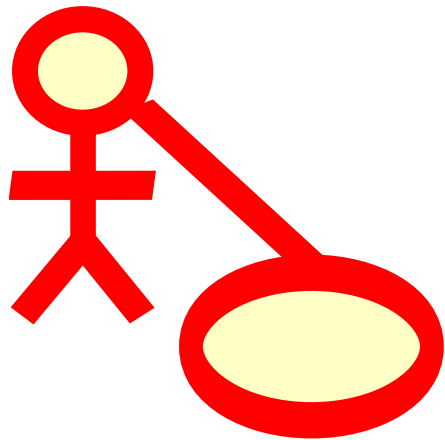


# El manual de l'Umbrello UML Modeller



## El manual de l'Umbrello UML Modeller

# Índex

<b>1</b>	<b>Introducció</b>	<b>8</b>
<b>2</b>	<b>Conceptes bàsics de l'UML</b>	<b>9</b>
2.1	Quant a UML . . . . .	9
2.2	Elements UML . . . . .	10
2.2.1	Diagrama de casos d'ús . . . . .	10
2.2.1.1	Cas d'ús . . . . .	10
2.2.1.2	Actor . . . . .	11
2.2.1.3	Descripció del cas d'ús . . . . .	11
2.2.2	Diagrama de classes . . . . .	11
2.2.2.1	Classe . . . . .	12
2.2.2.1.1	Atributs . . . . .	12
2.2.2.1.2	Operacions . . . . .	13
2.2.2.1.3	Plantilles . . . . .	13
2.2.2.2	Associacions de la classe . . . . .	13
2.2.2.2.1	Generalització . . . . .	13
2.2.2.2.2	Associacions . . . . .	13
2.2.2.2.3	Agregació . . . . .	14
2.2.2.2.4	Composició . . . . .	14
2.2.2.3	Altres elements del diagrama de classes . . . . .	14
2.2.2.3.1	Interfícies . . . . .	14
2.2.2.3.2	Tipus de dades . . . . .	14
2.2.2.3.3	Enumeracions . . . . .	15
2.2.2.3.4	Paquets . . . . .	15
2.2.3	Diagrames de seqüències . . . . .	15
2.2.4	Diagrames de col·laboració . . . . .	15
2.2.5	Diagrama d'estats . . . . .	16
2.2.5.1	Estat . . . . .	17
2.2.6	Diagrama d'activitats . . . . .	17
2.2.6.1	Activitat . . . . .	18
2.2.7	Elements d'ajuda . . . . .	18
2.2.8	Diagrames de components . . . . .	19

## El manual de l'Umbrello UML Modeller

2.2.9	Diagrames de desplegament . . . . .	19
2.2.10	Diagrames de relació d'entitats . . . . .	19
2.2.10.1	Entitat . . . . .	20
2.2.10.1.1	Atributs de l'entitat . . . . .	20
2.2.10.1.2	Restriccions . . . . .	21
2.2.11	Conceptes del diagrama de relació ampliada d'entitats (EER) . . . . .	21
2.2.11.1	Especialització . . . . .	21
2.2.11.1.1	Separat d'especialització . . . . .	21
2.2.11.1.2	Superposat d'especialització . . . . .	22
2.2.11.1.3	Categoria . . . . .	22
<b>3</b>	<b>Treballant amb l'Umbrello UML Modeller</b>	<b>24</b>
3.1	Interfície d'usuari . . . . .	24
3.1.1	Vista en arbre . . . . .	25
3.1.2	Finestra Documentació i Historial d'ordres . . . . .	25
3.1.3	Àrea de treball . . . . .	25
3.2	Creació, càrrega i desat dels models . . . . .	26
3.2.1	Model nou . . . . .	26
3.2.2	Desat del model . . . . .	26
3.2.3	Càrrega d'un model . . . . .	26
3.3	Edició dels models . . . . .	26
3.4	Afegir o eliminar diagrames . . . . .	27
3.4.1	Creació de diagrames . . . . .	27
3.4.2	Eliminar diagrames . . . . .	27
3.4.3	Reanomenar els diagrames . . . . .	27
3.5	Edició de diagrames . . . . .	27
3.5.1	Inserció d'elements . . . . .	28
3.5.2	Suprimir elements . . . . .	28
3.5.3	Edició dels elements . . . . .	28
3.5.4	Edició de les classes . . . . .	29
3.5.4.1	Arranjament general de la classe . . . . .	29
3.5.4.2	Arranjament dels atributs de la classe . . . . .	29
3.5.4.3	Arranjament de les operacions de la classe . . . . .	29
3.5.4.4	Arranjament de les plantilles de la classe . . . . .	29
3.5.4.5	Pàgina de les associacions de la classe . . . . .	29
3.5.4.6	Pàgina de visualització de la classe . . . . .	29
3.5.4.7	Pàgina estil de la classe . . . . .	30
3.5.5	Associacions . . . . .	30
3.5.5.1	Punts d'ancoratge . . . . .	30
3.5.6	Notes, text i quadres . . . . .	30
3.5.6.1	Àncores . . . . .	31

<b>4</b>	<b>Importació de codi i generació de codi</b>	<b>32</b>
4.1	Generació de codi . . . . .	32
4.1.1	Generant el codi . . . . .	32
4.1.1.1	Opcions de generació . . . . .	33
4.1.1.1.1	Detall en els comentaris . . . . .	33
4.1.1.1.2	Carpetes . . . . .	33
4.1.1.1.3	Política de sobreescritura . . . . .	34
4.1.1.1.4	Llenguatge . . . . .	34
4.1.1.2	Estat del progrés de la generació del codi . . . . .	34
4.2	Importació de codi . . . . .	34
<b>5</b>	<b>Altres característiques</b>	<b>36</b>
5.1	Altres característiques de l'Umbrello UML Modeller . . . . .	36
5.1.1	Còpia d'objectes com a imatges PNG . . . . .	36
5.1.2	Exportació a una imatge . . . . .	36
5.1.3	Impressió . . . . .	36
5.1.4	Carpetes lògiques . . . . .	37
<b>6</b>	<b>Configuració</b>	<b>38</b>
6.1	Configuració general . . . . .	38
6.1.1	Miscel·lània . . . . .	38
6.1.2	Desar automàticament . . . . .	39
6.1.3	Inici . . . . .	39
6.1.4	Notificacions . . . . .	39
6.2	Configuració del tipus de lletra . . . . .	39
6.3	Configuració de la interfície d'usuari . . . . .	40
6.3.1	General . . . . .	40
6.3.2	Associacions . . . . .	40
6.3.3	Color . . . . .	40
6.4	Configuració de les classes . . . . .	41
6.4.1	Mostra . . . . .	41
6.4.2	Àmbit inicial . . . . .	41
6.5	Configuració de l'importador de codi . . . . .	42
6.5.1	Camins de cerca d'inclusions . . . . .	42
6.5.2	Importació de C++ . . . . .	42
6.6	Configuració de la generació de codi . . . . .	43
6.6.1	Pestanya general de Configuració de la generació de codi . . . . .	43
6.6.1.1	Llenguatge . . . . .	43
6.6.1.2	Carpetes . . . . .	43
6.6.1.3	Política de sobreescritura . . . . .	43
6.6.2	Pestanya Format de la Configuració de la generació de codi . . . . .	44

## El manual de l'Umbrello UML Modeller

6.6.2.1	Detall en els comentaris . . . . .	44
6.6.2.2	Línies . . . . .	44
6.6.3	Opcions del llenguatge . . . . .	45
6.6.3.1	Generació de codi C++ . . . . .	45
6.6.3.1.1	Documentació . . . . .	45
6.6.3.1.2	General . . . . .	45
6.6.3.1.3	Generació del cos del mètode . . . . .	46
6.7	Configuració del visualitzador de codi . . . . .	47
6.8	Configuració de la disposició automàtica . . . . .	48
<b>7</b>	<b>Autors i història</b>	<b>49</b>
<b>8</b>	<b>Copyright</b>	<b>50</b>

## **Resum**

L'Umbrello UML Modeller ajuda al procés de desenvolupament de programari usant el llenguatge unificat de modelatge (UML) estàndard de la indústria que us permet crear diagrames per a dissenyar i documentar els vostres sistemes.

# Capítol 1

## Introducció

L'Umbrello UML Modeller és una eina de diagrames UML que us pot ajudar en el procés de desenvolupament de programari. Especialment durant les fases d'anàlisi i disseny d'aquest procés, l'Umbrello UML Modeller us ajuda a obtenir un producte d'alta qualitat. UML també es pot usar per a documentar els vostres dissenys de programari per a ajudar-vos a vós i als vostres companys desenvolupadors.

Tenir un bon model del vostre programari és la millor manera de comunicar-vos amb altres desenvolupadors que treballen en el projecte i amb els vostres clients. Un bon model és extremadament important per projectes de mitjana i gran envergadura, però també és molt útil pels petits. Encara que esteu treballant en un petit projecte d'una sola persona us beneficiareu d'un bon model perquè dona una vista general que pot ajudar a programar bé les coses a la primera.

UML és el llenguatge de diagrames usat per a descriure aquests models. Podeu representar les vostres idees en UML usant diferents tipus de diagrames. L'Umbrello UML Modeller 2.11 implementa els tipus següents:

- Diagrama de classes
- Diagrama de seqüències
- Diagrama de col·laboració
- Diagrama de casos d'ús
- Diagrama d'estats
- Diagrama d'activitats
- Diagrama de components
- Diagrama de desplegament
- Diagrama de relació d'entitats

Podeu trobar més informació quant a UML en el lloc web d'OMG, <http://www.omg.org> que va crear l'estàndard UML.

Desitgem que gaudiu de l'Umbrello UML Modeller i que us ajudi a crear programari d'alta qualitat. L'Umbrello UML Modeller és un programari lliure i disponible de forma gratuïta, l'única cosa que us demanem és que informeu de qualsevol error, problema o suggeriment als desenvolupadors de l'Umbrello UML Modeller en [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org) o <https://bugs.kde.org>.



## Capítol 2

# Conceptes bàsics de l'UML

### 2.1 Quant a UML

Aquest capítol us dona una vista general ràpida dels conceptes bàsics de UML. Tingueu en compte que aquesta no és una guia d'aprenentatge global de UML sinó més aviat una breu introducció a UML que es pot llegir com una guia d'aprenentatge de UML. Si voleu aprendre més quant al llenguatge de modelatge unificat, o en general quant a l'anàlisi i disseny de programari, adreceu-vos a un dels molts llibres disponibles quant al tema. Hi ha també una gran quantitat de guies d'aprenentatge a Internet que podeu prendre com a punt de partida.

El llenguatge unificat de modelatge (UML) és un llenguatge per a fer diagrames o notació per a especificar, visualitzar i modelar documents de sistemes de programari orientat a objectes. UML no és un mètode de desenvolupament, això significa que no us diu el que heu de fer primer i el que heu de fer a continuació o com heu de dissenyar el sistema, però us ajuda a visualitzar el vostre disseny i comunicar-vos amb els altres. UML és controlat pel grup Object Management (OMG) i és l'estàndard de la indústria per a descriure gràficament el programari.

UML està dissenyat pel disseny de programari orientat a objectes i té un ús limitat per a altres paradigmes de programació.

UML es compon de molts elements de model que representen les diferents parts d'un sistema de programari. Els elements UML s'usen per a crear diagrames, que representen una certa part, o un punt de vista del sistema. L'Umbrello UML Modeller implementa els tipus de diagrames següents:

- *Diagrames de casos d'ús* mostra els actors (persones o altres usuaris del sistema), els casos d'ús (els escenaris quan usen el sistema), i les seves relacions
- *Diagrames de classe* mostra les classes i les relacions entre elles
- *Diagrames de seqüència* mostra objectes i una seqüència de crides a mètodes que fan a altres objectes.
- *Diagrames de col·laboració* mostra objectes i les seves relacions, posant l'accent en els objectes que participen en l'intercanvi de missatges
- *Diagrames d'estat* mostra els estats, els canvis d'estat i els esdeveniments en un objecte o una part del sistema
- *Diagrames d'activitat* mostra les activitats i els canvis d'una activitat a una altra amb els esdeveniments que ocorren en alguna part del sistema
- *Diagrames de component* mostra els components de programació d'alt nivell (com ara les KParts o els components reutilitzables (beans) de Java).

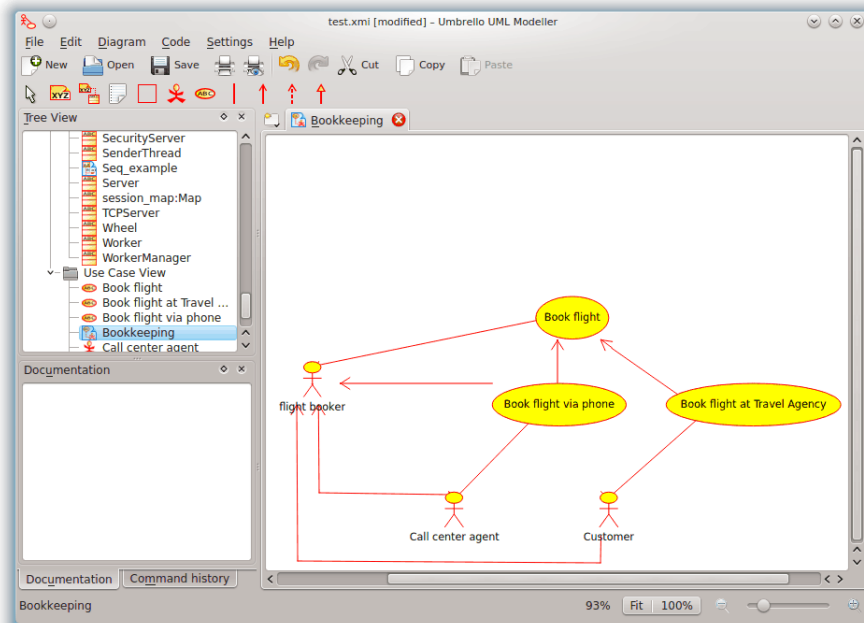
- *Diagrames d'implementació* mostra els casos dels components i les seves relacions.
- *Diagrames de relació d'entitats* mostren les dades i les relacions i les restriccions entre les dades.

## 2.2 Elements UML

### 2.2.1 Diagrama de casos d'ús

El diagrama de casos d'ús descriu les relacions i les dependències entre un grup de *casos d'ús* i els actors que participen en el procés.

És important adonar-se que els diagrames de casos no són adequats per a representar el disseny, i no poden descriure les interioritats d'un sistema. Els diagrames de casos d'ús estan destinats a facilitar la comunicació amb els futurs usuaris del sistema, i amb el client, i són especialment útils per a determinar les característiques necessàries que ha de tenir el sistema. Els diagrames de casos d'ús diuen, *què* ha de tenir el sistema però no — i no poden — especificar *com* aconseguir-ho.



L'Umbrello UML Modeller mostrant un diagrama de casos d'ús

#### 2.2.1.1 Cas d'ús

Un *cas d'ús* descriu — des del punt de vista dels actors — un grup d'activitats en un sistema que produeix un resultat concret i tangible.

