

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

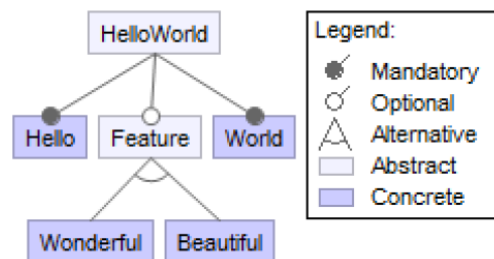
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ Os Value Objects não devem referenciar Entities.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level".
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.

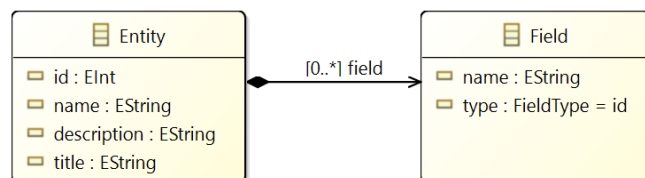


1. Uma abordagem SPL...
 - (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - (d) nenhuma das respostas anteriores
2. O modelo apresentado...
 - (a) permite representar no mínimo 2 programas e no máximo 4
 - (b) permite representar no máximo 3 programas
 - (c) permite representar no mínimo 3 programas
 - (d) nenhuma das respostas anteriores
3. Técnicas de implementação de SPL incluem...
 - (a) *Feature-oriented programming* e *functional programming*
 - (b) *Feature-oriented programming* e *Aspect-oriented programming*
 - (c) *Feature-oriented programming* e *domain analysis*
 - (d) nenhuma das respostas anteriores
4. Sobre a variabilidade numa linha de produtos...
 - (a) não interessa para a engenharia de domínio
 - (b) não é capturada num *feature model*
 - (c) é capturada apenas com *features* marcadas como *optional*
 - (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).
2. Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

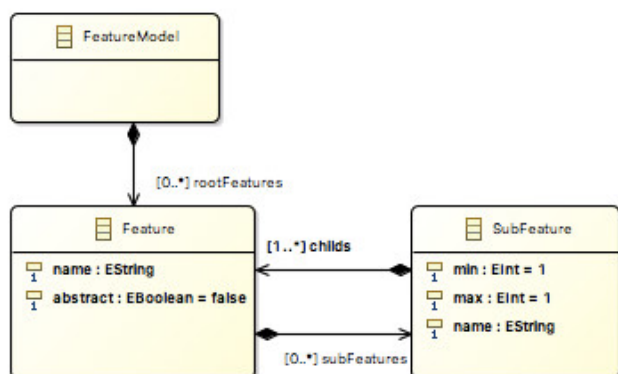
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
  
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro ofbiz.cs semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $);
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
  
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.
2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

Answer Key for Exam A

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

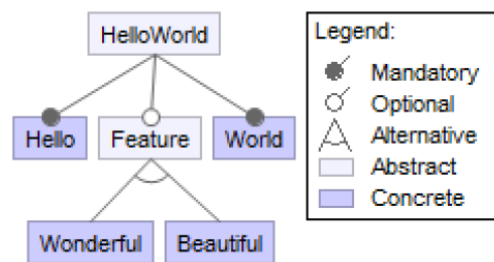
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- False Todas as partes de um CIM devem ter uma correspondência na implementação.
- False Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- False Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- False A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- True A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- False O MOF permite associações unárias, binárias e ternárias entre classes.
- True O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- False Os Value Objects não devem referenciar Entities.
- False Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- True Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- True O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- False Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- True A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- False Associações bidirecionais entre *entities* não são permitidas.
- True No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level".
- True A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- False As abordagens "Agile" e o MDSE não são compatíveis.
- True O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore.
- False Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- True Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. Uma abordagem SPL...

- ☐ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- ☐ (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- ☐ (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- ☐ (d) nenhuma das respostas anteriores

2. O modelo apresentado...

- ☐ (a) permite representar no mínimo 2 programas e no máximo 4
- ☐ (b) permite representar no máximo 3 programas
- ☐ (c) permite representar no mínimo 3 programas
- ☐ (d) nenhuma das respostas anteriores

3. Técnicas de implementação de SPL incluem...

- ☐ (a) *Feature-oriented programming* e *functional programming*
- ☐ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- ☐ (c) *Feature-oriented programming* e *domain analysis*
- ☐ (d) nenhuma das respostas anteriores

4. Sobre a variabilidade numa linha de produtos...

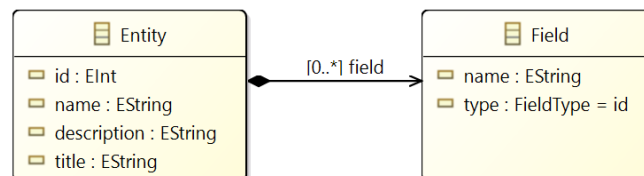
- ☐ (a) não interessa para a engenharia de domínio
- ☐ (b) não é capturada num *feature model*
- ☐ (c) é capturada apenas com *features* marcadas como *optional*
- ☐ (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

2. Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

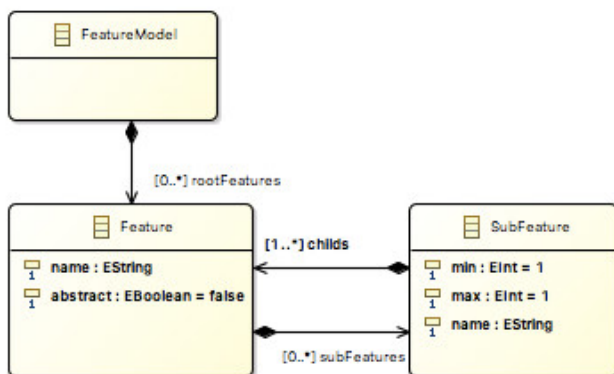
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
    
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```



```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Acceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14   [/if]
15
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20   aClass.concat('.java')]
21 /]
```

2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7
8   [file (anEntity.name.classFileName(), false, 'UTF-8')]
9
10 public class [anEntity.name.toUpperFirst()/] {
11   [for (aField:Field | anEntity.field)]
12   String [aField.name/];
13   [/for]
14
15 }
16   [/file]
17
```

