma relação de um-para-um entre modelos e diagramas.  |
| True  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| True  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| False | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| False | O MOF não permite relações de herança.   |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.
2. Nas linhas de produtos de software...
  - (a) usa-se aspect-oriented programming para modelar a variabilidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.

3. Admitindo uma abordagem de linhas de produtos de software...

- (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

4. Num modelo de domínio...

- (a) os *value objects* podem conter referências para *entities*.
- (b) os *value objects* não podem fazer parte de *aggregates*.
- (c) os *value objects* não podem ser parâmetros de *services*.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

(h) Meta-metamodelo	(a) OCL
(g) Linguagem de transformação <i>model-to-text</i>	(b) Xtext
(e) UML Structure Diagram	(c) ATL
(b) Ferramenta para DSL textuais	(d) Feature model
(f) UML Behavior Diagram	(e) Class diagram
(d) Linhas de produtos de software	(f) State diagram
(a) Linguagem para especificar constrangimentos	(g) Acceleo
(c) Linguagem de transformação <i>model-to-model</i>	(h) ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.
2. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.

## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.
2. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
3. A associação investments representa uma coleção.
4. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].
2. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].
3. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: self.end > self.start;.
2. Indique o que falta na linha 19 (nos "..."): subtopics.
3. Indique o que falta na linha 18 (nos "..."): parent.
4. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

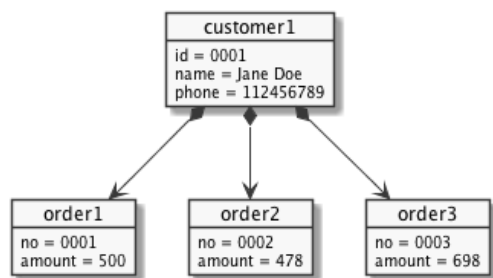
```

let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____>asSet())->asOrderedSet();

```

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.



## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ O Acceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papeis no processo de desenvolvimento de software.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ O MOF não permite relações de herança.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.

2. Admitindo uma abordagem de linhas de produtos de software. . .
  - (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
  - (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
  - (c) não se distingue engenharia de domínio e engenharia de aplicação.
  - (d) nenhuma das anteriores.
3. Nas linhas de produtos de software. . .
  - (a) usa-se aspect-oriented programming para modelar a variailidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.
4. Num modelo de domínio. . .
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Linguagem de transformação <i>model-to-model</i>	(a) Feature model
_____ UML Behavior Diagram	(b) Class diagram
_____ Meta-metamodelo	(c) ecore
_____ Linguagem de transformação <i>model-to-text</i>	(d) ATL
_____ UML Structure Diagram	(e) Xtext
_____ Linhas de produtos de software	(f) Acceleo
_____ Linguagem para especificar constrangimentos	(g) State diagram
_____ Ferramenta para DSL textuais	(h) OCL

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.**
2. **MDE.**

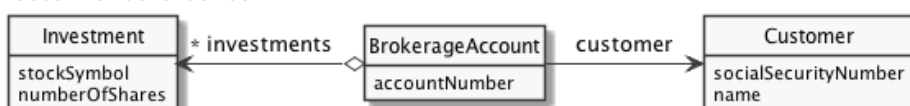
### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
2. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.
3. A associação \_\_\_\_\_ representa uma coleção.
4. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
2. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o \_\_\_\_\_ não aceita duplicados.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].
4. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.oclIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

```

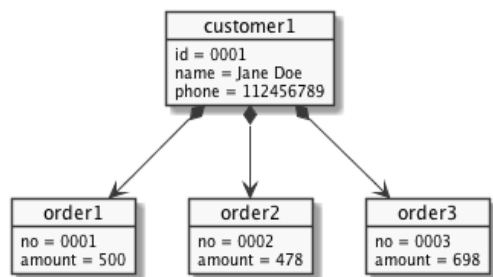
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

3. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.
4. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam B

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |       |  |
|-------|--|
| False | Existe uma relação de um-para-um entre modelos e diagramas.  |
| True  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |
| True  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |
| False | Em OCL o termo <code>this</code> é usado para se aceder à instância de contexto.   |
| False | O Acceleo é uma ferramenta de transformação entre modelos.   |
| False | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| False | A sintaxe abstrata apenas lida com a notação textual.  |
| True  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| True  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| False | O MOF não permite relações de herança.   |
| True  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| False | Um modelo de domínio é uma parte fundamental da camada UI.   |
| True  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| False | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| False | O OCL é uma exemplo de general-purpose language.   |
| True  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| False | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| False | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| True  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| True  | Num modelo de domínio os repositórios permitem persistir entidades.  |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.
2. Admitindo uma abordagem de linhas de produtos de software...
  - (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
  - (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
  - (c) não se distingue engenharia de domínio e engenharia de aplicação.
  - (d) nenhuma das anteriores.

3. Nas linhas de produtos de software. . .

- (a) usa-se aspect-oriented programming para modelar a variailidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- ☒ (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

4. Num modelo de domínio. . .

- ☒ (a) os *value objects* podem conter referências para *entities*.
- (b) os *value objects* não podem fazer parte de *aggregates*.
- (c) os *value objects* não podem ser parâmetros de *services*.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

<u>(d)</u>	Linguagem de transformação <i>model-to-model</i>	(a)	Feature model
<u>(g)</u>	UML Behavior Diagram	(b)	Class diagram
<u>(c)</u>	Meta-metamodelo	(c)	ecore
<u>(f)</u>	Linguagem de transformação <i>model-to-text</i>	(d)	ATL
<u>(b)</u>	UML Structure Diagram	(e)	Xtext
<u>(a)</u>	Linhas de produtos de software	(f)	Acceleo
<u>(h)</u>	Linguagem para especificar constrangimentos	(g)	State diagram
<u>(e)</u>	Ferramenta para DSL textuais	(h)	OCL

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.
2. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.

## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitassemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
2. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.
3. A associação investments representa uma coleção.
4. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
2. O Set e o Bag são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].
4. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: self.end > self.start;.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade `rootTopics` na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