Els casos d'ús són descripcions de les interaccions típiques entre els usuaris d'un sistema i el sistema en si mateix. Representen la interfície externa del sistema i especifica un forma dels requisits que ha de tenir el sistema (recordeu, només el que, no el com).

En treballar amb casos d'ús, és important recordar algunes regles simples:

- Cada cas d'ús és relatiu almenys a un actor

- Cada cas d'ús té un iniciador (és a dir un actor)
- Cada cas d'ús porta a un resultat pertinent (un resultat amb «valor comercial»)

Els casos d'ús també poden tenir relació amb altres casos d'ús. Els tres tipus de relacions més habituals entre casos d'ús són:

- «*inclou*» que especifica que un cas d'ús s'usa *dins* d'un altre cas d'ús
- «*amplia*» que especifica que en certes situacions, o en algun punt (anomenat punt d'ampliació) un cas d'ús serà ampliat per un altre.
- *Generalització* especifica que un cas d'ús hereta les característiques del cas d'ús 'superior', i pot substituir-ne una part o afegir-ne de noves de manera similar a l'herència entre classes.

### 2.2.1.2 Actor

Un actor és una entitat externa (fora del sistema) que interactua amb el sistema participant (i sovint iniciant) d'un cas d'ús. Els actors poden ser persones (per exemple, usuaris del sistema), altres sistemes informàtics o esdeveniments externs.

Els actors no representen les persones o sistemes *físicament*, però representen els seus *rols*. Això significa que quan una persona interactua amb el sistema de diverses maneres (assumint diferents rols) es representa amb diversos actors. Per exemple, una persona que dona atenció al client per telèfon i agafa ordres del client en el sistema serà representat per un actor «Personal d'atenció al client» i un actor «Representant de vendes»

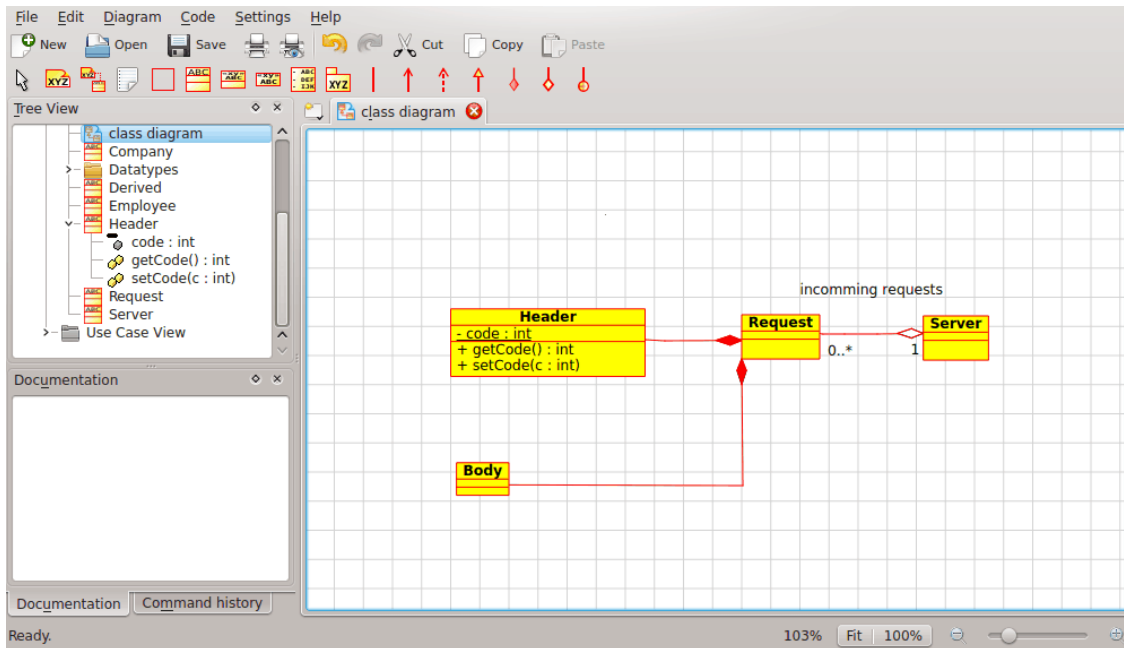
### 2.2.1.3 Descripció del cas d'ús

Les descripcions dels casos d'ús són narracions de text del cas d'ús. Normalment prenen la forma d'una nota o document enllaçat d'alguna manera al cas d'ús i expliquen els processos o activitats que es duen a terme en el cas d'ús.

## 2.2.2 Diagrama de classes

Els diagrames de classes mostren les diferents classes que componen el sistema i com es relacionen amb cada una de les altres. Els diagrames de classes es diu que són «estàtics» perquè mostren les classes, juntament amb els seus mètodes i atributs així com les relacions estàtiques entre elles: quines classes «coneixen» a altres classes o quines classes «són part» d'una altra classe, però no mostren les crides a mètodes entre elles.

## El manual de l'Umbrello UML Modeller

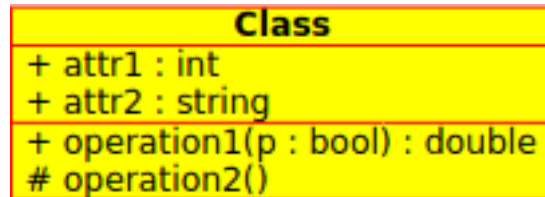


*L'Umbrello UML Modeller mostrant un diagrama de classes*

### 2.2.2.1 Classe

Una classe defineix els atributs i els mètodes d'un grup d'objectes. Tots els objectes d'aquesta classe (instàncies d'aquesta classe) comparteixen el mateix comportament, i tenen el mateix conjunt d'atributs (cada objecte té el seu propi conjunt). A vegades s'usa el terme «tipus» en lloc de classe, però és important mencionar que no són el mateix, i tipus és un terme més general.

En UML, les classes es representen amb rectangles, amb el nom de la classe i també poden mostrar els atributs i les operacions de la classe en dos altres «compartiments» dintre del rectangle.



*Representació visual d'una classe en UML*

#### 2.2.2.1.1 Atributs

En UML, els atributs es mostren amb almenys el seu nom, i també poden mostrar el seu tipus, el valor inicial i altres propietats. Els atributs també es poden mostrar amb la seva visibilitat:

- + Suport per atributs *publics*
- # Suport per atributs *protegits*
- - Suport per atributs *privats*

### 2.2.2.1.2 Operacions

Les operacions (mètodes) també es mostren amb almenys el seu nom, i també poden mostrar els seus paràmetres i els tipus de retorn. Les operacions poden, com els atributs, mostrar la seva visibilitat:

- + Suport per operacions *públiques*
- # Suport per operacions *protegides*
- - Suport per operacions *privades*

### 2.2.2.1.3 Plantilles

Les classes poden tenir plantilles, un valor que s'utilitza per a una classe no especificada o tipus. El tipus de plantilla s'especifica quan s'inicia una classe (és a dir s'ha creat un objecte). Les plantilles existeixen en el C++ modern i s'introduiran en Java 1.5 on s'anomenaran genèrics.

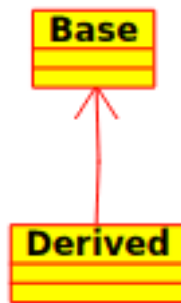
### 2.2.2.2 Associacions de la classe

Les classes es poden relacionar (associar-se) unes amb les altres de maneres diferents:

#### 2.2.2.2.1 Generalització

L'herència és un dels conceptes fonamentals de la programació orientada a objectes, en la que una classe «guanya» tots els atributs i operacions de la classe de la qual hereta, i pot substituir/-modificar alguna part d'aquesta, així com afegir més atributs i operacions a les seves pròpies.

En UML una *generalització* d'una associació entre dues classes les posa en una jerarquia que representa el concepte d'herència d'una classe derivada d'una classe base. En UML, les generalitzacions es representen per una línia connectant les dues classes, amb una fletxa del costat de la classe base.



*Representació visual d'una generalització en UML*

#### 2.2.2.2.2 Associacions

Una associació representa una relació entre classes, i dona la semàntica i estructura comunes per molts tipus de «connexions» entre objectes.

Les associacions són el mecanisme que permet als objectes comunicar-se entre ells, descriuen la connexió entre diferents classes (la connexió entre els objectes reals s'anomena connexió, o *enllaç*).

Les associacions poden tenir un rol que especifica l'ús de l'associació o pot ser uni- o bidireccional (indica que els dos objectes que participen en la relació poden enviar missatges l'un a l'altre, o si només un d'ells coneix de l'altre). Cada final de l'associació també conté un valor de multiplicitat,

que dicta quants objectes d'aquesta banda de l'associació es poden relacionar amb un objecte de l'altra banda.

En UML les associacions es representen com línies que connecten les classes que participen en la relació, i també poden mostrar el rol i la multiplicitat de cada un dels participants. La multiplicitat es mostra com a un interval [min..màx] de valors no negatius, amb una estrella (\*) en el costat del màxim es representa l'infinit.



Representació visual d'una associació en UML

### 2.2.2.2.3 Agregació

Les agregacions són un tipus d'associació especial en el que les dues classes que participen no tenen el mateix estat, però fan una relació «sencera-part». Una agregació descriu com la classe que pren el rol de sencera, està composta (té) altres classes, que prenen el rol de les parts. Per les agregacions, la classe que actua com a sencera sempre té la multiplicitat d'u.

En UML les agregacions són representades per una associació que mostra un rombe del costat de la sencera.

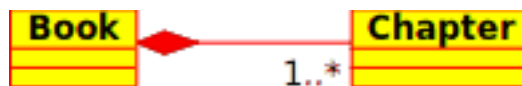


Representació visual d'una relació d'agregació en UML

### 2.2.2.2.4 Composició

Les composicions són associacions que representen agregacions *molt fortes*. Això vol dir que, les composicions també formen relacions sencera-part, però la relació és tan forta que les parts no poden existir per si mateixes. Només existeixen dintre de la sencera, i si la sencera es destrueix, també moren les parts.

En UML les composicions es representen per un rombe sòlid en el costat de la sencera.



### 2.2.2.3 Altres elements del diagrama de classes

Els diagrames de classes poden contenir altres elements a més de les classes.

#### 2.2.2.3.1 Interfícies

Les interfícies són classes abstractes, el que significa que les instàncies no poden ser directament creades per elles. Poden contenir operadors però no atributs. Les classes poden heretar de les interfícies (a través de la realització d'una associació) i llavors es poden fer les instàncies des d'aquestes classes.

#### 2.2.2.3.2 Tipus de dades

Els tipus de dades són primitives que normalment estan integrades en un llenguatge de programació. Els exemples comuns inclouen nombres enters i booleans. No poden relacionar-se amb les classes, però les classes poden relacionar-se amb ells.

### 2.2.2.3.3 Enumeracions

Les enumeracions són una simple llista de valors. Un exemple típic és una enumeració pels dies de la setmana. Les opcions d'una enumeració s'anomenen literal de l'enumeració. Com els tipus de dades no poden relacionar-se amb les classes per les classes poden relacionar-se amb elles.

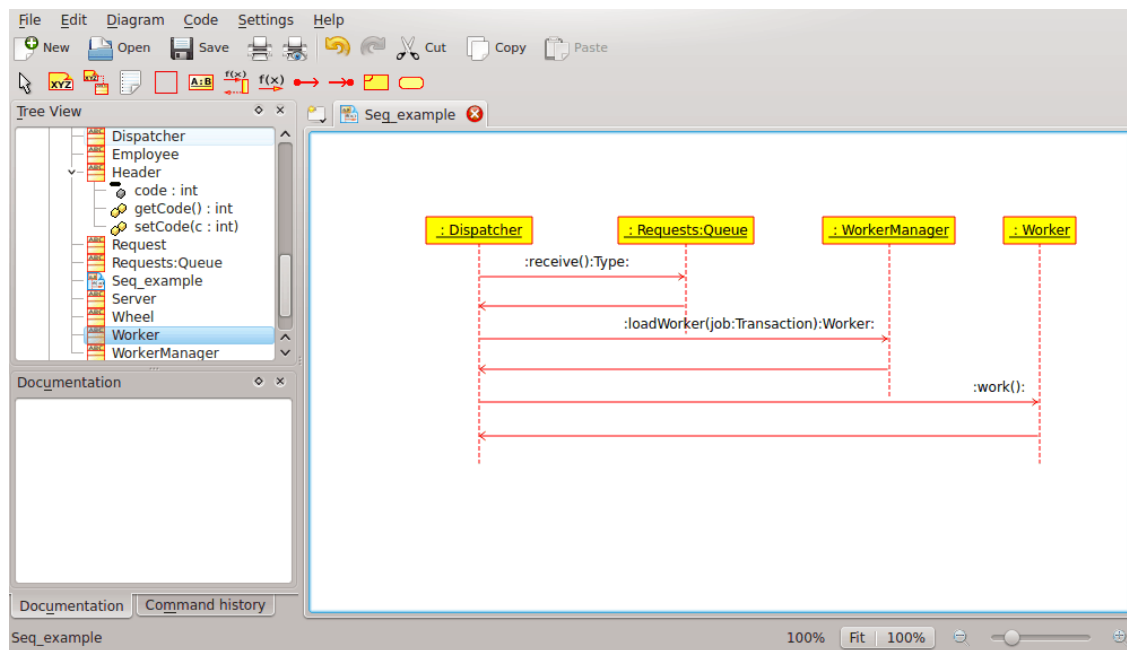
### 2.2.2.3.4 Paquets

Els paquets representen un espai de noms en un llenguatge de programació. En un diagrama s'usen per a representar parts d'un sistema que conté més d'una classe, potser centenars de classes.

## 2.2.3 Diagrames de seqüències

Els diagrames de seqüències mostren els missatges d'intercanvi (és a dir les crides a mètodes) entre diversos objectes en una situació específica en un temps delimitat. Els objectes són instàncies de les classes. Els diagrames de seqüències posen un èmfasi especial en l'ordre i els temps en què s'han enviat els missatges als objectes.

En els diagrames de seqüències es representen a través de línies de traços verticals amb el nom de l'objecte en la part superior. L'eix del temps també és vertical, augmentant a mesura que baixa, per què els missatges s'envien des d'un objecte a un altre en la forma de fletxes amb l'operació i el nom del paràmetre.