```
18 [/for]
19 [/template]
20
21 [query public classFileName(aClass: String): String =
22     aClass.concat('.java')
23 /]
```

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

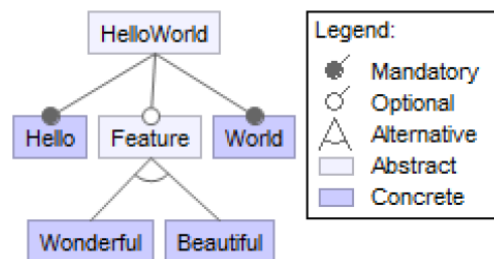
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- _____ Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- _____ Os Value Objects não devem referenciar Entities.
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado {4}.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado {false, false, false, false}.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



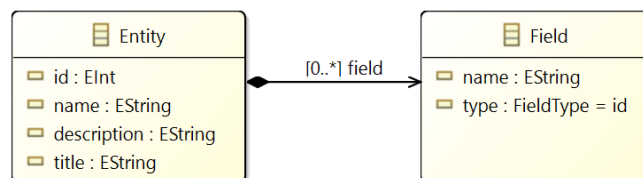
- O modelo apresentado...
 - permite representar no mínimo 2 programas e no máximo 4
 - permite representar no máximo 3 programas
 - permite representar no mínimo 3 programas
 - nenhuma das respostas anteriores
- Uma abordagem SPL...
 - no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - nenhuma das respostas anteriores
- Técnicas de implementação de SPL incluem...
 - Feature-oriented programming* e *functional programming*
 - Feature-oriented programming* e *Aspect-oriented programming*
 - Feature-oriented programming* e *domain analysis*
 - nenhuma das respostas anteriores
- Sobre a variabilidade numa linha de produtos...
 - não interessa para a engenharia de domínio
 - não é capturada num *feature model*
 - é capturada apenas com *features* marcadas como *optional*
 - nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

- Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.

Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.



- Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).

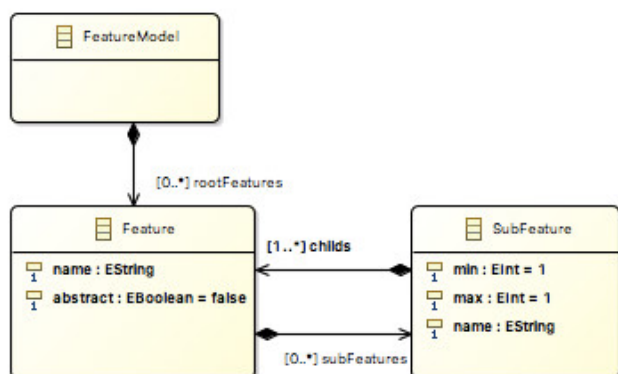
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro ofbiz.cs semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
    
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.
2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

Answer Key for Exam B

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

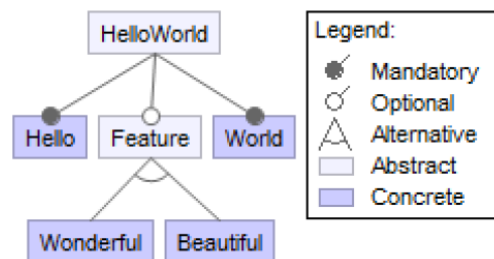
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- False O MOF permite associações unárias, binárias e ternárias entre classes.
- False Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- False As abordagens "Agile" e o MDSE não são compatíveis.
- False Associações bidirecionais entre *entities* não são permitidas.
- False A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- True O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- False Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- True No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- False Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- True O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore.
- True Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- True A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- False Os Value Objects não devem referenciar Entities.
- True A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- True O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- False Todas as partes de um CIM devem ter uma correspondência na implementação.
- False Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- True A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- True Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- False Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. O modelo apresentado...

- (a) permite representar no mínimo 2 programas e no máximo 4
- ☒ (b) permite representar no máximo 3 programas
- (c) permite representar no mínimo 3 programas
- (d) nenhuma das respostas anteriores

2. Uma abordagem SPL...

- ☒ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- (d) nenhuma das respostas anteriores

3. Técnicas de implementação de SPL incluem...

- (a) *Feature-oriented programming* e *functional programming*
- ☒ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- (c) *Feature-oriented programming* e *domain analysis*
- (d) nenhuma das respostas anteriores

4. Sobre a variabilidade numa linha de produtos...

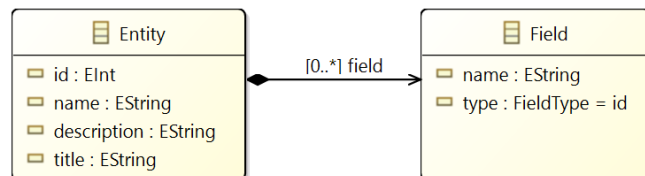
- (a) não interessa para a engenharia de domínio
- (b) não é capturada num *feature model*
- (c) é capturada apenas com *features* marcadas como *optional*
- ☒ (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.

Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.



Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

2. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

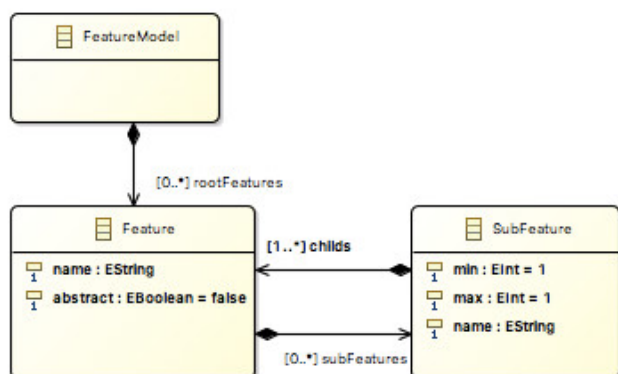
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
    
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```

```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8
9   public class [anEntity.name.toUpperFirst()/] {
10     [for (aField:Field | anEntity.field)]
11       String [aField.name/];
12     [/for]
13   }
14   [/file]
15 [/for]
16 [/template]
17
18 [query public classFileName(aClass: String): String =
19   aClass.concat('.java')]
20
21 [query public classFileName(aClass: String): String =
22   aClass.concat('.java')]
23 /]
```

2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14 [/if]
15
```

```
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20     aClass.concat('.java')
21 /]
```

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

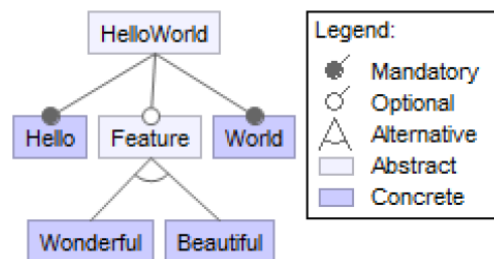
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.
- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- _____ Os Value Objects não devem referenciar Entities.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ Em Aceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.