```

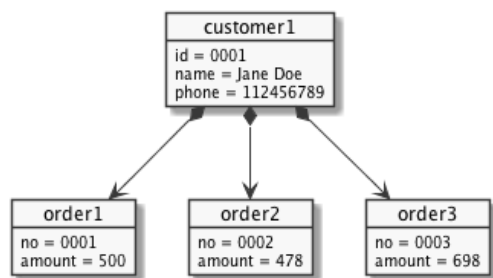
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

3. Indique o que falta na linha 18 (nos "..."): parent.
4. Indique o que falta na linha 19 (nos "..."): subtopics.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.



## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.
- \_\_\_\_\_ O MOF não permite relações de herança.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.
- \_\_\_\_\_ O Aceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papeis no processo de desenvolvimento de software.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Admitindo uma abordagem de linhas de produtos de software...
  - (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
  - (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
  - (c) não se distingue engenharia de domínio e engenharia de aplicação.
  - (d) nenhuma das anteriores.

2. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.
3. Nas linhas de produtos de software...
  - (a) usa-se aspect-oriented programming para modelar a variailidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.
4. Num modelo de domínio...
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Linguagem para especificar constrangimentos	(a) ecore
_____ Ferramenta para DSL textuais	(b) State diagram
_____ Linguagem de transformação <i>model-to-text</i>	(c) OCL
_____ Linguagem de transformação <i>model-to-model</i>	(d) Class diagram
_____ Linhas de produtos de software	(e) Feature model
_____ UML Structure Diagram	(f) Xtext
_____ Meta-metamodelo	(g) Acceleo
_____ UML Behavior Diagram	(h) ATL