*L'Umbrello UML Modeller mostrant un diagrama de seqüències*

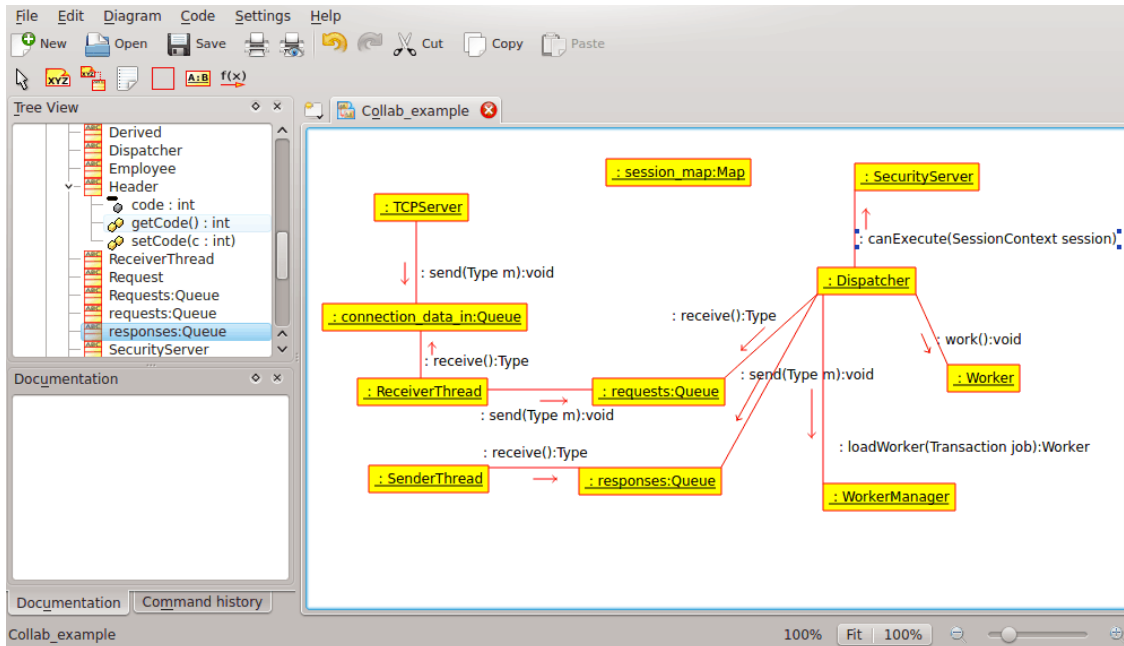
Els missatges poden ser o bé síncrons, el tipus normal de missatge de crida on es passa el control a l'objecte anomenat fins que el mètode s'hagi acabat d'executar, o asíncrons on es torna el control directament a l'objecte cridat. Els missatges síncrons tenen un quadre vertical al costat de l'objecte cridat per a mostrar el flux de control del programa.

## 2.2.4 Diagrames de col·laboració

Els diagrames de col·laboració mostren les interaccions que ocorren entre els objectes que participen d'una situació específica. Això és més o menys la mateixa informació mostrada pels diagrames de seqüències, però aquí l'èmfasi es posa en com ocorren les interaccions en el temps mentre

que en els diagrames de col·laboració es posen les relacions entre els objectes i la seva topologia en primer pla.

En un diagrama de col·laboració els missatges enviats des d'un objecte a un altre són representats per fletxes, mostrant el nom del missatge, els paràmetres, i la seqüència del missatge. Els diagrames de col·laboració són especialment útils per a mostrar el flux o situació d'un programa específic i és un dels millors tipus de diagrames per a demostrar o explicar ràpidament un procés en la lògica del programa.



L'Umbrello UML Modeller mostrant un diagrama de col·laboració

## 2.2.5 Diagrama d'estats

Els diagrames d'estats mostren els diferents estats d'un objecte durant la seva vida i els estímuls que causen que l'objecte canviï el seu estat.

Els diagrames d'estats visualitzen els objectes com a *màquines d'estat* o autòmats finits que poden ser en un grup d'estats finits i poden canviar el seu estat a través d'un grup finit d'estímuls. Per exemple, un objecte de tipus *ServidorXarxa* pot tenir un dels estats següents durant la seva vida:

- Preparat
- Escoltant
- Treballant
- Aturat

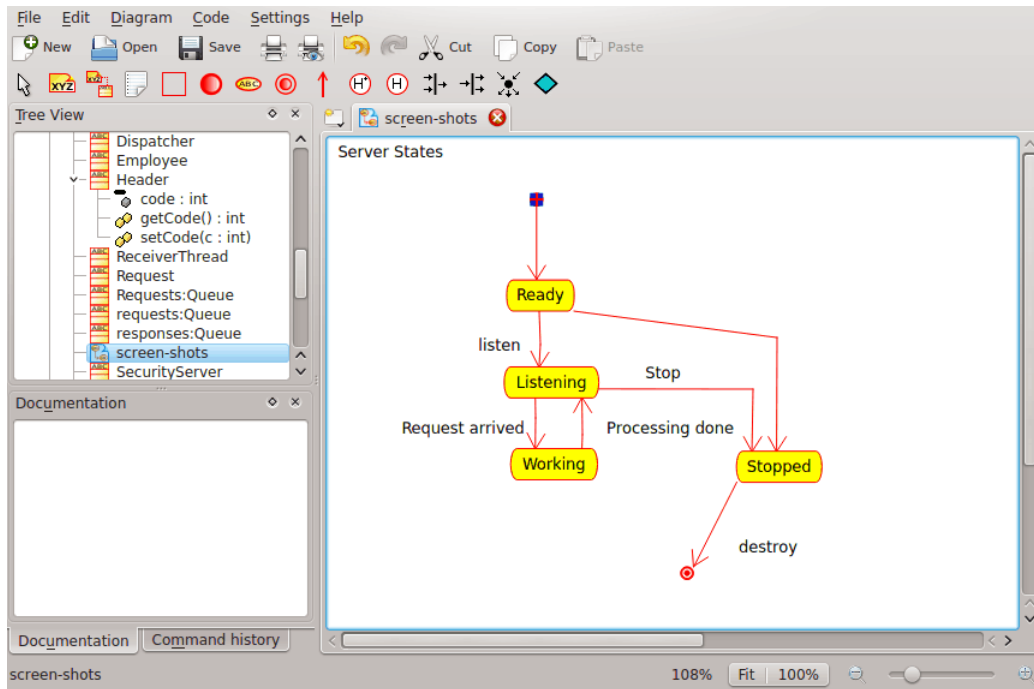
i els esdeveniments que poden causar que l'objecte canviï el seu estat són

- L'objecte és creat
- L'objecte rep un missatge d'escolta
- Un client sol·licita una connexió sobre la xarxa
- Un client finalitza una petició



## El manual de l'Umbrello UML Modeller

- La petició s'ha executat i finalitzat
- L'objecte rep un missatge d'aturada
- etc



*L'Umbrello UML Modeller mostrant un diagrama d'estats*

### 2.2.5.1 Estat

Els estats són els blocs de construcció dels diagrames d'estats. Un estat pertany exactament a una classe i representa un resum dels valors dels atributs que una classe pot prendre. Un estat UML descriu l'estat d'un objecte d'una classe particular

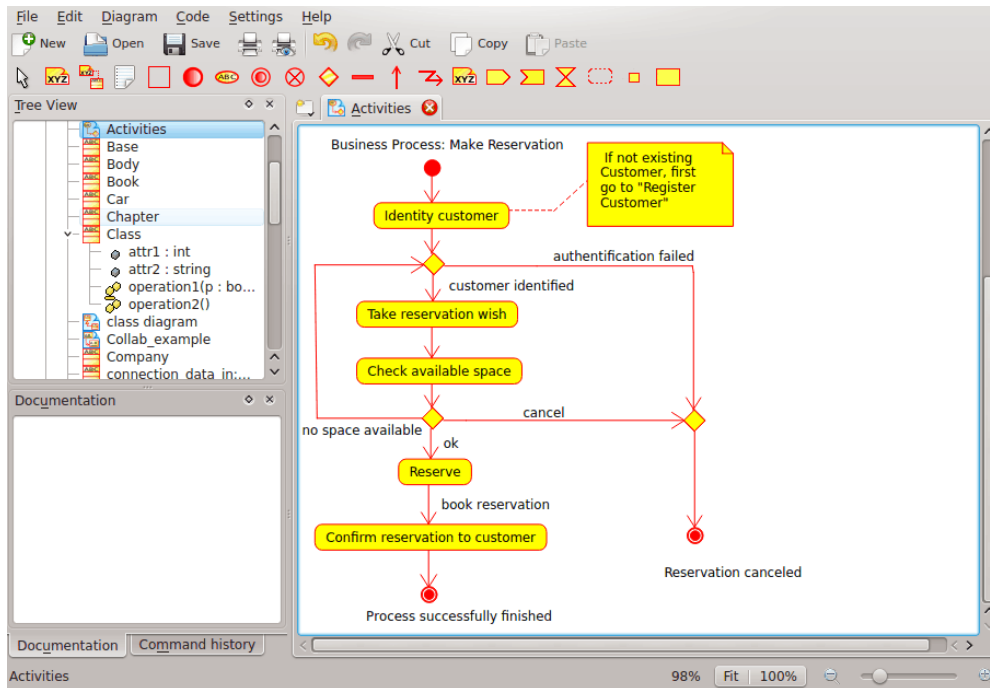
Tingueu en compte que no tots els canvis en un dels atributs d'un objecte han de ser representats per un estat, sinó únicament els canvis que poden afectar significativament el funcionament de l'objecte

Hi ha tipus especials d'estat: Inici i final. Són especials en què no hi ha cap esdeveniment que pugui fer que l'objecte retorni al seu estat d'inici, de la mateixa manera que no hi ha cap esdeveniment que pugui fer que un objecte surti del seu estat de final una vegada que l'ha assolit.

### 2.2.6 Diagrama d'activitats

Els diagrames d'activitats descriuen la seqüència d'activitats en un sistema amb l'ajuda d'activitats. Els diagrames d'activitats són una forma especial dels diagrames d'estats, que únicament (o principalment) contenen activitats.

## El manual de l'Umbrello UML Modeller



*L'Umbrello UML Modeller mostrant un diagrama d'activitats*

Els diagrames d'activitats són semblants als diagrames de flux de procediments, amb la diferència que totes les activitats estan clarament vinculades a objectes.

Els diagrames d'activitats sempre estan associats a una *classe*, una *operació* o un *cas d'ús*.

Els diagrames d'activitats implementen les activitats seqüencials així com les paral·leles. L'execució en paral·lel es representa amb icones forquilla/espera, i per les activitats executant-se en paral·lel no és important l'ordre en què es duen a terme (poden executar-se al mateix temps o una rere l'altra)

### 2.2.6.1 Activitat

Una activitat és un simple pas en un procés. Una activitat és un estat en el sistema amb activitat interna i, almenys, una transició sortint. Les activitats també poden tenir més d'una transició sortint, si tenen diferents condicions.

Les activitats poden formar jerarquies, això significa que una associació es pot compondre de diverses activitats «detallades», en aquest cas, les transicions sortints i entrants han de coincidir amb les transicions del diagrama detallat.

### 2.2.7 Elements d'ajuda

Hi ha uns pocs elements en UML que no tenen un valor semàntic real pel model, però ajuden a clarificar parts del diagrama. Els elements són

- Línies de text
- Notes de text i àncores
- Quadres

Les línies de text són útils per a afegir texts curts d'informació a un diagrama. És text lliure i no té significat per al propi model.

Les notes són útils per a afegir informació més detallada quant a un objecte o situació específica. Tenen un gran avantatge, ja que les notes es poden ancorar a elements UML per a mostrar que la nota «pertany» a un objecte o situació específica.

Els quadres són rectangles lliures que es poden usar per a agrupar elements per a fer els diagrames més llegibles. No tenen cap significat en el model.

### **2.2.8 Diagrames de components**

Els diagrames de components mostren els components de programari (ja sigui les tecnologies de components com les KParts, els components CORBA o els Beans de Java o simplement seccions del sistema que són clarament distingibles) i els estris que estan fets d'ells com ara fitxers de codi font, biblioteques de programació o les taules de les bases de dades relacionals.

Els components poden tenir interfícies (és a dir classes abstractes amb operacions) que permeten les associacions entre components.

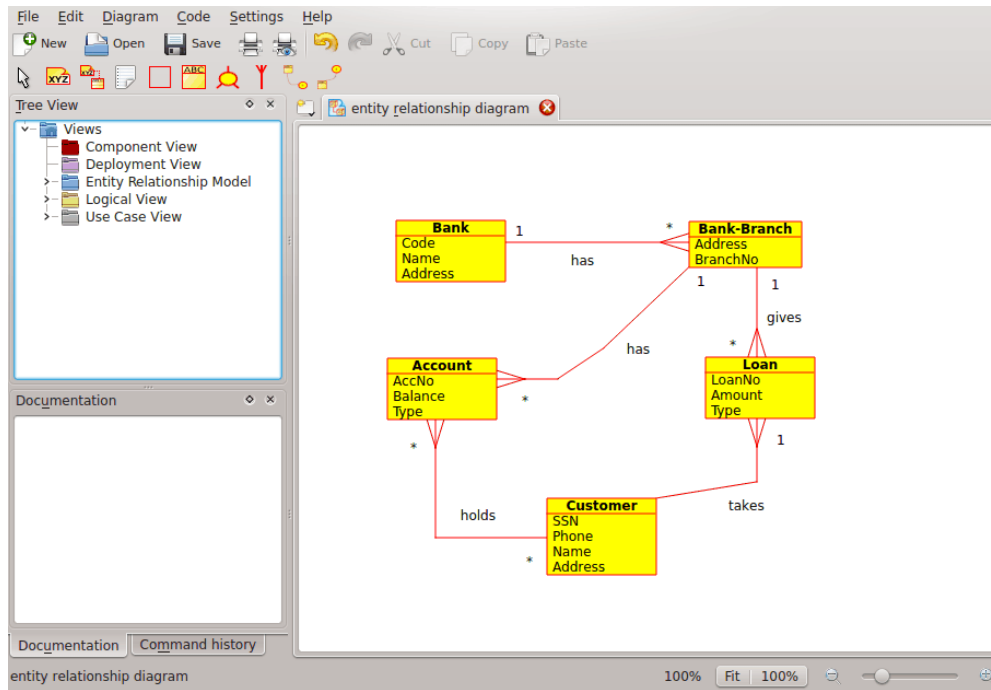
### **2.2.9 Diagrames de desplegament**

Els diagrames de desplegament mostren les instàncies de components en temps d'execució i les seves associacions. Inclouen els nodes que són els recursos físics, típicament un sol equip. També es mostren les interfícies i objectes (instàncies de classe).

### **2.2.10 Diagrames de relació d'entitats**

Els diagrames de relació d'entitats (diagrames ER) mostren el disseny conceptual d'aplicacions de base de dades. Representen les diferents entitats (conceptes) en el sistema d'informació i les relacions existents i les restriccions entre elles. Una extensió dels diagrames de relació d'entitats amb el nom «diagrama ampliat de relació reforçada» o «diagrama de relació d'entitats avançat» (EER), s'usen per a incorporar tècniques de disseny orientades a objectes en els diagrames ER.