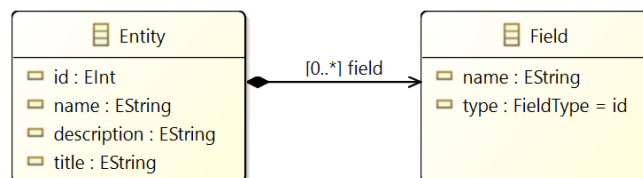


- O modelo apresentado...
 - permite representar no mínimo 2 programas e no máximo 4
 - permite representar no máximo 3 programas
 - permite representar no mínimo 3 programas
 - nenhuma das respostas anteriores
- Técnicas de implementação de SPL incluem...
 - Feature-oriented programming* e *functional programming*
 - Feature-oriented programming* e *Aspect-oriented programming*
 - Feature-oriented programming* e *domain analysis*
 - nenhuma das respostas anteriores
- Uma abordagem SPL...
 - no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - nenhuma das respostas anteriores
- Sobre a variabilidade numa linha de produtos...
 - não interessa para a engenharia de domínio
 - não é capturada num *feature model*
 - é capturada apenas com *features* marcadas como *optional*
 - nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

- Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).
- Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

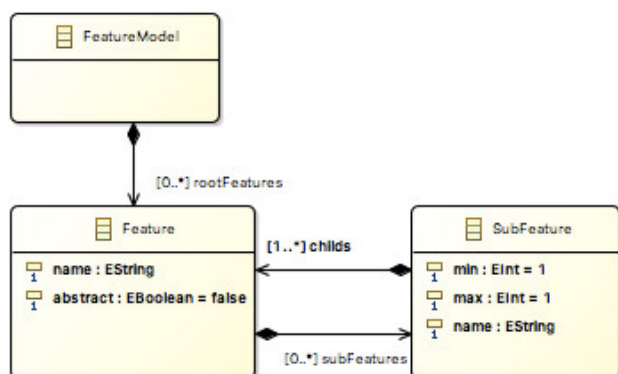
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro ofbiz.cs semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $);
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
    
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.
2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

Answer Key for Exam C

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

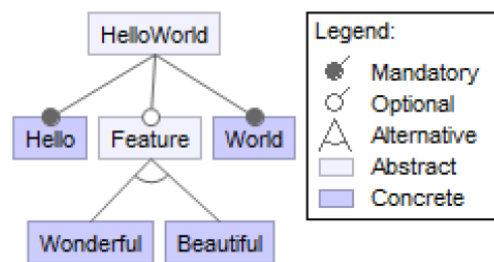
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- | | |
|-------|---|
| False | Todas as partes de um CIM devem ter uma correspondência na implementação. |
| False | Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado. |
| True | A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar. |
| False | Os Value Objects não devem referenciar Entities. |
| False | A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software. |
| False | Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space". |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->select(x x > 3)</code> tem como resultado <code>{4}</code> . |
| False | O MOF permite associações unárias, binárias e ternárias entre classes. |
| False | Em Aceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output. |
| True | Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura. |
| True | O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore. |
| True | No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level" |
| False | As abordagens "Agile" e o MDSE não são compatíveis. |
| True | O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes. |
| True | O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para. |
| False | Associações bidirecionais entre <i>entities</i> não são permitidas. |
| False | Os Factory são apropriados para criação de objectos mas não para a sua reconstituição. |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->collect(x x > 4)</code> tem como resultado <code>{false, false, false, false}</code> . |
| False | Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão. |
| True | Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra. |

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. O modelo apresentado...

- (a) permite representar no mínimo 2 programas e no máximo 4
- ☒ (b) permite representar no máximo 3 programas
- (c) permite representar no mínimo 3 programas
- (d) nenhuma das respostas anteriores

2. Técnicas de implementação de SPL incluem...

- (a) *Feature-oriented programming* e *functional programming*
- ☒ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- (c) *Feature-oriented programming* e *domain analysis*
- (d) nenhuma das respostas anteriores

3. Uma abordagem SPL...

- ☒ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- (d) nenhuma das respostas anteriores

4. Sobre a variabilidade numa linha de produtos...

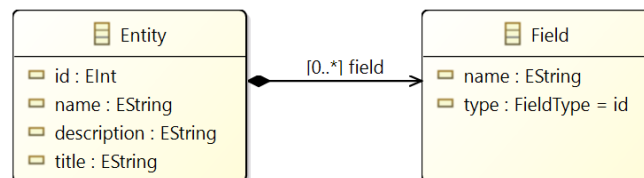
- (a) não interessa para a engenharia de domínio
- (b) não é capturada num *feature model*
- (c) é capturada apenas com *features* marcadas como *optional*
- ☒ (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

2. Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

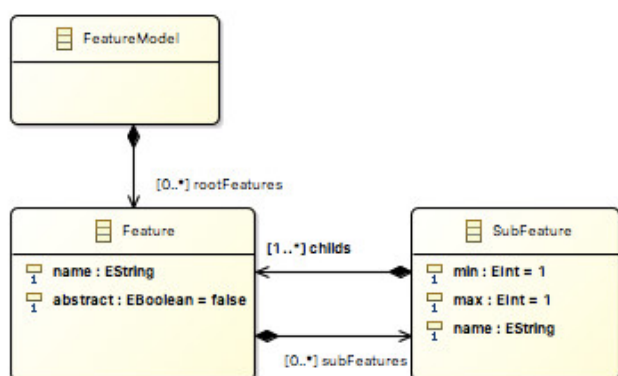
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
    
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```

```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Acceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14 [/if]
15
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20   aClass.concat('.java')]
21 /]
```

2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7
8   [file (anEntity.name.classFileName(), false, 'UTF-8')]
9
10 public class [anEntity.name.toUpperFirst()/] {
11   [for (aField:Field | anEntity.field)]
12   String [aField.name/];
13   [/for]
14
15 }
16   [/file]
17
```