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.**
2. **MDE.**

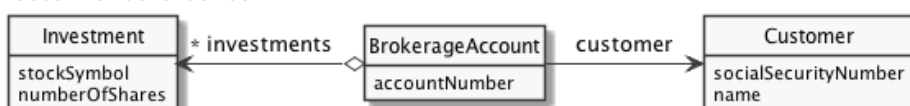
### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.
2. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.
3. Se necessitassemos de modelar o endereço postal do *Customer* apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
4. A associação \_\_\_\_\_ representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. O *Set* e o *Bag* são dois tipos de coleções do OCL. Destas duas o \_\_\_\_ não aceita duplicados.
2. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
3. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].
4. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.oclIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade *rootTopics* na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

```

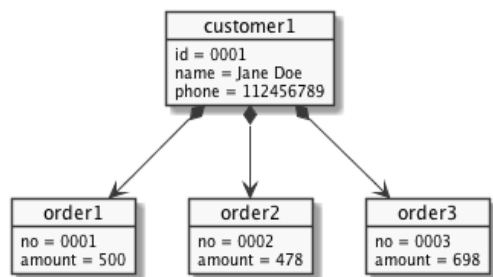
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

2. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.
3. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.
4. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de um modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam C

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |       |  |
|-------|--|
| False | A sintaxe abstrata apenas lida com a notação textual.  |
| True  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |
| False | O MOF não permite relações de herança.   |
| False | O OCL é uma exemplo de general-purpose language.   |
| False | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| False | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| True  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| True  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| False | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| False | Um modelo de domínio é uma parte fundamental da camada UI.   |
| True  | Num modelo de domínio os repositórios permitem persistir entidades.  |
| False | O Acceleo é uma ferramenta de transformação entre modelos.   |
| True  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| True  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| False | Existe uma relação de um-para-um entre modelos e diagramas.  |
| False | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| True  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| False | Em OCL o termo <b>this</b> é usado para se aceder à instância de contexto.   |
| True  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| True  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Admitindo uma abordagem de linhas de produtos de software...

(a)	existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
(b)	existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
(c)	não se distingue engenharia de domínio e engenharia de aplicação.
(d)	nenhuma das anteriores.
2. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...

(a)	a ferramenta Acceleo é orientada à geração de modelos
(b)	a ferramenta ATL é orientada à geração de modelos através da aplicação de <i>templates</i>
(c)	a ferramenta Acceleo é orientada à geração de texto através da aplicação de <i>templates</i>
(d)	nenhuma das anteriores.

3. Nas linhas de produtos de software. . .

- (a) usa-se aspect-oriented programming para modelar a variailidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

4. Num modelo de domínio. . .

- (a) os *value objects* podem conter referências para *entities*.
- (b) os *value objects* não podem fazer parte de *aggregates*.
- (c) os *value objects* não podem ser parâmetros de *services*.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

(c)	Linguagem para especificar constrangimentos	(a)	ecore
(f)	Ferramenta para DSL textuais	(b)	State diagram
(g)	Linguagem de transformação <i>model-to-text</i>	(c)	OCL
(h)	Linguagem de transformação <i>model-to-model</i>	(d)	Class diagram
(e)	Linhas de produtos de software	(e)	Feature model
(d)	UML Structure Diagram	(f)	Xtext
(a)	Meta-metamodelo	(g)	Acceleo
(b)	UML Behavior Diagram	(h)	ATL

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.
2. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.

## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.
2. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.
3. Se necessitássemos de modelar o endereço postal do *Customer* apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
4. A associação investments representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. O Set e o Bag são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.
2. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
3. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].
4. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade `rootTopics` na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

```

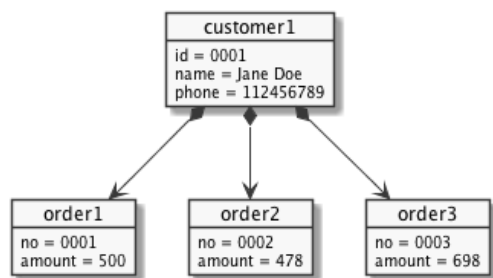
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____>asSet())->asOrderedSet();

```

2. Indique o que falta na linha 19 (nos "..."): subtopics.
3. Indique o que falta na expressão do invariante da linha 20: self.end > self.start.
4. Indique o que falta na linha 18 (nos "..."): parent.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.



## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.
- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.
- \_\_\_\_\_ O Acceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ O MOF não permite relações de herança.
- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.

2. Num modelo de domínio. . .
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.
3. Admitindo uma abordagem de linhas de produtos de software. . .
  - (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
  - (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
  - (c) não se distingue engenharia de domínio e engenharia de aplicação.
  - (d) nenhuma das anteriores.
4. Nas linhas de produtos de software. . .
  - (a) usa-se aspect-oriented programming para modelar a variailidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Linguagem para especificar constrangimentos	(a) Acceleo
_____ Linguagem de transformação <i>model-to-model</i>	(b) Feature model
_____ Linhas de produtos de software	(c) ATL
_____ Linguagem de transformação <i>model-to-text</i>	(d) ecore
_____ UML Structure Diagram	(e) Class diagram
_____ Meta-metamodelo	(f) OCL
_____ Ferramenta para DSL textuais	(g) State diagram
_____ UML Behavior Diagram	(h) Xtext

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.**
2. **MDE.**

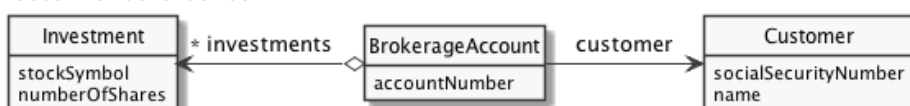
### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
2. A associação \_\_\_\_\_ representa uma coleção.
3. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.
4. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].
2. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].
3. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o \_\_\_\_\_ não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

```

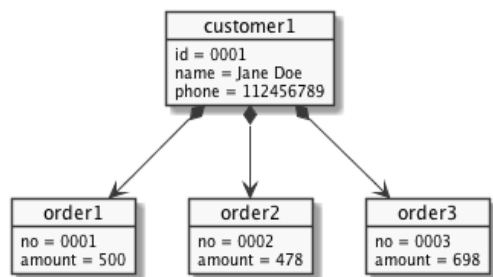
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

3. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.
4. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de um modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam D

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |       |  |
|-------|--|
| False | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| True  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| True  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| False | Existe uma relação de um-para-um entre modelos e diagramas.  |
| True  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |
| True  | Num modelo de domínio os repositórios permitem persistir entidades.  |
| False | A sintaxe abstrata apenas lida com a notação textual.  |
| True  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| True  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| False | Em OCL o termo <b>this</b> é usado para se aceder à instância de contexto.   |
| False | Um modelo de domínio é uma parte fundamental da camada UI.   |
| False | O OCL é uma exemplo de general-purpose language.   |
| False | O Acceleo é uma ferramenta de transformação entre modelos.   |
| False | O MOF não permite relações de herança.   |
| False | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| True  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| False | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| False | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| True  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| True  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.
2. Num modelo de domínio...
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.

3. Admitindo uma abordagem de linhas de produtos de software...

- (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

4. Nas linhas de produtos de software...

- (a) usa-se aspect-oriented programming para modelar a variabilidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

(f)	Linguagem para especificar constrangimentos	(a)	Acceleo
(c)	Linguagem de transformação <i>model-to-model</i>	(b)	Feature model
(b)	Linhas de produtos de software	(c)	ATL
(a)	Linguagem de transformação <i>model-to-text</i>	(d)	ecore
(e)	UML Structure Diagram	(e)	Class diagram
(d)	Meta-metamodelo	(f)	OCL
(h)	Ferramenta para DSL textuais	(g)	State diagram
(g)	UML Behavior Diagram	(h)	Xtext

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.
2. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.

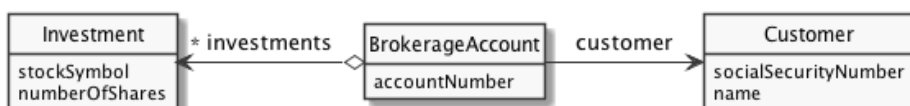
## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitassemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
2. A associação investments representa uma coleção.
3. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.
4. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].
2. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].
3. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.oclIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na linha 19 (nos "..."): subtopics.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

```

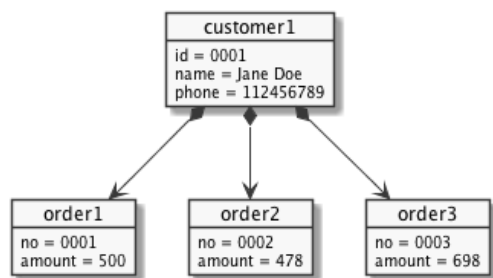
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet()->asOrderedSet());

```

3. Indique o que falta na expressão do invariante da linha 20: self.end > self.start.
4. Indique o que falta na linha 18 (nos "..."): parent.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.



## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ O MOF não permite relações de herança.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papeis no processo de desenvolvimento de software.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ O Aceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Nas linhas de produtos de software...
  - (a) usa-se aspect-oriented programming para modelar a variailidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.

2. Num modelo de domínio. . .
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.
3. Admitindo uma abordagem de linhas de produtos de software. . .
  - (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
  - (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
  - (c) não se distingue engenharia de domínio e engenharia de aplicação.
  - (d) nenhuma das anteriores.
4. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Linhas de produtos de software	(a) Class diagram
_____ UML Structure Diagram	(b) State diagram
_____ UML Behavior Diagram	(c) Feature model
_____ Ferramenta para DSL textuais	(d) ATL
_____ Meta-metamodelo	(e) Acceleo
_____ Linguagem de transformação <i>model-to-model</i>	(f) Xtext
_____ Linguagem para especificar constrangimentos	(g) OCL
_____ Linguagem de transformação <i>model-to-text</i>	(h) ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.**
2. **MDE.**

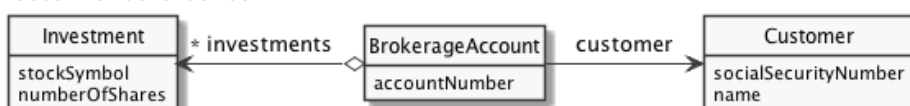
### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.
2. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
3. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.
4. A associação \_\_\_\_\_ representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
2. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o \_\_\_\_\_ não aceita duplicados.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].
4. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

```

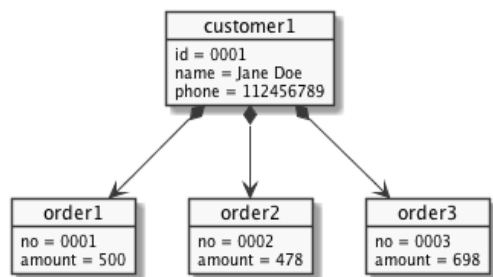
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet())->asOrderedSet();