## El manual de l'Umbrello UML Modeller



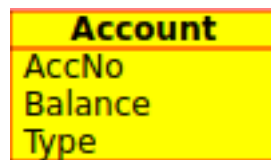
*Umbrello mostrant un diagrama de relació d'entitats*

### 2.2.10.1 Entitat

Una *entitat* és qualsevol concepte en el món real amb una existència independent. Poden ser un objecte amb existència física (per exemple un ordinador, robot) o poden ser un objecte amb una existència conceptual (p. ex. cursos de la universitat). Cada entitat té un conjunt d'atributs que descriuen les propietats de l'entitat.

*Nota:* No hi ha notacions estàndard per a representar diagrames ER. Diferents textos sobre aquest tema utilitzen diferents notacions. Els conceptes i notacions per diagrames EER usats en Umbrello són del llibre següent: *Elmasri R. i Navathe S. (2004). Fundamentals of Database Systems 4a edn. Addison Wesley*

En un diagrama ER, les entitats es representen amb rectangles, amb el nom de l'entitat en la part superior, i també poden mostrar els atributs de l'entitat en un altre «compartiment» dintre del rectangle.



*Representació visual d'una entitat en un diagrama ER*

#### 2.2.10.1.1 Atributs de l'entitat

En els diagrames ER, els atributs de les entitats es mostren amb el seu nom en un compartiment diferent de l'entitat a la qual pertanyen.

### 2.2.10.1.2 Restriccions

Les restriccions en els diagrames ER especifiquen les restriccions en les dades en l'esquema d'informació.

Hi ha quatre tipus de restriccions permeses en Umbrello:

- *Clau primària*: el grup d'atributs declarats com a *clau primària* són únics a l'entitat. Pot ser només una clau primària en una entitat i cap dels atributs que la constitueixen pot ser NULL.
- *Clau única*: els grups d'atributs declarats com a *únics* són únics a l'entitat. Hi pot haver moltes restriccions úniques a l'entitat. Els atributs que la componen poden ser NULL. Les claus úniques i les claus primàries només identifiquen una fila en una taula (entitat)
- *Clau externa*: una clau externa és una restricció referencial entre dues taules. La clau externa identifica una columna o un grup de columnes en una (referència) taula que es refereix a una columna o conjunt de columnes en una altra (referenciada) taula. Les columnes a la taula referenciada han de formar una clau primària o clau única.
- *Restricció de comprovació*: una restricció de comprovació (també coneguda com a restricció de comprovació de taula) és una condició que defineix les dades vàlides en afegir o actualitzar una entrada en una taula d'una base de dades relacional. Una restricció de comprovació s'aplica a cada fila de la taula. La restricció ha de ser un predicat. Pot referir-se a una o múltiples columnes de la taula.  
Exemple: preu >= 0

## 2.2.11 Conceptes del diagrama de relació ampliada d'entitats (EER)

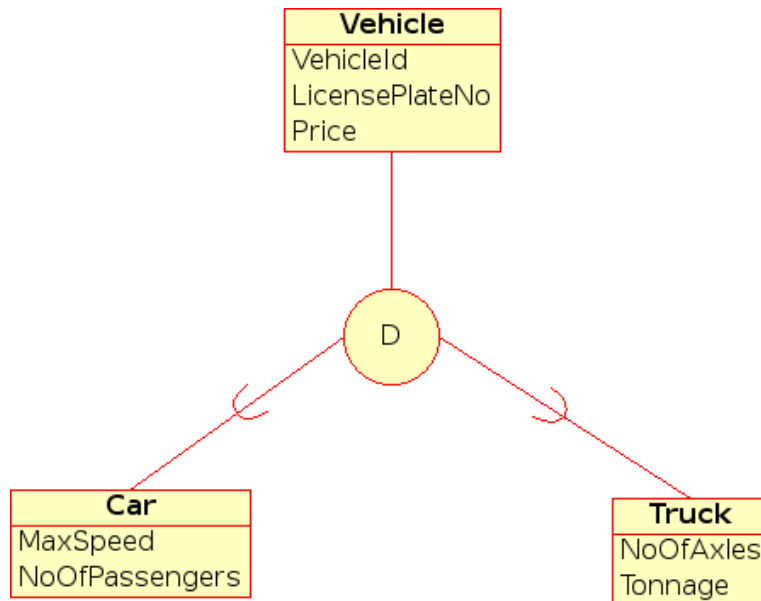
### 2.2.11.1 Especialització

L'especialització és una manera de formar entitats noves usant entitats que ja s'han definit. Les entitats noves, conegudes com a entitats derivades, prenen (o hereten) atributs de les entitats preexistents, que són referenciades com a entitats base. El seu objectiu és ajudar a la reutilització de les dades existents amb poca o cap modificació.

En Umbrello, es poden separar i superposar especialitzacions

#### 2.2.11.1.1 Separat d'especialització

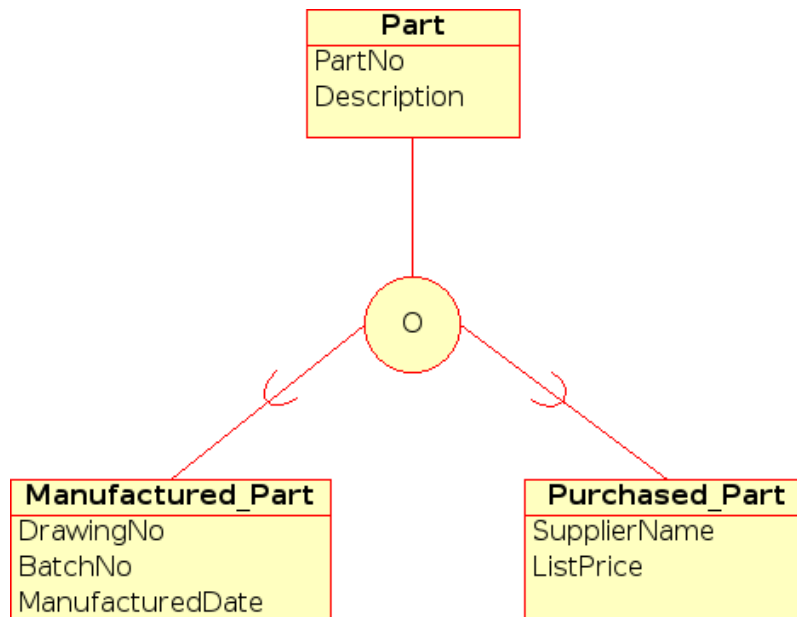
El separat d'especialitzacions especifica que les subclasses de l'especialització s'han de separar. Això vol dir que una entitat només pot ser membre d'una de les entitats derivades de l'especialització



Representació visual d'un separat d'especialització en un diagrama EER

### 2.2.11.1.2 Superposat d'especialització

Quan les entitats derivades no tenen restricció de separat, el seu conjunt d'entitats es diu que estan en l'especialització superposada. Això significa que la mateixa entitat del món real pot ser un membre de més d'una entitat derivada de l'especialització

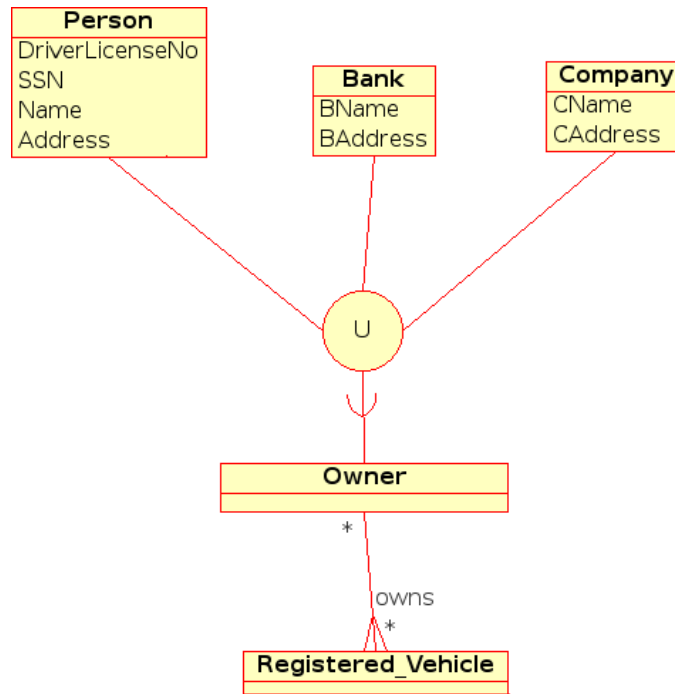


Representació visual d'una especialització superposada en un diagrama EER

### 2.2.11.1.3 Categoria

Una entitat derivada es diu que és una *categoria* que representa una col·lecció d'objectes que és un subconjunt de la unió de diferents tipus d'entitats. Una categoria és modelada quan sorgeix la

necessitat d'una única relació de superclasse/subclasse amb més d'una superclasse, on la superclasse representa diferents tipus d'entitats. (Com l'herència múltiple en la programació orientada a objectes).



Representació visual d'una categoria en un diagrama EER

## Capítol 3

# Treballant amb l'Umbrello UML Modeller

Aquest capítol us presentarà la interfície d'usuari de l'Umbrello UML Modeller i us explicarà tot el que necessiteu conèixer per a començar amb el modelatge. Totes les accions en l'Umbrello UML Modeller són accessibles a través del menú i les barres d'eines, però l'Umbrello UML Modeller també fa un ús intensiu dels menús contextuais del botó dret del ratolí. Podeu clicar amb el botó dret del ratolí en la majoria d'elements en l'àrea de treball de l'Umbrello UML Modeller o en la vista en arbre per a obtenir un menú amb les funcions més útils que poden ser aplicades a l'element particular sobre el qual esteu treballant. Alguns usuaris troben això una mica confús al començament perquè estan més familiaritzats a treballar amb el menú o les barres d'eines, però una vegada que us acostumeu a clicar amb el botó dret del ratolí accelerarà el vostre treball.

### 3.1 Interfície d'usuari

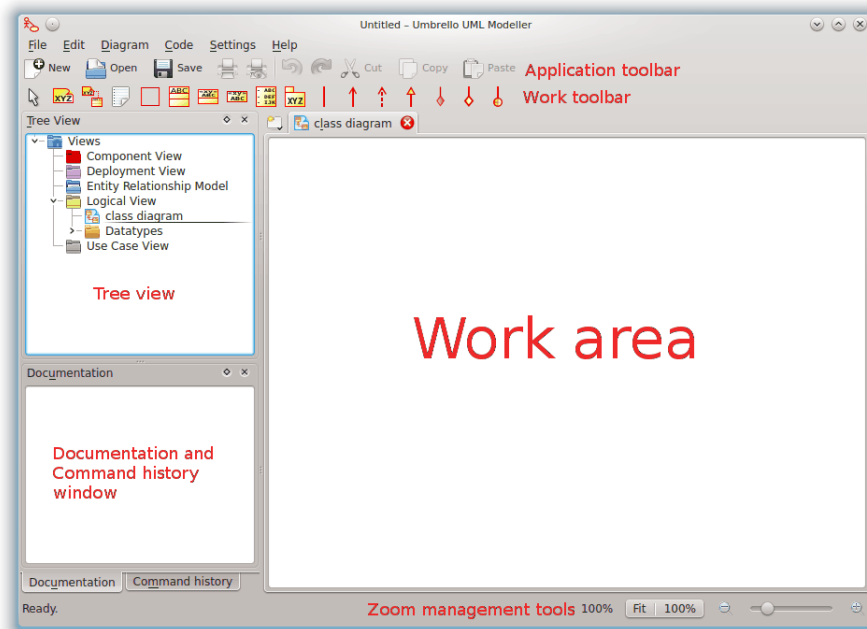
La finestra principal de l'Umbrello UML Modeller està dividida en tres àrees que us ajuden a mantenir una vista general del vostre sistema sencer i accedir als diferents diagrames de manera ràpida mentre trebal·leu en el vostre model.

Les tres àrees s'anomenen:

- Vista en arbre
- Àrea de treball
- Finestra Documentació i Historial d'ordres



## El manual de l'Umbrello UML Modeller



*La interfície d'usuari de l'Umbrello UML Modeller*

### 3.1.1 Vista en arbre

La vista en arbre està ubicada normalment en la part superior esquerra de la finestra i mostra la totalitat dels diagrames, classes, actors i els casos d'ús que construeixen el model. La vista en arbre us permet obtenir una vista general ràpida dels elements que componen el model. La vista en arbre també us dona un accés ràpid per a commutar entre els diferents diagrames del model i inserir elements del model en el diagrama actual.

Si esteu treballant en un model amb alguna cosa més que unes poques classes i diagrames la vista en arbre us pot ajudar a estar a sobre de les coses organitzant els elements del model en carpetes. Podeu crear carpetes seleccionant l'opció apropiada del menú contextual (clic amb el botó dret del ratolí en una de les carpetes de la vista en arbre) i podeu organitzar els elements movent-los a la carpeta apropiada (arrossegant i deixant anar)

### 3.1.2 Finestra Documentació i Historial d'ordres

Les finestres Documentació i Historial d'ordres són la petita finestra ubicada a la part inferior esquerra de l'Umbrello UML Modeller i us dona una vista prèvia ràpida de la documentació per l'element actualment seleccionat i l'historial d'ordres de la sessió de treball. La finestra de documentació és més aviat petita perquè està pensada simplement perquè pugueu donar una ullada a la documentació de l'element i al resum de l'historial d'ordres prenent l'espai mínim possible de la pantalla. Si necessiteu veure la documentació més detalladament, sempre podeu obrir les propietats de l'element.

### 3.1.3 Àrea de treball

L'àrea de treball és la finestra principal en l'Umbrello UML Modeller i és on es du a terme l'acció real. Useu l'àrea de treball per a editar i visualitzar els diagrames en el model. L'àrea de treball mostra el diagrama actiu actualment. Actualment només es pot visualitzar un diagrama alhora en l'àrea de treball.

## 3.2 Creació, càrrega i desat dels models

La primera cosa que necessiteu per a començar a fer alguna cosa útil amb l'Umbrello UML Modeller és crear un model on treballar. En iniciar l'Umbrello UML Modeller sempre carrega el darrer model usat o crea un model buit nou (depenent del vostre arranjament establert en el diàleg de configuració). Això us permetrà començar a treballar immediatament.

### 3.2.1 Model nou

Si necessiteu crear un model nou, ho podeu fer seleccionant l'entrada **Nou** del menú **Fitxer**, o clicant en la icona **Nou** de la barra d'eines de l'aplicació. Si actualment esteu treballant en un model que ha estat modificat l'Umbrello UML Modeller us demanarà si voleu desar els canvis abans de carregar el model nou.

### 3.2.2 Desat del model

Podeu desar el model en qualsevol moment seleccionant l'opció **Desa** del menú **Fitxer** o clicant el botó **Desa** de la barra d'eines de l'aplicació. Si necessiteu desar el model sota un nom diferent, podeu usar l'opció **Desa com a...** del menú **Fitxer**.

Per la vostra comoditat l'Umbrello UML Modeller també us ofereix l'opció de desar automàticament la feina cada cert període de temps. Podeu configurar si voleu aquesta opció així com els intervals de temps en **Arranjament** de l'Umbrello UML Modeller

### 3.2.3 Càrrega d'un model

Per a carregar un model que ja existeix podeu seleccionar l'opció **Obre** del menú **Fitxer** o clicar en la icona **Obre** de la barra d'eines de l'aplicació. Els models usats més recentment també són disponibles sota el submenú **Obre recent** en el menú **Fitxer** per a accelerar l'accés als models usats més freqüentment.