```
18 [/for]
19 [/template]
20
21 [query public classFileName(aClass: String): String =
22     aClass.concat('.java')
23 /]
```

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

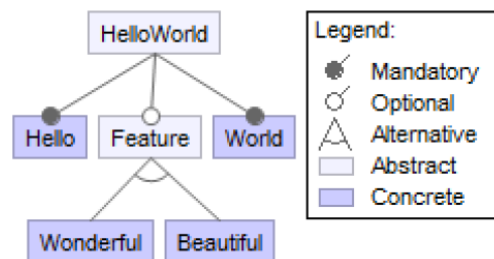
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- _____ Em Aceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ Os Value Objects não devem referenciar Entities.
- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.

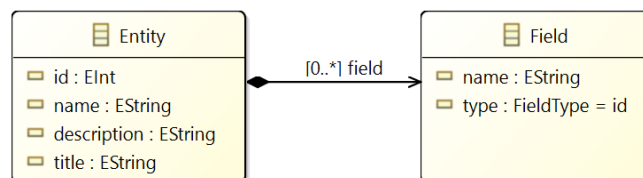


- O modelo apresentado...
 - permite representar no mínimo 2 programas e no máximo 4
 - permite representar no máximo 3 programas
 - permite representar no mínimo 3 programas
 - nenhuma das respostas anteriores
- Técnicas de implementação de SPL incluem...
 - Feature-oriented programming* e *functional programming*
 - Feature-oriented programming* e *Aspect-oriented programming*
 - Feature-oriented programming* e *domain analysis*
 - nenhuma das respostas anteriores
- Sobre a variabilidade numa linha de produtos...
 - não interessa para a engenharia de domínio
 - não é capturada num *feature model*
 - é capturada apenas com *features* marcadas como *optional*
 - nenhuma das respostas anteriores
- Uma abordagem SPL...
 - no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

- Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).
- Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

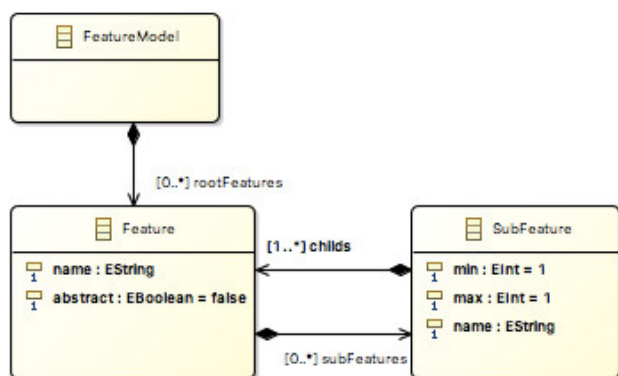
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
  
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro ofbiz.cs semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $);
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
  
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.
2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

Answer Key for Exam D

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

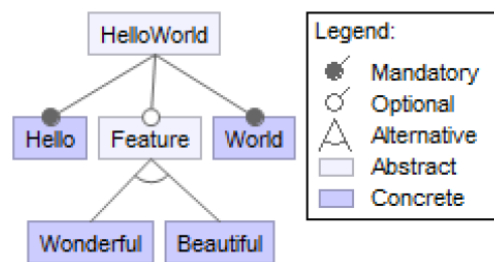
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- | | |
|-------|---|
| False | Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão. |
| False | Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space". |
| False | Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado. |
| True | Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura. |
| True | Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra. |
| True | O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes. |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->collect(x x > 4)</code> tem como resultado <code>{false, false, false, false}</code> . |
| True | No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level" |
| False | Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output. |
| False | O MOF permite associações unárias, binárias e ternárias entre classes. |
| False | Associações bidirecionais entre <i>entities</i> não são permitidas. |
| False | As abordagens "Agile" e o MDSE não são compatíveis. |
| True | A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar. |
| False | A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software. |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->select(x x > 3)</code> tem como resultado <code>{4}</code> . |
| True | O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore. |
| False | Os Factory são apropriados para criação de objectos mas não para a sua reconstituição. |
| True | O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para. |
| False | Os Value Objects não devem referenciar Entities. |
| False | Todas as partes de um CIM devem ter uma correspondência na implementação. |

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. O modelo apresentado...

- (a) permite representar no mínimo 2 programas e no máximo 4
- ☒ (b) permite representar no máximo 3 programas
- (c) permite representar no mínimo 3 programas
- (d) nenhuma das respostas anteriores

2. Técnicas de implementação de SPL incluem...

- (a) *Feature-oriented programming* e *functional programming*
- ☒ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- (c) *Feature-oriented programming* e *domain analysis*
- (d) nenhuma das respostas anteriores

3. Sobre a variabilidade numa linha de produtos...

- (a) não interessa para a engenharia de domínio
- (b) não é capturada num *feature model*
- (c) é capturada apenas com *features* marcadas como *optional*
- ☒ (d) nenhuma das respostas anteriores

4. Uma abordagem SPL...

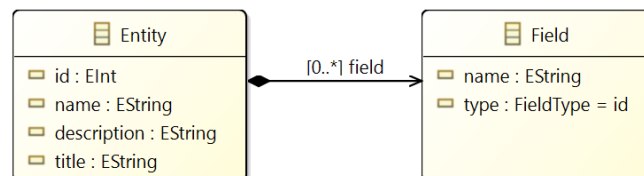
- ☒ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

2. Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

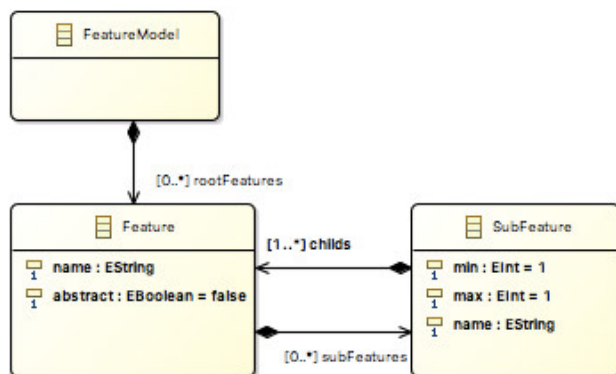
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
    
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```