```

2. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.
3. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.
4. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de um modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam E

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |       |  |
|-------|--|
| False | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| False | Em OCL o termo <b>this</b> é usado para se aceder à instância de contexto.   |
| True  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| False | Existe uma relação de um-para-um entre modelos e diagramas.  |
| True  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |
| False | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| False | A sintaxe abstrata apenas lida com a notação textual.  |
| True  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |
| True  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| False | O MOF não permite relações de herança.   |
| True  | Num modelo de domínio os repositórios permitem persistir entidades.  |
| True  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| True  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| False | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| False | Um modelo de domínio é uma parte fundamental da camada UI.   |
| True  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| False | O Acceleo é uma ferramenta de transformação entre modelos.   |
| True  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| False | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| False | O OCL é uma exemplo de general-purpose language.   |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Nas linhas de produtos de software...
  - (a) usa-se aspect-oriented programming para modelar a variailidade.
  - (b) usam-se modelos de *features* para implementar a variabilidade
  - (c) usam-se modelos de *features* para modelar a variabilidade
  - (d) usa-se aspect-oriented programming para implementar as partes comuns.
2. Num modelo de domínio...
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.

3. Admitindo uma abordagem de linhas de produtos de software. . .

- (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

4. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...

- (a) a ferramenta Acceleo é orientada à geração de modelos
- (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
- (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

(c) Linhas de produtos de software	(a) Class diagram
(a) UML Structure Diagram	(b) State diagram
(b) UML Behavior Diagram	(c) Feature model
(f) Ferramenta para DSL textuais	(d) ATL
(h) Meta-metamodelo	(e) Acceleo
(d) Linguagem de transformação <i>model-to-model</i>	(f) Xtext
(g) Linguagem para especificar constrangimentos	(g) OCL
(e) Linguagem de transformação <i>model-to-text</i>	(h) ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.
2. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.

## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.
2. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
3. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.
4. A associação investments representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
2. O Set e o Bag são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].
4. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade `rootTopics` na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

```

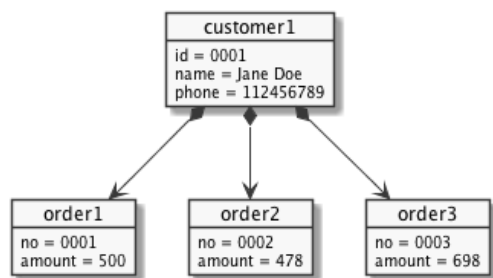
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____>asSet())->asOrderedSet();

```

2. Indique o que falta na linha 19 (nos "..."): subtopics.
3. Indique o que falta na linha 18 (nos "..."): parent.
4. Indique o que falta na expressão do invariante da linha 20: self.end > self.start.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.



## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°1**"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- \_\_\_\_\_ Um modelo de domínio é uma parte fundamental da camada UI.
- \_\_\_\_\_ É comum usar-se o OCL para especificar o corpo dos métodos.
- \_\_\_\_\_ A sintaxe abstrata apenas lida com a notação textual.
- \_\_\_\_\_ As regras do ATL do tipo *matched rules* são exemplos típicos de construções imperativas.
- \_\_\_\_\_ PSM é uma sigla MDA que significa modelo independente da plataforma.
- \_\_\_\_\_ Existe uma relação de um-para-um entre modelos e diagramas.
- \_\_\_\_\_ O MOF não permite relações de herança.
- \_\_\_\_\_ Num modelo de domínio os repositórios permitem persistir entidades.
- \_\_\_\_\_ Em OCL o termo **this** é usado para se aceder à instância de contexto.
- \_\_\_\_\_ Num modelo de domínio os objectos com identidade podem ser *value objects* ou *entities*.
- \_\_\_\_\_ As linhas de produtos de software são estratégias de reutilização dentro das organizações.
- \_\_\_\_\_ As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida.
- \_\_\_\_\_ O Acceleo é uma ferramenta de transformação entre modelos.
- \_\_\_\_\_ O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.
- \_\_\_\_\_ A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.
- \_\_\_\_\_ Os modelos *business process* são exemplos de modelos independentes de computação.
- \_\_\_\_\_ O mecanismo de *profiles* do UML permite estender o UML.
- \_\_\_\_\_ Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.
- \_\_\_\_\_ As transformações intra-modelos são baseadas na teoria dos grafos.
- \_\_\_\_\_ O OCL é uma exemplo de general-purpose language.

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.

2. Num modelo de domínio. . .

- (a) os *value objects* podem conter referências para *entities*.
- (b) os *value objects* não podem fazer parte de *aggregates*.
- (c) os *value objects* não podem ser parâmetros de *services*.
- (d) nenhuma das anteriores.