L'Umbrello UML Modeller només pot treballar en un model alhora, de manera que si demaneu al programa que carregui un model i el model actual ha estat modificat des de la darrera vegada que l'heu desat, l'Umbrello UML Modeller preguntarà si s'han de desar els canvis per a evitar qualsevol pèrdua de treball. També podeu iniciar dues o més instàncies de l'Umbrello UML Modeller alhora, de manera que pugueu copiar i enganxar entre ambdues instàncies.

## 3.3 Edició dels models

En l'Umbrello UML Modeller, hi ha bàsicament dues maneres d'editar els elements del model.

- Editar els elements del model directament a través de la vista en arbre
- Editar els elements a través d'un diagrama

Usant el menú contextual dels diferents elements en la vista en arbre podreu afegir, eliminar, i modificar la majoria dels elements del model. Clicant amb el botó Dret en les carpetes en la vista d'arbre se us donarà opcions per a crear els diferents tipus de diagrames així com, depenent de si la carpeta és una *Vista de casos d'ús* o una *Vista lògica*, els actors, els casos d'ús, les classes, etc.

Una vegada que heu afegit elements al model també podeu editar un element accedint al seu diàleg de propietats, que trobareu seleccionant l'opció *Propietats* del menú contextual mostrat en clicar amb el botó dret en els elements en la vista en arbre.

També podeu editar el model creant o modificant elements a través dels diagrames. Es donen més detalls de com fer això en les seccions següents.

## 3.4 Afegir o eliminar diagrames

El model UML consisteix en un conjunt d'elements UML i les associacions entre ells. No obstant això no podeu veure el model directament, heu d'usar els *Diagrames* per a veure'l.

### 3.4.1 Creació de diagrames

Per a crear un diagrama nou en el model simplement seleccioneu el tipus de diagrama que necessiteu des del submenú **Nou** en el menú **Diagrama** i donar-li un nom. El diagrama serà creat i activat, i podreu veure'l immediatament en la vista en arbre.

Recordeu que l'Umbrello UML Modeller fa extensiu l'ús dels menús contextuals: també podeu clicar amb el botó dret del ratolí en una carpeta en la vista en arbre i seleccionar el tipus de diagrama apropiat del submenú **Nou** en el menú contextual. Noteu que podeu crear diagrames de casos d'ús només en les carpetes de vista de casos d'ús, i els altres tipus de diagrames només es poden crear en les carpetes de vista lògica.

### 3.4.2 Eliminar diagrames

Si necessiteu eliminar un diagrama del model, podeu fer-ho activant-lo i seleccionant **Suprimeix** del menú **Diagrama**. També podeu aconseguir això seleccionant **Suprimeix** del menú contextual dels diagrames en la vista en arbre

Com que suprimir un diagrama és una cosa seriosa que pot causar la pèrdua de treball si es fa accidentalment, l'Umbrello UML Modeller us demanarà que confirmeu l'operació de supressió abans d'eliminar realment el diagrama. Una vegada que s'hagi suprimit un diagrama i s'hagi desat el fitxer no hi ha forma de desfer l'acció.

### 3.4.3 Reanomenar els diagrames

Si voleu canviar el nom d'un diagrama existent, podeu fer-ho fàcilment seleccionant l'opció Reanomena del seu menú botó dret del ratolí en la vista en arbre.

Una altra manera de reanomenar un diagrama és fer-ho a través del seu diàleg de propietats, que podeu obtenir seleccionant Propietats del seu menú contextual o fent-hi doble clic en la vista en arbre.

## 3.5 Edició de diagrames

En treballar amb un diagrama, l'Umbrello UML Modeller intentarà guiar-vos aplicant algunes regles simples sobre quins elements són vàlids en els diferents tipus de diagrames, així com les relacions que poden existir entre ells. Si sou un expert en UML probablement no ho notareu mai, però això ajuda als novicis de UML a crear diagrames d'acord amb l'estàndard.

Una vegada que heu creat els vostres diagrames és l'hora d'editar-los. Aquí cal fixar-se en la diferència (per als principiants subtil) entre l'edició del diagrama i l'edició del *model*. Com ja sabeu els diagrames són *vistes* del model. Per exemple, si creeu una classe editant el diagrama de classes, esteu realment editant ambdós, el diagrama i el model. Si canvieu el color o altres opcions de visualització d'una classe en el diagrama de classes, només esteu editant el diagrama, però no canvia res en el model.

### 3.5.1 Inserció d'elements

Una de les primeres coses que heu de fer en editar un diagrama nou és inserir elements (classes, actors, casos d'ús, etc.). Hi ha bàsicament dues maneres de per això:

- Arrossegar i deixar anar elements en el model des de la vista en arbre
- Crear elements nous en el model i afegir-los al diagrama alhora, usant una de les eines d'edició de la barra d'eines de treball

Per a inserir elements que ja existeixen en el model, simplement arrossegueu-los de la vista en arbre i deixeu-los anar on vulgueu en el diagrama. Sempre podeu moure elements en el diagrama usant l'eina de selecció

La segona manera d'afegir elements als diagrames és usant les eines d'edició de la barra d'eines de treball (noteu que això també afegeix els elements al model).

La barra d'eines de treball era ubicada per omissió a la part superior de la finestra. Les eines disponibles en aquesta barra d'eines (els botons que podeu veure) canvien depenent del tipus de diagrama en el qual esteu treballant actualment. El botó per l'eina actual està activat en la barra d'eines. Podeu canviar a l'eina de selecció prement la tecla **Esc**.

Quan hàgiu seleccionat una eina d'edició des de la barra d'eines de treball (per exemple, l'eina per a inserir classes) el punter del ratolí canvia a una creu, i podeu inserir elements en el model amb un clic simple en el diagrama. Noteu que els elements en UML han de tenir un *nom únic*. Per tant, si teniu una classe en un altre diagrama el nom de la qual és «ClasseA» i useu l'eina d'inserir classes en un altre diagrama no podeu usar aquest nom «ClasseA» per a la classe nova. Si se suposa que aquests són dos elements diferents, heu de donar-hi un nom únic. Si esteu intentant afegir el mateix element al diagrama, l'eina Insereix classe no és la correcta per a això. En lloc d'això heu d'arrossegar i deixar anar la classe des de la vista en arbre.

### 3.5.2 Suprimir elements

Podeu suprimir qualsevol element seleccionant l'opció **Suprimeix** del seu menú contextual.

De nou, hi ha una *gran* diferència entre eliminar un objecte d'un diagrama, i suprimir un objecte del model: si esborreu un objecte des d'un diagrama, només esteu eliminant l'objecte per aquest diagrama en particular: l'element encara serà part del model i si hi ha altres diagrames usant el mateix element, no patiran cap canvi. Si d'altra banda, esborreu l'element de la vista en arbre, l'esteu esborrant realment del *model*. Com que l'element ja no existeix en el model, serà eliminat automàticament de tots els diagrames en què aparegui.

### 3.5.3 Edició dels elements

Podeu editar la majoria dels elements UML en el model i els diagrames obrint el seu diàleg de propietats i seleccionant l'opció apropiada. Per a editar les propietats d'un objecte, seleccioneu **Propietats** del seu menú contextual (clic botó dret del ratolí). Cada element té un diàleg que consisteix en diverses pàgines on podeu configurar les opcions corresponents a aquest element. Per alguns elements, com ara els actors, només podeu establir un conjunt d'opcions, com ara el nom de l'objecte i la documentació, mentre que per altres elements, com ara les classes, podeu editar els atributs i les operacions, seleccioneu que voleu que es mostri en el diagrama (tota la signatura de l'operació o només els noms de les operacions, etc) i també els colors que voleu usar per a la línia i el farciment de la representació de la classe en el diagrama.

Pels elements UML també podeu obrir el diàleg de propietat fent-hi doble clic si esteu usant l'eina de selecció (la fletxa).

Noteu que també podeu seleccionar l'opció de propietats des del menú contextual dels elements en la vista en arbre. Això us permet també editar les propietats dels diagrames, com ara establir si s'ha de mostrar la graella o no.

### 3.5.4 Edició de les classes

Tot i que l'edició de les propietats de tots els objectes ja s'ha tractat en la secció anterior, reservem una secció especial a les classes perquè són una mica més complicades i tenen més opcions que la majoria dels altres elements UML.

En el diàleg de propietats d'una classe podeu establir-ho tot, des del color que usa a les operacions i atributs que té.

#### 3.5.4.1 Arranjament general de la classe

La pàgina d'opcions generals del diàleg de propietats s'explica per si mateixa. Aquí podeu canviar el nom de la classe, la visibilitat, la documentació, etc. Aquesta pàgina sempre està disponible.

#### 3.5.4.2 Arranjament dels atributs de la classe

En la pàgina d'arranjament dels atributs podeu afegir, editar o suprimir atributs (variables) de la classe. Podeu desplaçar els atributs amunt i avall de la llista prement els botons de fletxa del costat. Aquesta pàgina sempre està disponible.

#### 3.5.4.3 Arranjament de les operacions de la classe

De manera similar a la pàgina d'arranjament d'atributs, en la pàgina d'operacions podeu afegir, editar o eliminar operacions per a la classe. En afegir o editar una operació, introduïu les dades en el diàleg *Propietats de l'operació*. Si voleu afegir paràmetres a l'operació, heu de clicar el botó **Paràmetre nou...**, que mostrarà el diàleg *Propietats del paràmetre*. Aquesta pàgina sempre està disponible.

#### 3.5.4.4 Arranjament de les plantilles de la classe

Aquesta pàgina us permet afegir plantilles de classe que són les classes no especificades o tipus de dades. En Java 1.5 s'anomenen genèriques.

#### 3.5.4.5 Pàgina de les associacions de la classe

La pàgina **Associacions de la classe** mostra totes les associacions d'aquesta classe en el diagrama actual. Fent doble clic en una associació es mostren les seves propietats, i depenent del tipus de l'associació podeu modificar alguns paràmetres com ara establir la multiplicitat i el nom del rol. Si l'associació no permet que es modifiquin aquestes opcions, el diàleg de propietats de l'associació és de només lectura i només podeu modificar la documentació associada amb aquesta associació.

Aquesta pàgina només està disponible si obriu les propietats de la classe des d'un diagrama. Si seleccioneu les propietats de la classe des del menú contextual de la vista en arbre, aquesta pàgina no està disponible.

#### 3.5.4.6 Pàgina de visualització de la classe

En la pàgina **Opcions de visualització**, podeu establir que és el que es mostrarà en el diagrama. Una classe pot ser mostrada com a només un rectangle amb el nom de la classe (útil si teniu moltes classes en el diagrama, o de moment no esteu interessat en els detalls de cada classe) o com a una visualització completa mostrant els paquets, estereotips, i atributs i operacions amb signatura i visibilitat completes.

Depenent de la quantitat d'informació que voleu veure podeu seleccionar les opcions corresponents en aquesta pàgina. Els canvis que feu aquí només són *opcions de visualització* pel diagrama. Això significa que «ocultar» les operacions d'una classe només fa que no es mostrin en el diagrama, però l'operació encara és part del model. Aquesta opció només està disponible si seleccioneu les propietats de la classe des del diagrama. Si obriu les propietats de la classe des de la vista en arbre aquesta pàgina no hi és, ja que aquestes opcions de visualització no tenen sentit en aquest cas

#### 3.5.4.7 Pàgina estil de la classe

A la pàgina **Estil del giny** podeu configurar els colors que voleu per la línia i el farciment del giny. Aquesta opció òbviament només té sentit per les classes mostrades en els diagrames, i no hi és si obriu el diàleg de propietats de la classe des de la vista en arbre.

### 3.5.5 Associacions

Les associacions relacionen dos objectes UML l'un a l'altre. Normalment les associacions es defineixen entre dues classes, però també poden existir alguns tipus d'associacions entre casos d'ús i actors.

Per a crear una associació seleccioneu l'eina apropiada de la barra d'eines de treball (associació genèrica, generalització, agregació, etc.) i cliqueu en el primer element que participa de l'associació i després cliqueu en el segon element que participa. Noteu que aquests dos clics, un en cada un dels objectes que participen de l'associació, *no* és arrossegar un objecte i deixar-lo anar en l'altre.

Si intenteu usar una associació d'una manera contrària a l'especificació UML l'Umbrello UML Modeller rebutjarà crear l'associació i obtindreu un missatge d'error. Aquest és el cas, si per exemple, existeix una generalització des de la classe A a la classe B i intenteu crear una altra generalització des de la classe B a la classe A

Clicant amb el botó Dret en una associació es mostrarà un menú de context amb les accions que hi podeu aplicar. Si necessiteu suprimir una associació, simplement seleccioneu l'opció **Suprimeix** del seu menú contextual. També podeu seleccionar l'opció **Propietats** i, depenent del tipus d'associació, editar els atributs com ara els rols i les multiplicitats.

#### 3.5.5.1 Punts d'ancoratge

Les associacions es dibuixen, per omissió, com una línia fina que connecta els dos objectes en el diagrama.

Podeu afegir punts d'ancoratge per a doblegar una associació fent doble clic en algun punt de la línia de l'associació. Això inserirà un punt d'ancoratge (mostrat com un punt blau quan se selecciona la línia de l'associació) que podeu moure per a donar forma a l'associació

Si necessiteu eliminar un punt d'ancoratge, feu-hi un doble clic de nou per a eliminar-lo

Noteu que l'única manera d'editar les propietats d'una associació és a través del menú contextual. Si intenteu fer-hi doble clic com amb els altres objectes UML, això només inserirà un punt d'ancoratge.

### 3.5.6 Notes, text i quadres

Les notes, les línies de text i els quadres són elements que poden estar presents en qualsevol tipus de diagrama, però sense tenir un valor semàntic real, són molt útils per a afegir comentaris extres o explicacions que poden fer el diagrama més fàcil d'entendre.

Per a afegir una nota o una línia de text, seleccioneu l'eina corresponent de la barra d'eines de treball i cliqueu en el diagrama on voleu posar el comentari. Podeu editar el text obrint l'element a través del seu menú contextual o en el cas de les notes, també fent-hi doble clic.

### 3.5.6.1 Àncores

Les àncores s'usen per a enllaçar una nota de text i un altre element UML. Per exemple, normalment s'usa una nota de text per a explicar o fer algun comentari quant a una classe o una associació en particular, en aquest cas useu l'àncora per a deixar clar que la nota «pertany» a aquest element en particular.

Per a afegir una àncora entre una nota i un altre element UML, useu l'eina d'àncora de la barra d'eines de treball. Primer heu de clicar en la nota i després en l'element UML que voleu enllaçar a la nota.