```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8   [file (anEntity.name.classFileName(), false, 'UTF-8')]
9
10 public class [anEntity.name.toUpperFirst()/] {
11   [for (aField:Field | anEntity.field)]
12   String [aField.name/];
13   [/for]
14
15 }
16   [/file]
17
18 [/for]
19 [/template]
20
21 [query public classFileName(aClass: String): String =
22   aClass.concat('.java')]
23 /]
```

2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14   [/if]
15
```

```
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20     aClass.concat('.java')
21 /]
```

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

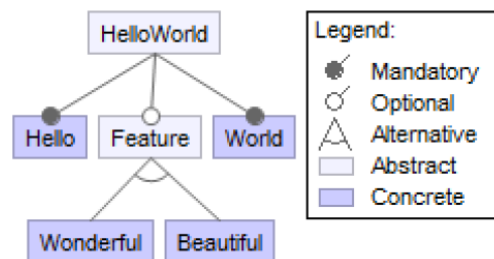
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- _____ Em Aceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.
- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.
- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado {4}.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado {false, false, false, false}.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ Os Value Objects não devem referenciar Entities.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.

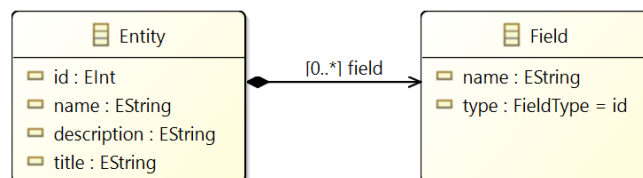


- O modelo apresentado...
 - permite representar no mínimo 2 programas e no máximo 4
 - permite representar no máximo 3 programas
 - permite representar no mínimo 3 programas
 - nenhuma das respostas anteriores
- Sobre a variabilidade numa linha de produtos...
 - não interessa para a engenharia de domínio
 - não é capturada num *feature model*
 - é capturada apenas com *features* marcadas como *optional*
 - nenhuma das respostas anteriores
- Uma abordagem SPL...
 - no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - nenhuma das respostas anteriores
- Técnicas de implementação de SPL incluem...
 - Feature-oriented programming* e *functional programming*
 - Feature-oriented programming* e *Aspect-oriented programming*
 - Feature-oriented programming* e *domain analysis*
 - nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

- Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).
- Considere o seguinte diagrama representativo de um extrato de um ficheiro `ofbiz.ecore` semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

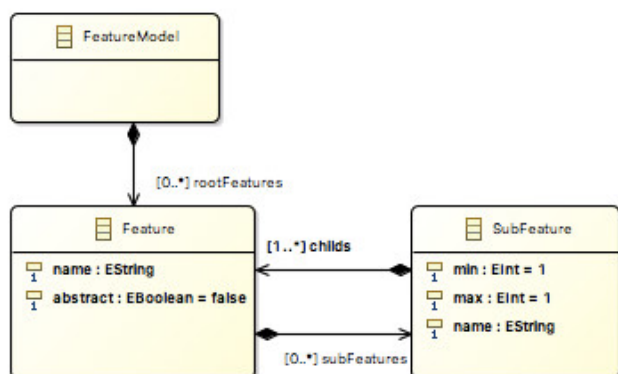
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro ofbiz.cs semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org/eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
    
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.
2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

Answer Key for Exam E

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

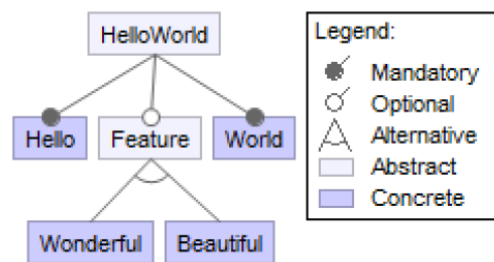
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- | | |
|-------|---|
| False | As abordagens "Agile" e o MDSE não são compatíveis. |
| True | Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra. |
| True | Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura. |
| False | Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output. |
| False | Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão. |
| False | Todas as partes de um CIM devem ter uma correspondência na implementação. |
| False | O MOF permite associações unárias, binárias e ternárias entre classes. |
| False | Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado. |
| True | O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para. |
| True | O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes. |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->select(x x > 3)</code> tem como resultado <code>{4}</code> . |
| False | Os Factory são apropriados para criação de objectos mas não para a sua reconstituição. |
| True | O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore. |
| True | A expressão OCL <code>Set{1, 2, 3, 4}->collect(x x > 4)</code> tem como resultado <code>{false, false, false, false}</code> . |
| False | Associações bidirecionais entre <i>entities</i> não são permitidas. |
| False | Os Value Objects não devem referenciar Entities. |
| False | A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software. |
| False | Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space". |
| True | No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level" |
| True | A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar. |

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. O modelo apresentado...

- (a) permite representar no mínimo 2 programas e no máximo 4
- ☒ (b) permite representar no máximo 3 programas
- (c) permite representar no mínimo 3 programas
- (d) nenhuma das respostas anteriores

2. Sobre a variabilidade numa linha de produtos...

- (a) não interessa para a engenharia de domínio
- (b) não é capturada num *feature model*
- (c) é capturada apenas com *features* marcadas como *optional*
- ☒ (d) nenhuma das respostas anteriores

3. Uma abordagem SPL...

- ☒ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- (d) nenhuma das respostas anteriores

4. Técnicas de implementação de SPL incluem...

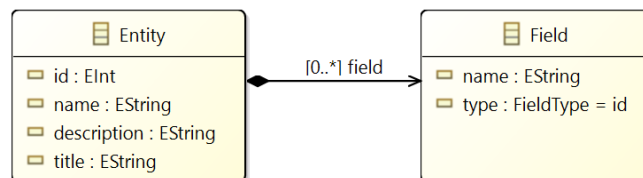
- (a) *Feature-oriented programming* e *functional programming*
- ☒ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- (c) *Feature-oriented programming* e *domain analysis*
- (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

2. Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

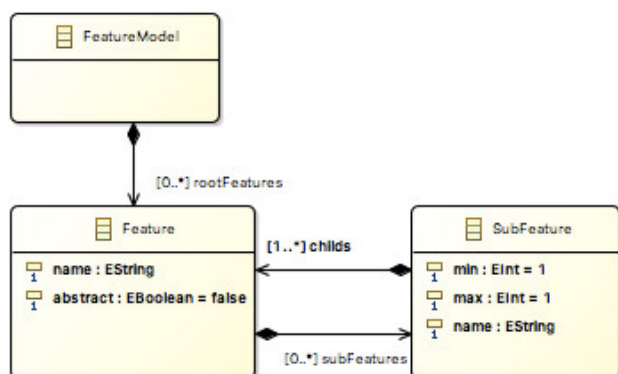
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
  