3. Nas linhas de produtos de software. . .

- (a) usa-se aspect-oriented programming para modelar a variailidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

4. Admitindo uma abordagem de linhas de produtos de software. . .

- (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

_____ Linguagem para especificar constrangimentos	(a) OCL
_____ UML Structure Diagram	(b) ATL
_____ UML Behavior Diagram	(c) Acceleo
_____ Linguagem de transformação <i>model-to-text</i>	(d) Xtext
_____ Ferramenta para DSL textuais	(e) Class diagram
_____ Linguagem de transformação <i>model-to-model</i>	(f) State diagram
_____ Meta-metamodelo	(g) Feature model
_____ Linhas de produtos de software	(h) ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.**
2. **MDE.**

### Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como **"Folha Anexa N°2"**
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitássemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como \_\_\_\_\_ [Entity — Value Object — Service].
2. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o \_\_\_\_\_.
3. Dos três conceitos ilustrados no diagrama apenas o \_\_\_\_\_ faz sentido que seja modelado como *Value Object*.
4. A associação \_\_\_\_\_ representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é \_\_\_\_\_ [horizontal — vertical].
2. Num processo de desenvolvimento de software baseado em modelos será vulgar termos \_\_\_\_\_ [apenas um — vários] metamodelos.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de \_\_\_\_\_ [model interpretation — code generation].
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o \_\_\_\_\_ não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: \_\_\_\_\_.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é \_\_\_\_\_ e o que falta no segundo espaço é \_\_\_\_\_.

```

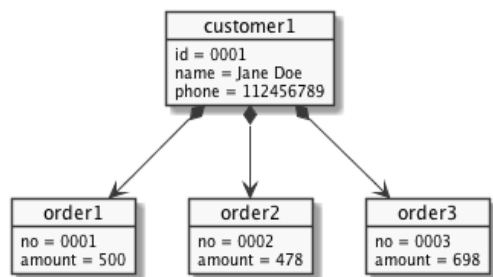
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet()->asOrderedSet());

```

3. Indique o que falta na linha 19 (nos "..."): \_\_\_\_\_.
4. Indique o que falta na linha 18 (nos "..."): \_\_\_\_\_.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidades de metamodelação do **ecore**.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de um modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame.

# Answer Key for Exam F

## Parte 1

### Atenção:

1. As respostas do Grupo IV devem ser colocadas numa folha em branco identificada como "Folha Anexa Nº1"
2. Deve colocar o seu número e nome no topo da folha anexa

### Grupo I (4v)

Assinale com V as afirmações verdadeiras e com F as falsas. **Duas respostas erradas descontam uma certa.**

- |              |  |
|--------------|--|
| <u>False</u> | Um modelo de domínio é uma parte fundamental da camada UI.   |
| <u>True</u>  | É comum usar-se o OCL para especificar o corpo dos métodos.  |
| <u>False</u> | A sintaxe abstrata apenas lida com a notação textual.  |
| <u>False</u> | As regras do ATL do tipo <i>matched rules</i> são exemplos típicos de construções imperativas.   |
| <u>False</u> | PSM é uma sigla MDA que significa modelo independente da plataforma.   |
| <u>False</u> | Existe uma relação de um-para-um entre modelos e diagramas.  |
| <u>False</u> | O MOF não permite relações de herança.   |
| <u>True</u>  | Num modelo de domínio os repositórios permitem persistir entidades.  |
| <u>False</u> | Em OCL o termo <code>this</code> é usado para se aceder à instância de contexto.   |
| <u>False</u> | Num modelo de domínio os objectos com identidade podem ser <i>value objects</i> ou <i>entities</i> .                                   |
| <u>True</u>  | As linhas de produtos de software são estratégias de reutilização dentro das organizações.   |
| <u>False</u> | As linhas de produtos de software são abordagens que normalmente justificam o seu custo logo a partir da primeira aplicação produzida. |
| <u>False</u> | O Acceleo é uma ferramenta de transformação entre modelos.   |
| <u>True</u>  | O MDSE introduz novas tarefas e papéis no processo de desenvolvimento de software.   |
| <u>True</u>  | A gramática de uma linguagem está para um programa como um metamodelo está para um modelo.   |
| <u>True</u>  | Os modelos <i>business process</i> são exemplos de modelos independentes de computação.  |
| <u>True</u>  | O mecanismo de <i>profiles</i> do UML permite estender o UML.  |
| <u>True</u>  | Um dos objectivos do MDSE é diminuir a distância conceptual entre a análise e a implementação no desenvolvimento de software.          |
| <u>True</u>  | As transformações intra-modelos são baseadas na teoria dos grafos.   |
| <u>False</u> | O OCL é uma exemplo de general-purpose language.   |