## Capítol 4

# Importació de codi i generació de codi

L'Umbrello UML Modeller és una eina de modelatge UML, i la seva utilitat principal és ajudar-vos en *l'anàlisi i el disseny* dels vostres sistemes. No obstant això, per a fer la transició entre el disseny i la *implementació*, l'Umbrello UML Modeller us permet generar el codi font en diversos llenguatges de programació perquè pugueu començar. També, si voleu començar a usar UML en un projecte C++ ja iniciat, l'Umbrello UML Modeller us pot ajudar a crear un model en el vostre sistema des del codi font analitzant el codi font i important les classes trobades.

### 4.1 Generació de codi

L'Umbrello UML Modeller pot generar codi font per diversos llenguatges de programació basat en el vostre model UML per a ajudar-vos a començar amb la implementació del vostre projecte. El codi generat consisteix en la declaració de les classes, amb els seus mètodes i atributs, de manera que pugueu «emplenar els espais en blanc» proporcionant la funcionalitat de les operacions de les classes.

L'Umbrello UML Modeller 2 proporciona implementació de generació de codi per ActionScript, Ada, C++, C#, D, IDL, Java™, JavaScript, MySQL, Pascal, Perl, PHP, PHP5, PostgreSQL, Python, Ruby, Tcl, Vala, i XMLSchema.

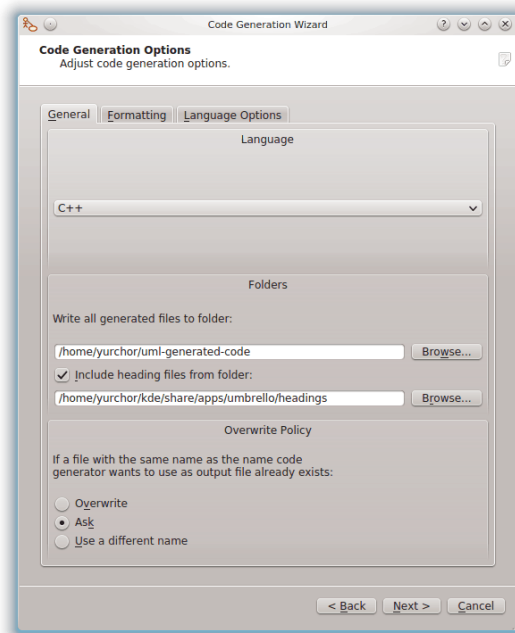
#### 4.1.1 Generant el codi

Per tal de poder generar codi amb l'Umbrello UML Modeller primer necessiteu crear o carregar un model que contingui almenys una classe. Quan esteu llest per a començar a escriure codi, seleccioneu l'entrada **Assistent de generació de codi...** del menú **Codi** per a iniciar un assistent que us guiarà a través del procés de generació de codi.

El primer pas és seleccionar les classes per les quals voleu generar el codi font. Per omisió totes les classes del model estan seleccionades, i podeu eliminar aquelles de les que no voleu generar el codi movent-les a la llista de l'esquerra.

El pas següent de l'assistent us permet modificar els paràmetres que el generador de codi usa mentre escriu el codi. Hi ha disponibles les opcions següents:





Opcions per a la generació de codi en l'Umbrello UML Modeller

#### 4.1.1.1 Opcions de generació

##### 4.1.1.1.1 Detall en els comentaris

L'opció **Escriure comentaris de documentació encara que estiguin buits** instrueix al generador de codi que escrigui els comentaris de l'estil `/** blah */` encara que els blocs de comentaris siguin buits. Si heu afegit documentació a les classes, mètodes o atributs en el model, el generador de codi escriurà aquests comentaris com documentació Doxygen independentment del que hàgiu establert aquí, però si seleccioneu aquesta opció, l'Umbrello UML Modeller escriurà blocs de comentaris per a totes les classes, mètodes i atributs encara que no hi hagi documentació en el model, en aquest cas haureu de documentar les classes més endavant directament en el codi font.

**Escriure comentaris per a les seccions encara que la secció estigui buida** fa que l'Umbrello UML Modeller escrigui comentaris en el codi font per a delimitar les diferents seccions d'una classe. Per exemple «mètodes públics» o «atributs» abans de les corresponents seccions. Si seleccioneu aquesta opció, l'Umbrello UML Modeller escriurà comentaris per a totes les seccions de la classe encara que la secció sigui buida. Per exemple, escriurà un comentari dient «mètodes protegits» encara que no hi hagi mètodes protegits en la classe.

##### 4.1.1.1.2 Carpetes

**Escriu tots els fitxers generats a una carpeta:** Aquí heu de seleccionar una carpeta on voleu que l'Umbrello UML Modeller desi el codi font generat.

L'opció **Inclou els fitxers de capçalera des de la carpeta** permet inserir una capçalera al començament de cada fitxer generat. Els fitxers de capçalera poden contenir informació del copyright o informació de la llicència i contenir variables que són avaluades en temps de generació. Podeu donar un cop d'ull a la plantilla de fitxers de capçalera empaquetada amb l'Umbrello UML Modeller per a veure com usar aquestes variables per a reemplaçar el vostre nom o la data actual en temps de generació.

#### 4.1.1.1.3 Política de sobreescritura

Aquesta opció li diu a l'Umbrello UML Modeller el que ha de fer si el fitxer que vol crear ja existeix en la carpeta de destinació. L'Umbrello UML Modeller *no pot modificar fitxers de codi font existents*, així que heu d'escollir entre sobre escriure el fitxer existent, ometre la generació d'aquest fitxer en particular o deixar que l'Umbrello UML Modeller usi un nom diferent, l'Umbrello UML Modeller afegirà un sufix al nom del fitxer.

#### 4.1.1.1.4 Llenguatge

L'Umbrello UML Modeller per omisió generarà el codi en el llenguatge que heu seleccionat com a llenguatge actiu, però amb l'assistent de generació de codi teniu l'opció de canviar aquest a un altre llenguatge.

#### 4.1.1.2 Estat del progrés de la generació del codi

El tercer i últim pas de l'assistent mostra l'estat del procés de la generació del codi. Només heu de clicar en el botó Genera perquè s'escriguin les classes.

Noteu que les opcions que heu seleccionat durant l'assistent de generació de codi només són vàlides per la generació actual. La pròxima vegada que executeu aquest assistent necessitareu tornar a seleccionar totes les opcions (la carpeta de capçaleres, la política de sobreescritura i tot això). Podeu establir les opcions predeterminades usades per l'Umbrello UML Modeller en la secció **Generació de codi** dels arranjaments de l'Umbrello UML Modeller, disponible en **Arranjament** → **Configura l'Umbrello UML Modeller...**

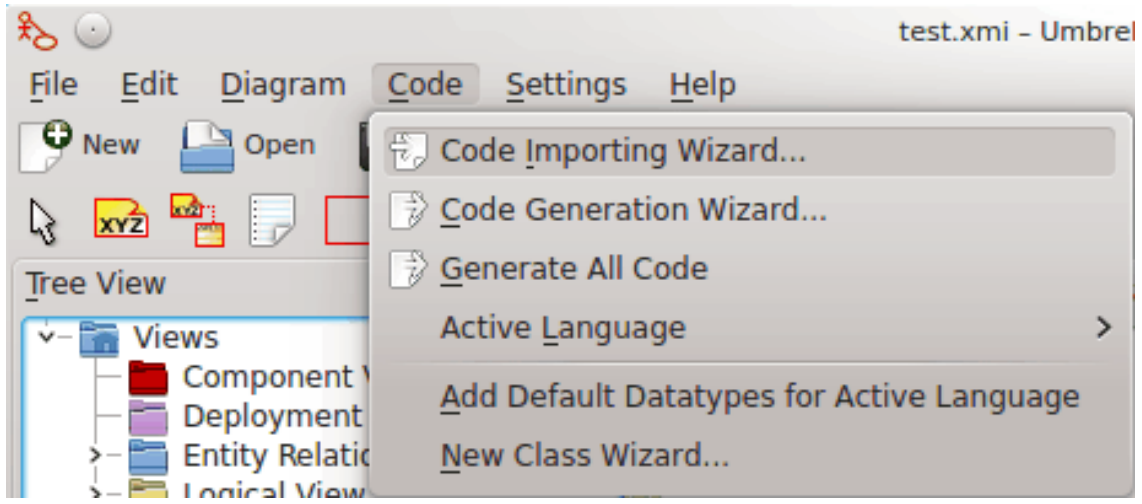
Si heu establert les opcions de generació de codi a les opcions correctes i voleu generar codi sense que calgui passar per l'assistent, podeu seleccionar **Genera tot el codi** complet del menú **Codi**. Això generarà codi per a totes les classes del model usant l'arranjament actual (incloent-hi la carpeta de sortida i la política de sobreescritura, per tant, l'heu d'usar amb cura).

## 4.2 Importació de codi

L'Umbrello UML Modeller pot importar el codi font dels vostres projectes existents per a ajudar-vos a construir models dels vostres sistemes. L'Umbrello UML Modeller 2 implementa el codi font ActionScript, Ada, C++, C#, D, IDL, Java™, Javascript, MySQL, Pascal, PHP i Vala.

Per a importar classes en el model, seleccioneu l'entrada **Assistent d'importació de codi...** del menú **Codi**. En el diàleg de fitxer seleccioneu els fitxers que contenen la declaració de classes i premeu **Següent >** després **Importació inicial** i **Finalitza**. S'importaran les classes i podreu trobar-les com a part del model en la vista en arbre. Noteu que l'Umbrello UML Modeller no crearà cap mena de diagrama per a mostrar les classes, només s'importaran en el model de manera que podeu usar-les més tard en qualsevol diagrama que vulgueu.

El manual de l'Umbrello UML Modeller



*Menú per a importar codi font en l'Umbrello UML Modeller*

## Capítol 5

# Altres característiques

### 5.1 Altres característiques de l'Umbrello UML Modeller

Aquest capítol explica breument algunes de les altres característiques que us ofereix l'Umbrello UML Modeller.

#### 5.1.1 Còpia d'objectes com a imatges PNG

A part d'oferir-vos la còpia normal, la funcionalitat de retallar i enganxar que podríeu esperar per a copiar objectes entre diferents diagrames, l'Umbrello UML Modeller pot copiar els objectes com a imatges PNG de manera que pugueu inserir-los en qualsevol altre tipus de document. No necessiteu fer res especial per a usar aquesta característica, simplement seleccioneu un objecte d'un diagrama (classe, actor, etc.) i copieu-lo (**Ctrl-C**, o usant el menú), llavors obriu un document Words del Calligra (o qualsevol programa en què pugueu enganxar imatges) i seleccioneu **Enganxa**. Aquesta és una gran característica per a exportar parts del diagrama com a imatges senzilles.

#### 5.1.2 Exportació a una imatge

També podeu exportar un diagrama complet com una imatge. L'única cosa que necessiteu fer és seleccionar el diagrama que voleu exportar, i a continuació l'opció del menú **Diagrama, Exporta com a imatge...**

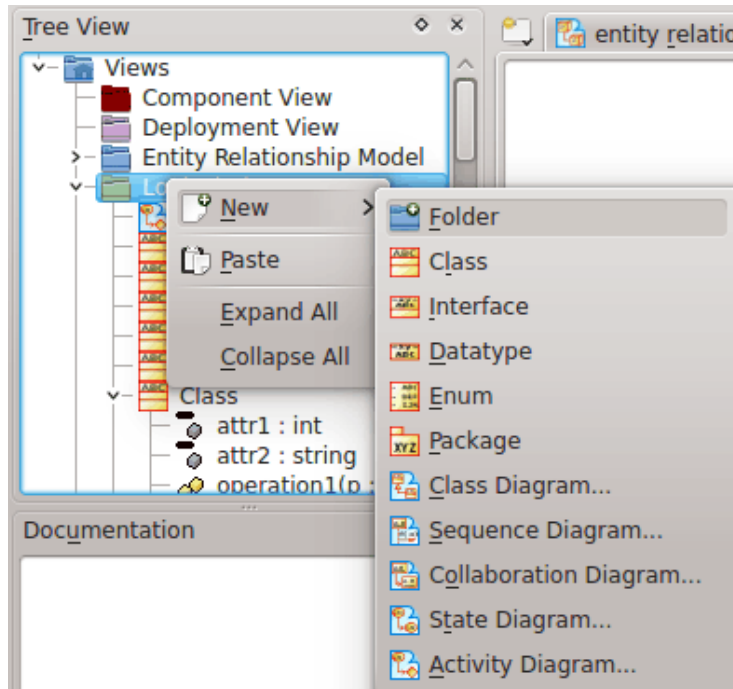
Podeu exportar diversos diagrames alhora utilitzant l'opció **Exporta els diagrames com a imatges...** des del menú **Fitxer**. Amb això, també es pot establir la resolució de la imatge, perquè les imatges no siguin tan borroses.

#### 5.1.3 Impressió

L'Umbrello UML Modeller permet imprimir diagrames individuals. Premeu el botó **Imprimeix** de la barra d'eines de l'aplicació o seleccioneu l'opció **Imprimeix** del menú **Fitxer** i obtindreu un diàleg d'impressió estàndard del KDE des d'on podreu imprimir els vostres diagrames.

### 5.1.4 Carpets lògiques

Per a organitzar millor el model, especialment per a projectes grans, podeu crear carpetes lògiques en la vista en arbre. Simplement seleccioneu l'opció **Nou** → **Carpeta** del menú contextual de les carpetes predeterminades de la vista en arbre per a crear-la. Les carpetes poden estar imbricades, i podeu moure objectes arrossegant-los i deixant-los anar des d'una carpeta a una altra.

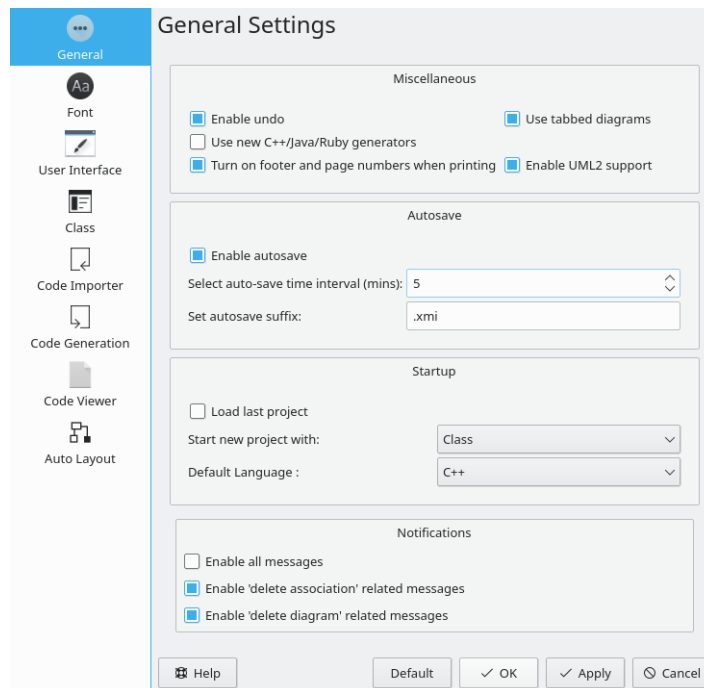


*Organització del model amb carpetes lògiques en l'Umbrello UML Modeller*

## Capítol 6

# Configuració

### 6.1 Configuració general



Opcions de la Configuració general a l'Umbrello UML Modeller

#### 6.1.1 Miscel·lània

- L'opció **Habilita desfer** permet desfer una acció prèvia.
- **Usa els generadors nous de C++/Java/Ruby** permet a l'usuari seleccionar els generadors de codi antic o nou
- Quan se selecciona **Activa el peu i els números de pàgina en imprimir**, imprimeix la informació i el número de pàgina del diagrama per al diagrama que s'imprimeix.
- **Usa diagrames amb pestanyes** ofereix l'opció de tenir múltiples finestres de diagrama en pestanyes obertes alhora.