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
  
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```



```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Acceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14 [/if]
15
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20   aClass.concat('.java')]
21 /]
```

2. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7
8   [file (anEntity.name.classFileName(), false, 'UTF-8')]
9
10 public class [anEntity.name.toUpperFirst()/] {
11   [for (aField:Field | anEntity.field)]
12   String [aField.name/];
13   [/for]
14
15 }
16   [/file]
17
```

```
18 [/for]
19 [/template]
20
21 [query public classFileName(aClass: String): String =
22     aClass.concat('.java')
23 /]
```

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "**Parte 1**".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

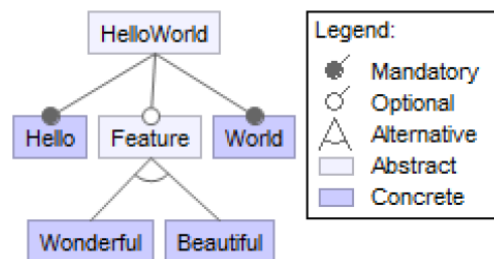
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- _____ Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- _____ O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- _____ Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- _____ Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- _____ As abordagens "Agile" e o MDSE não são compatíveis.
- _____ No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- _____ Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- _____ Os Value Objects não devem referenciar Entities.
- _____ A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- _____ O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- _____ A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- _____ Associações bidirecionais entre *entities* não são permitidas.
- _____ Todas as partes de um CIM devem ter uma correspondência na implementação.
- _____ Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- _____ A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- _____ O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em metamodelos ecore.
- _____ A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- _____ Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- _____ O MOF permite associações unárias, binárias e ternárias entre classes.
- _____ Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.

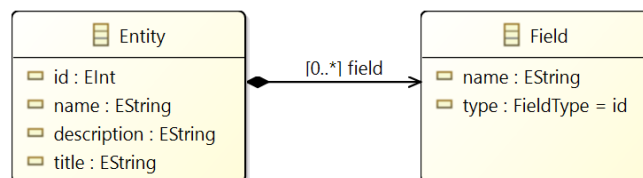


- O modelo apresentado...
 - permite representar no mínimo 2 programas e no máximo 4
 - permite representar no máximo 3 programas
 - permite representar no mínimo 3 programas
 - nenhuma das respostas anteriores
- Técnicas de implementação de SPL incluem...
 - Feature-oriented programming* e *functional programming*
 - Feature-oriented programming* e *Aspect-oriented programming*
 - Feature-oriented programming* e *domain analysis*
 - nenhuma das respostas anteriores
- Sobre a variabilidade numa linha de produtos...
 - não interessa para a engenharia de domínio
 - não é capturada num *feature model*
 - é capturada apenas com *features* marcadas como *optional*
 - nenhuma das respostas anteriores
- Uma abordagem SPL...
 - no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
 - no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
 - permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
 - nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

- Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.



Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.

- Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando).

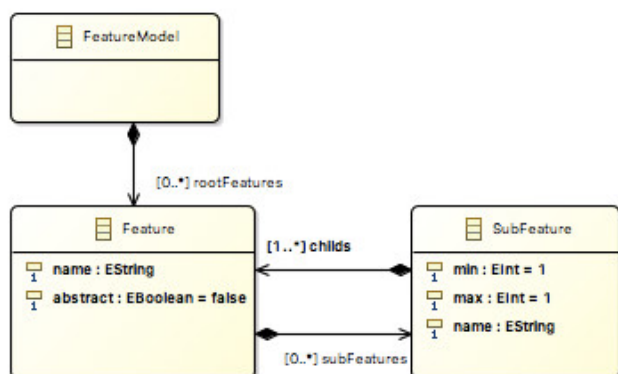
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
  
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```

1 SYNTAXDEF ofbiz
2 FOR <http://org/eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6     DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0') '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12     DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
        "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
13     Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
        | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
14     Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
        description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
15 }
  
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "  
2   entity: Entity {}  
3 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos **Entity** e **Field**:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;  
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'  
3 {  
4   class DocumentRoot {...}  
5   class Entity  
6   {  
7     attribute id : ecore::EInt[?];  
8     attribute name : String[?];  
9     attribute description : String[?];  
10    attribute title : String[?];  
11    property field : Field[*|1] { ordered composes };  
12    property primaryKey : Field[*|1] { ordered composes };  
13  }  
14  class Field  
15  {  
16    attribute name : String[?];  
17    attribute type : FieldType[?];  
18  }  
19  enum FieldType { serializable } {...}  
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]  
2 [module generate('http://org.eclipse/dsl/ofbiz')]  
3  
4 [template public generateElement(aDocumentRoot : DocumentRoot)]  
5 [comment @main/]  
6 [for(anEntity:Entity | aDocumentRoot.entity)]  
7   [file (anEntity.name.className(), false, 'UTF-8')]  
8 public class [anEntity.name.toUpperFirst()/] {  
9   [/file]  
10  
11 [/for]  
12 [/template]  
13  
14 [query public className(aClass: String): String =  
15   aClass.concat('.java')  
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.
2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

Answer Key for Exam F

= Parte 1 =

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 1".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 1 (4v)

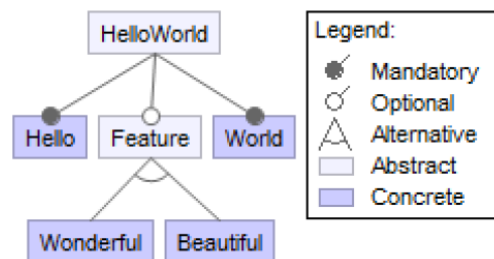
Assinale com V as afirmações verdadeiras e com F as falsas. **Cada resposta errada desconta meia certa.**