### Grupo II (1,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Duas respostas erradas descontam uma certa.**

1. Um dos aspectos fundamentais do MDE é a geração de artefactos. Relativamente à geração...
  - (a) a ferramenta Acceleo é orientada à geração de modelos
  - (b) a ferramenta ATL é orientada à geração de modelos através da aplicação de *templates*
  - (c) a ferramenta Acceleo é orientada à geração de texto através da aplicação de *templates*
  - (d) nenhuma das anteriores.
2. Num modelo de domínio...
  - (a) os *value objects* podem conter referências para *entities*.
  - (b) os *value objects* não podem fazer parte de *aggregates*.
  - (c) os *value objects* não podem ser parâmetros de *services*.
  - (d) nenhuma das anteriores.

3. Nas linhas de produtos de software. . .

- (a) usa-se aspect-oriented programming para modelar a variailidade.
- (b) usam-se modelos de *features* para implementar a variabilidade
- ☒ (c) usam-se modelos de *features* para modelar a variabilidade
- (d) usa-se aspect-oriented programming para implementar as partes comuns.

4. Admitindo uma abordagem de linhas de produtos de software. . .

- ☒ (a) existem dois níveis de engenharia: engenharia de domínio e engenharia de aplicação.
- (b) existem três níveis de engenharia: engenharia de domínio, engenharia de aplicação e engenharia de processo.
- (c) não se distingue engenharia de domínio e engenharia de aplicação.
- (d) nenhuma das anteriores.

### Grupo III (1,5v)

Associe as ferramentas da segunda coluna com os conceitos da primeira coluna. Duas respostas erradas descontam uma certa.

<u>(a)</u>	Linguagem para especificar constrangimentos	(a)	OCCL
<u>(e)</u>	UML Structure Diagram	(b)	ATL
<u>(f)</u>	UML Behavior Diagram	(c)	Acceleo
<u>(c)</u>	Linguagem de transformação <i>model-to-text</i>	(d)	Xtext
<u>(d)</u>	Ferramenta para DSL textuais	(e)	Class diagram
<u>(b)</u>	Linguagem de transformação <i>model-to-model</i>	(f)	State diagram
<u>(h)</u>	Meta-metamodelo	(g)	Feature model
<u>(g)</u>	Linhas de produtos de software	(h)	ecore

### Grupo IV (2,5v) - Responda na folha anexa N°1

Apresente (em folha anexa a entregar no final do exam) uma breve **definição** e **descrição** para os seguintes conceitos.

1. **SPL - Software Product Line.** As linhas de produtos de software são uma abordagem pragmática à reutilização de software "dentro" das organizações. A ideia é construir várias aplicações dentro de um mesmo domínio, reutilizando artefactos e conhecimento à medida que se desenvolvem as aplicações. Divide-se em duas grandes actividades: engenharia de domínio e engenharia de aplicações.
2. **MDE.** MDE (Model Driven Engineering) é uma abordagem à engenharia baseada em modelos. Nesta abordagem os modelos são o artefacto principal e assumem uma importância tão grande quanto o código. Costuma dizer-se que nesta abordagem modelos + transformações = software.

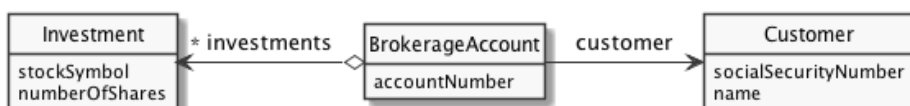
## Parte 2

**Atenção:**

1. Todas as justificações e respostas por extenso desta parte devem ser colocadas numa folha em branco identificada como "**Folha Anexa N°2**"
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número e nome no topo da folha anexa

### Grupo V (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato de um modelo de domínio **Preencha os espaços em falta de forma a completar correctamente o texto.**



1. Se necessitassemos de modelar o endereço postal do **Customer** apenas para este contexto devíamos fazê-lo como Value Object [Entity — Value Object — Service].
2. No caso da *Entity Customer* o seu atributo que podemos usar para a sua identidade é o socialSecurityNumber.
3. Dos três conceitos ilustrados no diagrama apenas o Investment faz sentido que seja modelado como *Value Object*.
4. A associação investments representa uma coleção.

## Grupo VI (2v) - Justifique na folha anexa N°2

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. A linguagem SQL é um exemplo de uma linguagem específica de domínio cujo "focus" é horizontal [horizontal — vertical].
2. Num processo de desenvolvimento de software baseado em modelos será vulgar termos vários [apenas um — vários] metamodelos.
3. Podemos afirmar que a utilização que o OFBiz faz dos modelos de entidades (e.g., entitymodel.xml) é um exemplo de model interpretation [model interpretation — code generation].
4. O **Set** e o **Bag** são dois tipos de coleções do OCL. Destas duas o Set não aceita duplicados.