### 6.1.2 Desar automàticament

- **Habilita el desament automàtic** dona l'opció de desar automàticament el fitxer.
- **Selecciona l'interval de temps del desament automàtic (minuts):** permet establir el temps abans que es desi automàticament el fitxer.
- **Estableix el sufix de desament automàtic:** el valor predeterminat és `.xmi` però permet establir una extensió de fitxer diferent.

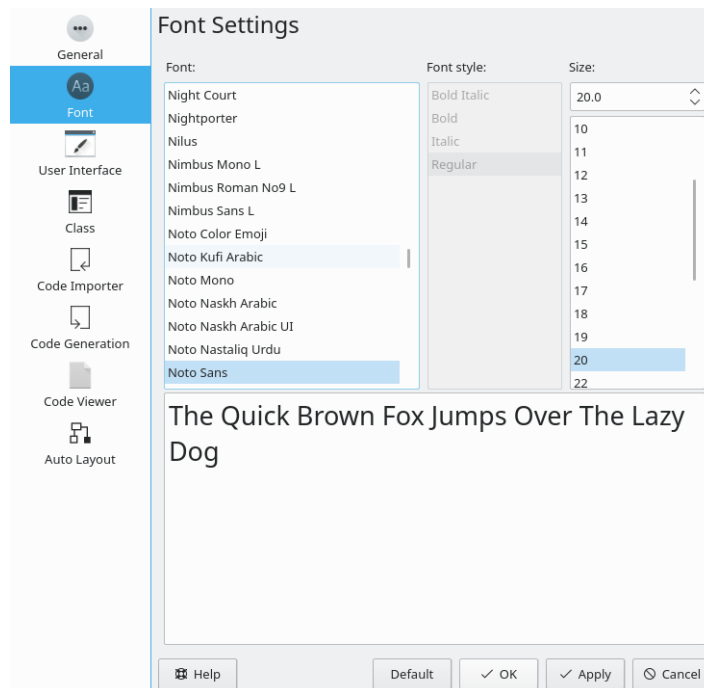
### 6.1.3 Inici

- Si s'estableix **Carrega l'últim projecte**, sempre es carregarà l'últim projecte de treball en iniciar el programa.
- **Comença el projecte nou amb:** permet triar amb quin tipus de diagrama UML començar en un projecte nou.
- **Idioma predeterminat:** és una opció per al llenguatge de programació que s'emprarà.

### 6.1.4 Notificacions

- **Activa tots els missatges** és una opció per a veure totes les notificacions o un conjunt reduït de notificacions.
- Si està marcada **Activa els missatges relacionats amb «suprimeix associació»**, assegura que rebreu tots els missatges d'aquest tipus.
- Si està marcada **Activa els missatges relacionats amb «suprimeix diagrama»**, assegura que rebreu tots els missatges d'aquest tipus.

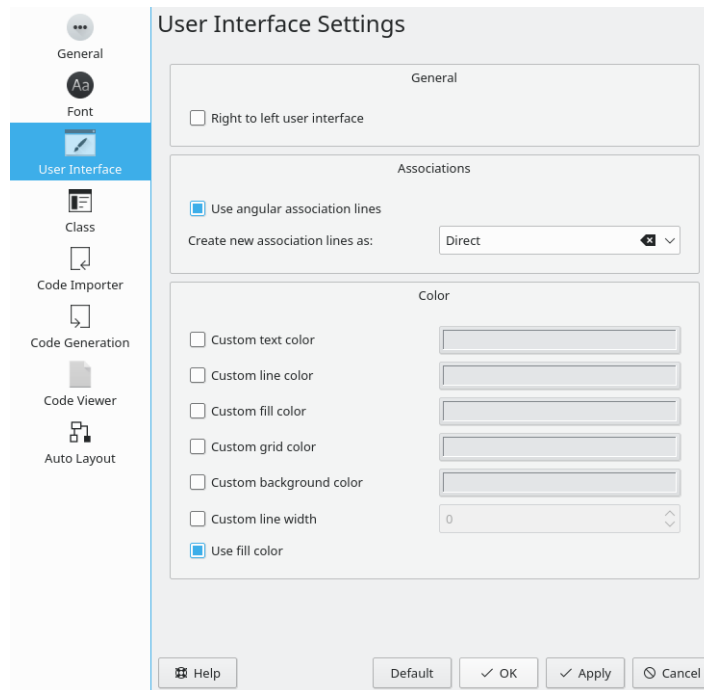
## 6.2 Configuració del tipus de lletra



Opcions de la Configuració del tipus de lletra del diagrama a l'Umbrello UML Modeller

Aquesta configuració del tipus de lletra estableix les característiques del text en els diagrames. Els tipus de lletra i la mida són les úniques opcions seleccionables.

## 6.3 Configuració de la interfície d'usuari



*Opcions de la Configuració de la interfície d'usuari a l'Umbrello UML Modeller*

### 6.3.1 General

**Interfície d'usuari de dreta a esquerra** configura la interfície per als idiomes de dreta a esquerra.

### 6.3.2 Associacions

**Usa línies d'associació angulars** permet que les línies d'associació variïn en qualsevol angle.

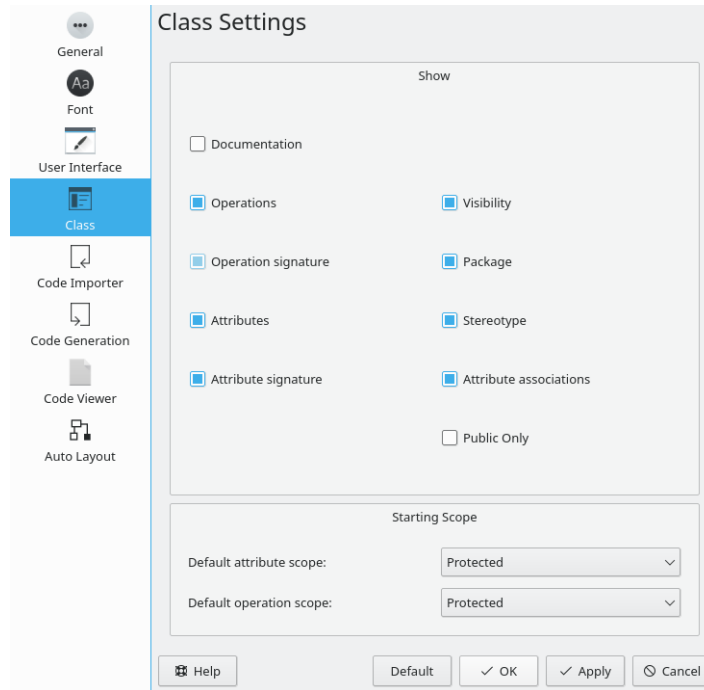
**Crea línies d'associació noves com a:** dona la possibilitat de canviar l'estil de línia de l'associació.

### 6.3.3 Color

La secció de color ofereix diverses opcions per a canviar els colors del text, línia, emplenat, quadrícula i fons, així com l'amplada de la línia.



## 6.4 Configuració de les classes



*Opcions de la Configuració de les classes a l'Umbrello UML Modeller*

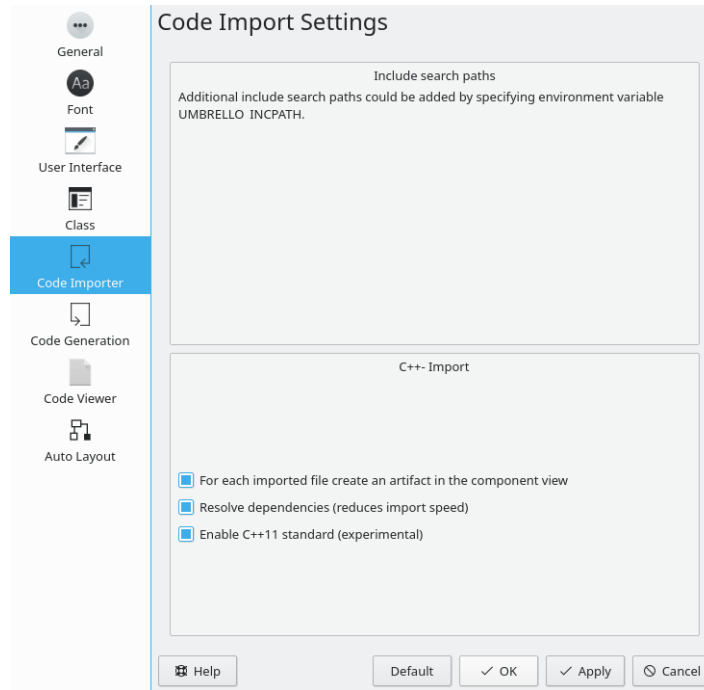
### 6.4.1 Mostra

La secció Mostra té nombroses opcions que determinen quines característiques de classe es mostraran en el diagrama de classes.

### 6.4.2 Àmbit inicial

Opcions per a les opcions predeterminades d'atribut i operació; pública, privada o protegida.

## 6.5 Configuració de l'importador de codi



*Opcions de la Configuració de l'importador de codi a l'Umbrello UML Modeller*

### 6.5.1 Camins de cerca d'inclusions

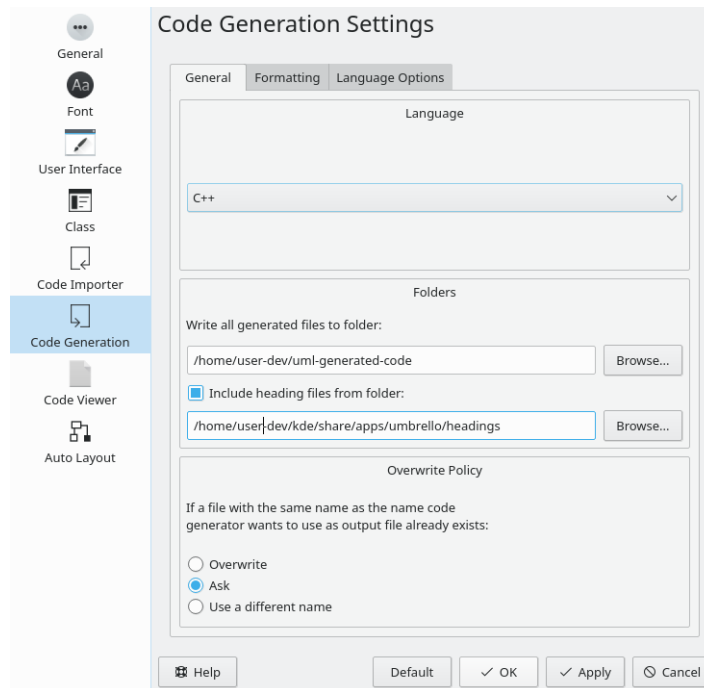
Es dona una recomanació general per a millorar la cerca en incloure UMBRELLO INCPATH com a una variable d'entorn.

### 6.5.2 Importació de C++

- **Crea un artefacte a la vista de components per a cada fitxer importat:** l'artefacte creat es podrà arrossegar a la vista Diagrama de classes on les dependències es podran veure amb facilitat juntament amb els atributs i funcions de cada fitxer.
- **Resol les dependències (redueix la velocitat d'importació):** assegura que es resolguin totes les dependències de fitxer, el qual es mostrarà a les dependències de classe en el Diagrama de classes.
- **Activa el C++11 estàndard (experimental):** Una característica experimental per a complir amb C++11, inhabilitar si no és necessària.

## 6.6 Configuració de la generació de codi

### 6.6.1 Pestanya general de Configuració de la generació de codi



#### *Opcions de la Configuració de la generació de codi - General a l'Umbrello UML Modeller*

El modelador UML Umbrello UML Modeller pot generar codi font per diversos llenguatges de programació basat en el vostre model d'UML per a ajudar-vos a començar amb la implementació del vostre projecte. El codi generat consisteix en la declaració de les classes, amb els seus mètodes i atributs, de manera que pugueu «emplenar els espais en blanc» proporcionant la funcionalitat de les operacions de les classes.

#### 6.6.1.1 Llenguatge

Trieu el llenguatge de programació per a emprar en els projectes. Les opcions que s'ofereixen són ActionScript, Ada, C++, C #, D, IDL, Java, JavaScript, MYSQL, Pascal, Perl, PHP, PHP5, PostgreSQL, Python, Ruby, SQL, Tcl, Vala i XMLSchema

#### 6.6.1.2 Carpetes

**Escriu tots els fitxers generats a una carpeta:** té un camp editable per al camí desitjat per als fitxers generats o de manera opcional un botó d'exploració per a seleccionar el camí.

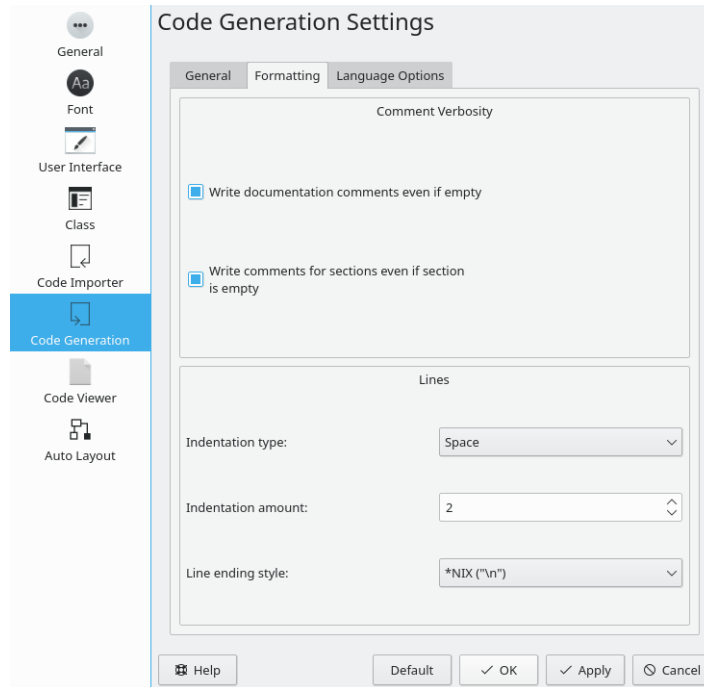
Si està marcada **Inclou els fitxers de capçalera des de la carpeta:**, permet a l'usuari especificar un camí en un camp editable o triar-lo amb un botó d'exploració.