- False Numa abordagem MDSE é boa prática assumir que um metamodelo uma vez criado nunca mais será alterado.
- True O "root" é o único elemento num Aggregate que objetos exteriores podem manter referências para.
- False Os Factory são apropriados para criação de objectos mas não para a sua reconstituição.
- True Numa transformação ATL, numa regra "declarativa", pode-se usar uma expressão para condicionar os elementos do modelo origem que são usados na regra.
- False As abordagens "Agile" e o MDSE não são compatíveis.
- True No contexto do UML os diagramas de deployment enquadram-se na categoria de diagramas "class level" por oposição a "instance level"
- False Em MDSE os modelos são exclusivamente usados no contexto do "problem space" enquanto as linguagens de programação o são no "solution space".
- False Os Value Objects não devem referenciar Entities.
- True A "Reduction Feature" dos modelos significa que estes apenas refletem uma seleção (relevante) das propriedades dos original que estão a representar.
- True O mecanismo de Profile do UML, em lugar de introduzir novos elementos no metamodelo do UML permite, através dos estereótipos, a adição de semântica a elementos existentes.
- True A expressão OCL `Set{1, 2, 3, 4}->collect(x | x > 4)` tem como resultado `{false, false, false, false}`.
- False Associações bidirecionais entre *entities* não são permitidas.
- False Todas as partes de um CIM devem ter uma correspondência na implementação.
- True Nas abordagens MDSE existe uma clara separação entre o desenvolvimento de aplicações e infraestrutura.
- False A abordagem MDSE é conhecida pela "famosa" equação: Models + Metamodels = Software.
- True O EuGENia é uma ferramenta orientada ao desenvolvimento de sintaxes gráficas para DSLs baseadas em meta-modelos ecore.
- True A expressão OCL `Set{1, 2, 3, 4}->select(x | x > 3)` tem como resultado `{4}`.
- False Em UML "cardinality" refere-se ao intervalo possível para a "multiplicity" permitida nas instâncias do elemento em questão.
- False O MOF permite associações unárias, binárias e ternárias entre classes.
- False Em Acceleo, quando se configura uma "Run Configuration", é necessário especificar o metamodelo de input e o metamodelo de output.

Pergunta 2 (2,5v)

Nas questões seguintes selecione a opção que permite obter uma afirmação correta. **Uma resposta errada desconta meia certa.**

Considere o seguinte *feature diagram* que graficamente modela uma SPL de programas Hello World e no qual a *feature* HelloWorld contém duas *sub-features* obrigatórias e uma opcional. Note ainda que com um *alternative-group* uma das *sub-features* tem de ser seleccionada.



1. O modelo apresentado...

- (a) permite representar no mínimo 2 programas e no máximo 4
- ☒ (b) permite representar no máximo 3 programas
- (c) permite representar no mínimo 3 programas
- (d) nenhuma das respostas anteriores

2. Técnicas de implementação de SPL incluem...

- (a) *Feature-oriented programming* e *functional programming*
- ☒ (b) *Feature-oriented programming* e *Aspect-oriented programming*
- (c) *Feature-oriented programming* e *domain analysis*
- (d) nenhuma das respostas anteriores

3. Sobre a variabilidade numa linha de produtos...

- (a) não interessa para a engenharia de domínio
- (b) não é capturada num *feature model*
- (c) é capturada apenas com *features* marcadas como *optional*
- ☒ (d) nenhuma das respostas anteriores

4. Uma abordagem SPL...

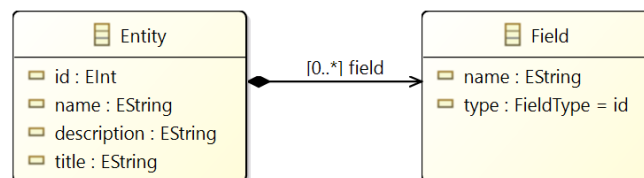
- ☒ (a) no processo *domain engineering* existe um conjunto de artefactos comuns para os sistemas a desenvolver
- (b) no processo *application engineering* não existe a necessidade de um conjunto de artefactos diferentes para cada sistema a desenvolver
- (c) permite em geral uma redução de custos desde que haja mais do que um sistema a desenvolver
- (d) nenhuma das respostas anteriores

Pergunta 3 (3,5v)

Apresente as respostas em folha anexa a entregar no final do exame.

1. Considere o seguinte diagrama representativo de um extrato de um ficheiro **ofbiz.ecore** semelhante ao que terá sido utilizado no projeto final da unidade curricular e que permitiria modelar vários elementos da plataforma Ofbiz.

Justifique que o elemento **Entity** tenha um atributo ID, mas não o elemento **Field**, com base em alguns padrões mencionados nas aulas, que devem ser definidos e descritos.



Resposta

Entity e ValueObjects. Do livro de Evans (2003): "An object that represents a descriptive aspect of the domain with no conceptual identity is called a VALUE OBJECT." e "objects are not fundamentally defined by their attributes, but rather by a thread of continuity and identity" são ENTITIES. No Ofbiz cada entidade deve ser distinguível das restantes e não com base nos seus atributos. No modelo apresentado elementos Field representam características de elementos Entity.

2. Indique qual o nível de modelação mais abstracto com MDA. Defina-o, descreva-o e indique um tipo de diagrama representativo de um modelo que possa ser classificado nesse nível (justificando). **Resposta** CIM (Computation Independent Model). Diagrama de casos de uso do UML.

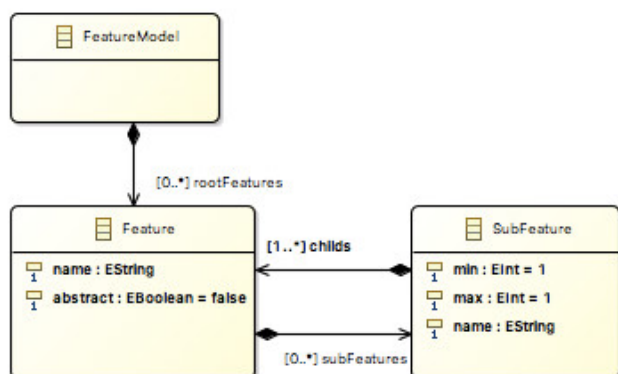
Nome: _____ Número: _____

Atenção:

1. Todas as justificações devem ser colocadas numa folha em branco identificada como "Parte 2".
2. Nas perguntas com justificação esta vale 50%.
3. Deve colocar o seu número, nome e versão do exame no topo da folha em branco.

Pergunta 4 (3v)

Considere o seguinte metamodelo ecore. Este metamodelo poderia ser usado como metamodelo para instâncias como a apresentada na pergunta 2. Considere ainda o seguinte extrato (incompleto) do metamodelo ecore em formato OCL. SubFeature permite representar diversos tipos de relações usadas nos diagramas de features. Por exemplo, para se representar uma relação **mandatory** usa-se **max** e **min** com valor 1. Para uma relação **optional** usamos **min**=0 e **max**=1.