## Grupo VII (2v) - Justifique na folha anexa N°2

Considere o seguinte extrato produzido pela ferramenta OCLinEcore e relativo a um metamodelo estudado nas aulas. O extrato está incompleto.

```

1 class _'Map' {
2   attribute title : String[1];
3   attribute created : ecore::EDate[?];
4   property elements : MapElement[*] { ordered composes };
5   property rootTopics : Topic[*] { ordered derived transient volatile } {
6     initial: ...
7   }
8   invariant mustHaveTitle: not title.ocIsUndefined();
9 }
10 class Topic extends MapElement {
11   operation allSubTopics() : Topic[*] {
12     body: self->closure(subtopics);
13   }
14   attribute description : String[?] { id };
15   attribute start : ecore::EDate[?];
16   attribute end : ecore::EDate[?];
17   attribute priority : Priority[?];
18   property ... #subtopics : Topic[?];
19   property ... #parent : Topic[*];
20   invariant EndAfterStart: ...
21 }
22 ...

```

**Preencha os espaços em falta de forma a completar correctamente o texto.**

1. Indique o que falta na expressão do invariante da linha 20: self.end > self.start.
2. Nos espaços do seguinte extrato (que se referem à inicialização da propriedade **rootTopics** na linha 6) o que falta no primeiro espaço é mindmap::Topic e o que falta no segundo espaço é topics.subtopics.

```

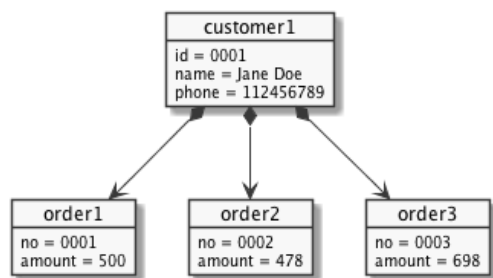
let topics : Set(mindmap::Topic) =
  self.elements->select(oclIsKindOf(_____))->collect(oclAsType(mindmap::Topic))->asSet() in
  topics->asOrderedSet()->symmetricDifference(topics._____->asSet()->asOrderedSet());

```

3. Indique o que falta na linha 19 (nos "..."): subtopics.
4. Indique o que falta na linha 18 (nos "..."): parent.

## Grupo VIII (3v) - Resposta na folha anexa N°2

Considere o seguinte diagrama de um modelo de objectos (instâncias) que foi estudado nas aulas.



1. Apresente em folha anexa um diagrama que ilustre um possível metamodelo para o diagrama anterior (use notação de diagramas de classes). Use os construtores possíveis no **ecore**. Não necessita de ser preciso na notação mas é necessário que explique o modelo que apresentar no contexto das possibilidade de metamodelação do **ecore**. Colocar aqui a resposta.

## Grupo IX (1,5v) - Resposta na folha anexa N°2

Considere os seguintes metamodelos ecore apresentados em sintaxe OCLinEcore.

```
1 package abc : abc = 'http://www.isep.ipp.pt/edom/2015/abc' {
2   class A {
3     attribute name : String[?];
4     property elements : B[*] { ordered composes };
5   }
6   class B {
7     attribute name : String[?];
8   }
9   class ABC {
10    property abcs : A[*] { ordered composes };
11  }
12 }
```

```
1 package xpto : xpto = 'http://www.isep.ipp.pt/edom/2015/xpto' {
2   class XPTO {
3     property elements : X[*] { ordered composes };
4   }
5   class X {
6     attribute description : String[?];
7     property childs : Y[*] { ordered composes };
8   }
9   class Y {
10    attribute description : String[?];
11  }
12 }
```

Considere ainda o seguinte extrato de uma modelo ATL de transformação de modelos ABC em modelos XPTO.

```
1 module abc2xpto;
2 create OUT : XPTO from IN : ABC;
3 rule abc2xpto {
4   from
5     abc: ABC!ABC
6   to
7     xpto: XPTO!XPTO ( elements <- abc.abcs )
8 }
9 rule a2x {
10  from
11    a: ABC!A
12  to
13    x: XPTO!X ( description <- a.name )
14 }
```

1. A transformação apresentada gera elementos X com base em elementos A. Altere/complete a transformação de forma a gerar elementos Y com base em elementos B, gerando também as correspondentes relações de composição. Apresente a sua resposta em folha anexa que deve entregar no final do exame. colocar aqui a resposta.