#### 6.6.1.3 Política de sobreescritura

Quan el codi es genera a la carpeta especificada, aquesta opció determinarà què passa si es troba un fitxer amb el mateix nom.

- **Sobreescriu** el fitxer sense cap avís ni opció.
- **Pregunta** tant si se sobreescriu el fitxer com si es canvia el nom.
- **Usa un nom diferent** quan ja existeix un fitxer que el reanomena amb un sufix.

## 6.6.2 Pestanya Format de la Configuració de la generació de codi



*Opcions de la Configuració de la generació de codi - Format a l'Umbrello UML Modeller*

### 6.6.2.1 Detall en els comentaris

**Escriu els comentaris de la documentació encara que estiguin buits:** genera comentaris per a classes i funcions fins i tot si estan buides.

**Escriu els comentaris per a les seccions encara que la secció estigui buida:** escriu comentaris per a les seccions privades, protegides i públiques, fins i tot si estan buides.

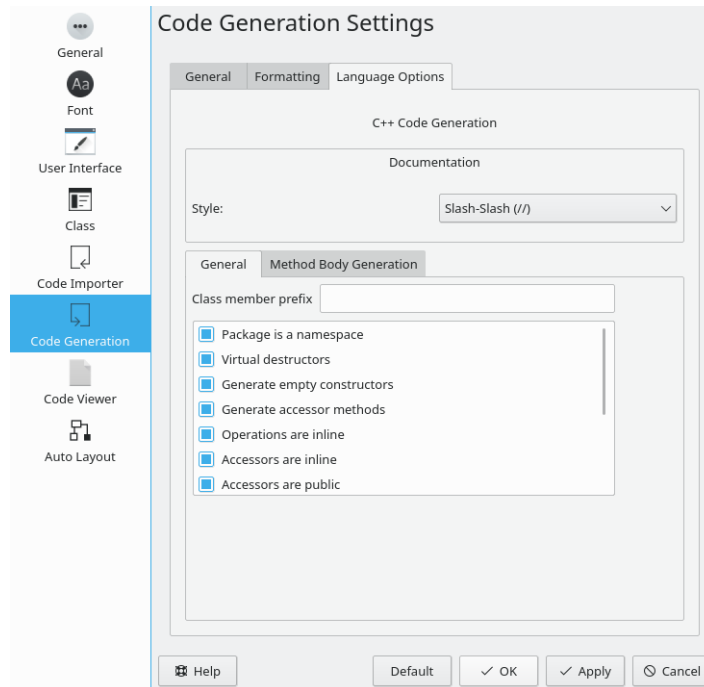
### 6.6.2.2 Línies

**Tipus de sagnat:** ofereix una opció entre sense sagnat, tabulació o espai.

**Quantitat de sagnat:** permet a l'usuari especificar el nombre d'espais per a la tabulació o l'opció de sagnar amb espais.

**Estil del final de línia:** és una opció entre els estils del final de línia de \*NIX, Windows i Mac.

### 6.6.3 Opcions del llenguatge



*Opcions de la Configuració de la generació de codi - Llenguatge a l'Umbrello UML Modeller*

Aquesta pàgina canvia per a cada llenguatge de programació seleccionat a la pestanya General. Actualment, les úniques opcions disponibles són per al llenguatge C++.

#### 6.6.3.1 Generació de codi C++

##### 6.6.3.1.1 Documentació

**Estil:** dona l'opció d'utilitzar «/\*\* \*/» o «//» com a estil de documentació

##### 6.6.3.1.2 General

Sota la pestanya **General** de la pestanya **Opcions del llenguatge**, es llisten diverses opcions de generació de codi.

- **Prefix dels membres de classe**

Una opció que permet afegir un prefix determinat per l'usuari als membres de classe quan es genera el codi.

- **Paquet és un espai de nom**

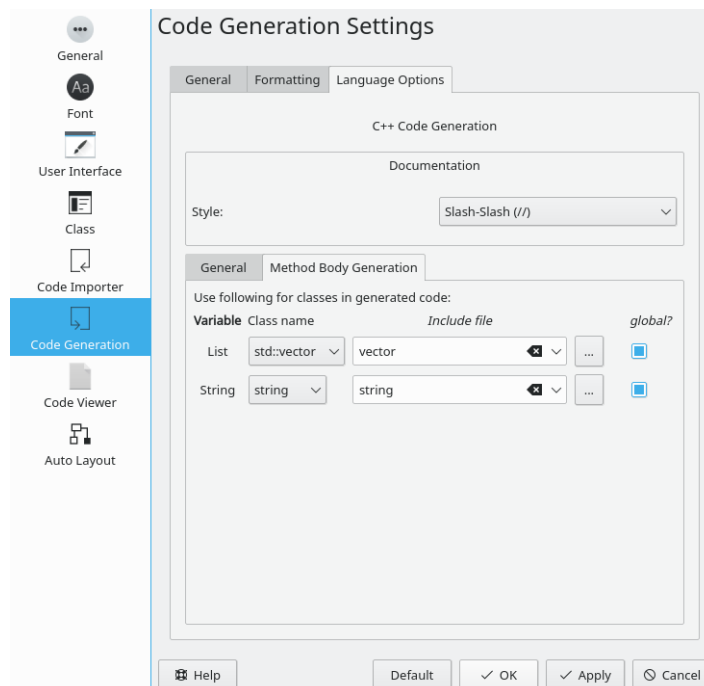
Els espais de nom proporcionen un mètode per a prevenir conflictes de nom en els projectes grans. Els símbols declarats dins d'un bloc d'espai de nom es col·loquen en un àmbit amb nom que evita que es confonguin amb els símbols amb noms idèntics en altres àmbits.

- **Destructors virtuals**

Tot i que els destructors no s'hereten, si una classe base declara el seu destructor virtual, el destructor derivat sempre l'anul·larà. Això fa possible eliminar objectes assignats dinàmicament de tipus polimòrfic a través de punters a la base.

- **Genera constructors buits**  
Generarà constructors que tenen claus buides.
- **Genera mètodes accessoris**  
Generarà mètodes per a accedir als tipus de dades.
- **Operacions són incloses**  
Genera els mètodes com a inclosos, però els compiladors són lliures de triar no incloure el mètode.
- **Accessors són inclosos**  
Els mètodes que accedeixen a les dades de la classe es generaran inclosos, però els compiladors són lliures de triar no incloure el mètode.
- **Accessors són públics**  
Els mètodes que es generen com a públics estaran disponibles per a qualsevol instanciació de la classe.
- **Crea els «getters» amb el prefix «get»**  
Col·locarà el prefix «get» en els mètodes que obtenen/retornen les dades de la classe.
- **Elimina el prefix «[a-zA-Z]\_» dels noms dels mètodes d'accés**  
Si s'ha introduït un prefix a **Prefix dels membres de classe**, aquesta opció l'eliminarà.
- **Els mètodes d'accés comencen per lletres majúscules**  
Això posa en majúscula la primera lletra del nom del mètode.
- **Usa «\» com a etiqueta de documentació en lloc de «@»**  
Una opció d'etiqueta per a utilitzar en documentar els paràmetres d'un mètode.

### 6.6.3.1.3 Generació del cos del mètode



Opcions de la Configuració de la generació de codi - Generació del cos del mètode l'Umbrello UML Modeller

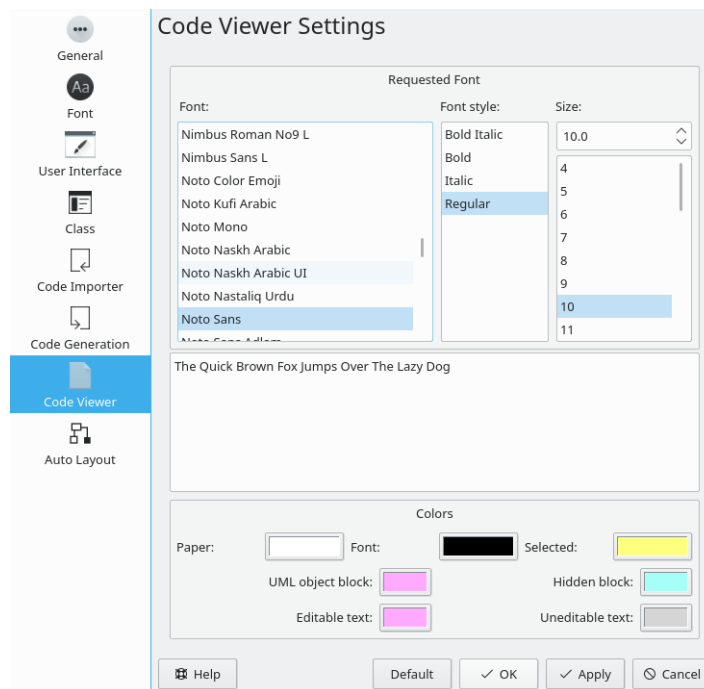
## Llista

Té opcions de QList, QVector i std::vector per al tipus de llista. Segueix un camp editable o seleccionable per a especificar el fitxer d'inclusió juntament amb un botó d'exploració per a trobar una selecció del fitxer d'inclusió. També hi ha una opció per a fer que la llista sigui global.

## Cadena

Opcions de cadena QString per al tipus de cadena. Segueix un camp editable o seleccionable per a especificar el fitxer d'inclusió juntament amb un botó d'exploració per a trobar una selecció del fitxer d'inclusió. També hi ha una opció per a fer que la cadena sigui global.

## 6.7 Configuració del visualitzador de codi

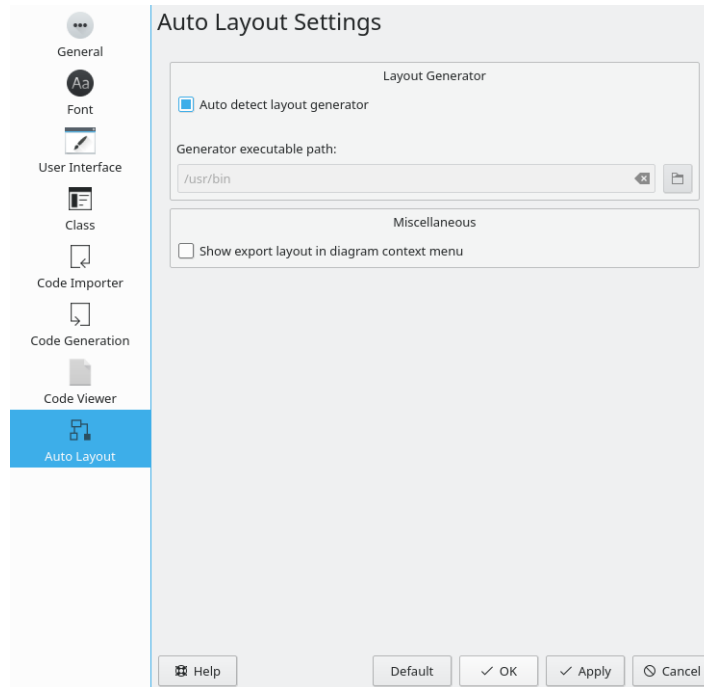


### *Opcions de la Configuració del visualitzador de codi a l'Umbrello UML Modeller*

Permet la personalització del visualitzador de codi. La secció **Tipus de lletra requerit** permet seleccionar el tipus de lletra, estil i mida del tipus de lletra. A sota de les eleccions es mostra una representació de les vostres eleccions.

A la secció **Colors**, es poden realitzar canvis en paper, tipus de lletra, seleccionada, bloc d'objecte UML, bloc ocult, text editable i text no editable. Es poden fer canvis en els colors fent clic al quadre de color al costat de l'etiqueta respectiva.

## 6.8 Configuració de la disposició automàtica



*Opcions de la Configuració de la disposició automàtica a l'Umbrello UML Modeller*

### **Detecta automàticament el generador de disposicions**

La funcionalitat de les disposicions automàtiques depèn dels generadors de disposicions proporcionats pel paquet GraphViz, el qual normalment s'instal·la a banda de l'Umbrello pel gestor de paquets. L'Umbrello integra la implementació per a detectar la ubicació de la instal·lació d'aquests generadors de disposicions. Pels casos en què aquesta dependència no està disponible o no s'ajusta, es podrà indicar un camí d'instal·lació diferent.

### **Mostra la disposició d'exportació en el menú contextual del diagrama**

L'exportació del fitxer «dot» es porta a terme usant la disposició d'exportació. Amb aquesta opció activada, la disposició d'exportació s'afegeix a la llista de disposicions de diagrama disponibles i activa una vista prèvia de l'exportació del «dot».



## Capítol 7

# Autors i història

El projecte va ser iniciat per Paul Hensgen com un dels seus projectes de la universitat. El nom original de l'aplicació era UML Modeller. Paul va fer tot el desenvolupament fins a finals del 2001 quan el programa va assolir la versió 1.0.

La versió 1.0 ja oferia un munt de funcionalitats, però després el projecte va ser revisat a la universitat del Paul; altres desenvolupadors van poder unir-se i van començar a fer valuoses aportacions al UML Modeller, com ara el canvi d'un format de fitxer binari a un fitxer XML, implementació per més tipus de diagrames UML, generació del codi i importació del codi simplement per a anomenar-ne unes quantes.

Paul es va haver de jubilar de l'equip de desenvolupament en l'estiu de 2002, però, com a programari de codi obert i gratuït que és, el programa continua millorant i evolucionant, i és mantingut per un grup de desenvolupadors de diferents parts del món. El setembre del 2002 el projecte va canviar el seu nom de UML Modeller, a Umbrello UML Modeller. Hi ha diverses raons pel canvi de nom, la més important és que només «uml» — conegut comunament com a — era un nom massa genèric i causava problemes amb algunes distribucions. L'altra raó important és que els desenvolupadors van pensar que Umbrello és un nom molt més fresc.

El desenvolupament de l'Umbrello UML Modeller així com els debats pel que fa a on el programa s'ha d'adreçar en les futures versions està obert i es porta a terme a través d'Internet. Si desitgeu contribuir al projecte, si us plau no dubteu a contactar amb els desenvolupadors. Hi ha moltes maneres en què podeu ajudar a l'Umbrello UML Modeller:

- Informant d'errors o suggerint millores
- Arranjant errors o afegint característiques
- Escrivint una bona documentació o traduint-la a altres idiomes
- I per descomptat.... programant amb nosaltres!

Com podeu veure, hi ha moltes maneres de poder contribuir. Totes elles són molt importants i tothom hi és benvingut a participar.

Els desenvolupadors de l'Umbrello UML Modeller es poden trobar en [umbrello-devel@kde.org](mailto:umbrello-devel@kde.org).

## Capítol 8

# Copyright

Copyright 2001, Paul Hensgen

Copyright 2002-2020, Els autors de l'Umbrello UML Modeller.

Aquesta documentació està llicenciada d'acord amb les clàusules de la [Llicència de Documentació Lliure de GNU](#).

Aquest programa està llicenciat d'acord amb les clàusules de la [Llicència Pública General de GNU](#).