```

1 import ecore :
    'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature =
    'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered
            composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        attribute abstract : Boolean[1];
13        ...
14    }
15    ...
16 }
    
```

1. Complete o metamodelo anterior (em OCL). Garanta ainda que **max** é sempre maior ou igual a **min**.

```

1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore#' ;
2
3 package feature : feature = 'http://www.isep.pt/feature'
4 {
5     class FeatureModel
6     {
7         property rootFeatures : Feature[*|1] { ordered composes };
8     }
9     class Feature
10    {
11        attribute name : String[1];
12        property subFeatures : SubFeature[*|1] { ordered composes };
13        attribute abstract : Boolean[1];
14    }
15    class SubFeature
16    {
17        attribute min : ecore::EInt[1] = '1';
18        attribute max : ecore::EInt[1] = '1';
19        property childs : Feature[+|1] { ordered composes };
20        attribute name : String[1];
21        invariant maxGTmin : self.max >= self.min;
22    }
23 }
    
```

Pergunta 5 (3v)

Considere o seguinte extrato de um ficheiro `ofbiz.cs` semelhante ao utilizado no projeto prático da unidade curricular.

```
1 SYNTAXDEF ofbiz
2 FOR <http://org.eclipse/dsl/ofbiz>
3 START DocumentRoot
4
5 TOKENS { ...
6   DEFINE FLOAT $('-')?((('1'..'9') ('0'..'9')* | '0')) '.' ('0'..'9')+ $;
7 }
8
9 TOKENSTYLES {...}
10
11 RULES {
12   DocumentRoot ::= "DocumentRoot" "{" ("title" ":" title['"', '"] | "description" ":" description['"', '"] |
13     "version" ":" version[FLOAT] | "copyright" ":" copyright['"', '"] | "entity" ":" entity)+ "}";
14   Entity ::= "Entity" "{" ("id" ":" id[] | "name" ":" name['"', '"] | "description" ":" description['"', '"]
15     | "title" ":" title['"', '"] | "field" ":" field | "primaryKey" ":" primaryKey)* "}";
16   Field ::= "Field" "{" ("name" ":" name['"', '"] | "type" ":" type[id:"id", idNe:"id-ne", idLong:"id-long",
17     description:"description", veryLong:"very-long", numeric:"numeric", dateTime:"date-time"])* "}";
18 }
```

1. Apresente em folha anexa uma possível continuação do código seguinte que permita representar receitas com os seus nomes e datas de lançamento e que deve obedecer à sintaxe concreta apresentada no extrato de código anterior.

```
1 DocumentRoot{title:"Ofbiz recipes" version:1.0 description:" "
2   entity: Entity {}
3 }
```

```
1 DocumentRoot{title:"My Ofbiz recipes" version:9.0 description:" "
2   entity: Entity {name:"Recipe"
3
4       field:Field {
5         name:"name"
6         type:description
7       }
8
9       field:Field {
10        name:"launchDate"
11        type:date-time
12      }
13    }
14 }
```

Pergunta 6 (4v)

Considere um metamodelo com elementos `Entity` e `Field`:

```
1 import ecore : 'http://www.eclipse.org/emf/2002/Ecore' ;
2 package ofbiz : _'org.eclipse.dsl.ofbiz' = 'http://org.eclipse/dsl/ofbiz'
3 {
4   class DocumentRoot {...}
5   class Entity
6   {
7     attribute id : ecore::EInt[?];
8     attribute name : String[?];
9     attribute description : String[?];
10    attribute title : String[?];
11    property field : Field[*|1] { ordered composes };
12    property primaryKey : Field[*|1] { ordered composes };
13  }
14  class Field
15  {
16    attribute name : String[?];
17    attribute type : FieldType[?];
18  }
```

```
19 enum FieldType { serializable } {...}
20 }
```

Considere o seguinte código Aceleo relativo a uma transformação do metamodelo em código fonte java:

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7   [file (anEntity.name.classFileName(), false, 'UTF-8')]
8 public class [anEntity.name.toUpperFirst()/] {}
9   [/file]
10
11 [/for]
12 [/template]
13
14 [query public classFileName(aClass: String): String =
15   aClass.concat('.java')]
16 /]
```

Responda às questões considerando que não há dependências entre as mesmas.

1. Altere a transformação de forma que, para cada classe, as suas variáveis de instância também sejam geradas a partir de elementos **Field** de cada **Entity**, devendo ser todas do tipo **String**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4 [template public generateElement(aDocumentRoot : DocumentRoot)]
5 [comment @main/]
6 [for(anEntity:Entity | aDocumentRoot.entity)]
7
8   [file (anEntity.name.classFileName(), false, 'UTF-8')]
9
10 public class [anEntity.name.toUpperFirst()/] {
11   [for (aField:Field | anEntity.field)]
12   String [aField.name/];
13   [/for]
14
15 }
16   [/file]
17
18 [/for]
19 [/template]
20
21 [query public classFileName(aClass: String): String =
22   aClass.concat('.java')]
23 /]
```

2. Altere a transformação de forma que sejam apenas geradas classes java (ficheiros .java) para os elementos **Entity** que tenham elementos **Field**.

```
1 [comment encoding = UTF-8 /]
2 [module generate('http://org/eclipse/dsl/ofbiz')]
3
4
5
6 [template public generateElement(aDocumentRoot : DocumentRoot)]
7 [comment @main/]
8 [for(anEntity:Entity | aDocumentRoot.entity)]
9   [if(anEntity.field -> size() > 0)]
10     [file (anEntity.name.classFileName(), false, 'UTF-8')]
11
12 public class [anEntity.name.toUpperFirst()/] {}
13   [/file]
14   [/if]
15
```

```
16 [/for]
17 [/template]
18
19 [query public classFileName(aClass: String): String =
20     aClass.concat('.java')
21 /]
```
