

1. Diagrames estàtics

1.1 Anàlisi i disseny orientat a objectes

El disseny estructurat, o també anomenat disseny orientat al flux de dades, permet establir la transició del Diagrama de Flux de Dades (DFD) a una descripció de l'estructura del programa, que es representa mitjançant un diagrama d'estructures. Les eines que faciliten i automatitzen aquest tipus de disseny es coneixen com a eines CASE. A la figura.1 es pot observar un exemple de Diagrama de Flux de Dades.

Es pot considerar que el procés de desenvolupament de programari té les mateixes fases. Segons diversos autors, pot haver-hi grans divergències entre les propostes de fases d'un projecte, però, en general, les proposades són aquestes:

- **Estudi previ**, on es delimita l'àmbit del projecte de desenvolupament d'un programari i se'n determina la viabilitat, els riscos, els costos i el calendari.
- **Anàlisi de requeriments**, on es descriuen de manera adient els requisits del programari, que són les necessitats d'informació que el programari ha de satisfer. L'anàlisi ha de deixar molt clar el que es vol fer a partir de l'estudi d'un problema determinat. Els requeriments que cal recollir hauran de ser funcionals i no funcionals. Caldrà que sigui independent de la tecnologia i del tipus de programació escollits per al desenvolupament del projecte.
- **Disseny**, on es projecta el programari amb vista a implementar-lo en un entorn tecnològic concret, definit per aspectes com ara el llenguatge de programació, l'entorn de desenvolupament i el gestor de bases de dades, de tots els quals en pot intervenir més d'un dins un sol projecte. A partir del que caldrà fer, definit a la fase anterior al disseny, es determina el com es farà. Caldrà tenir en compte les possibles conseqüències de les decisions preses en aquesta fase. El detall haurà de ser suficientment baix com perquè a partir d'aquestes indicacions es pugui desenvolupar el projecte.
- **Desenvolupament i proves**, on es porta a terme el desenvolupament del codi font que satisfaci les funcionalitats establertes a les fases anteriors. Aquesta fase no només es limita a la codificació manual, sinó que pot incloure la programació gràfica, la generació automàtica de codi i altres tècniques. En aquesta fase també s'inclouran les proves del programari. Algunes parts de la programació poden començar abans que el disseny estigui enllestit del tot, i també es poden provar parts del programari mentre encara se n'estan implementant d'altres.

- **Finalització i transferència**, on es prepara el programari per tal que estigui en explotació. Durant aquesta fase, és objecte de manteniment per eliminar errors no corregits durant la prova, introduir-hi millores i adaptar-lo a canvis tecnològics i organitzatius que es produeixin en l'entorn de treball dels usuaris.

1.2. Llenguatge unificat de modelització

1.2.1. Diagrames

una possible classificació dels diagrames es fa en funció de la visió del model del sistema que ofereixen. Les dues visions diferents de model de sistema que poden representar els diagrames UML són:

- **Visió estàtica (o estructural)**: per oferir aquest tipus de visió s'utilitzen objectes, atributs, operacions i relacions. La visió estàtica d'un sistema dóna més valor als elements que es trobaran en el model del sistema des d'un punt de vista de l'estructura del sistema. Descriuen aspectes del sistema que són estructurals i, per tant, permanents (allò que el sistema té).
- **Visió dinàmica (o de comportament)**: per oferir aquest tipus de visió es dóna més valor al comportament dinàmic del sistema, és a dir, a allò que ha de passar en el sistema. Amb els diagrames de comportament **es mostra** com es modela la col·laboració entre els elements del sistema i els seus canvis d'estat. Representen allò que pot fer el sistema modelitzat.

Diagramas estáticos (7 tipos):

- **Diagrama de paquets**, que representa essencialment les relacions de diferents menes entre els continguts de diferents paquets d'un model.
- **Diagrama de classes**, que probablement molts consideren el diagrama principal, atès que descriu classificadors de tota mena i diferents tipus de relacions entre ells abans de fer-los servir en altres diagrames.
- **Diagrama d'objectes**, que representa instàncies de classificadors definits en un diagrama de classes previ i relacions entre elles.
- **Diagrama d'estructures compostes**, que descriu casos en què, o bé les instàncies d'un classificador tenen com a parts instàncies d'altres, o bé en el comportament executant d'un classificador participen instàncies d'altres.
- **Diagrama de components**, que és un diagrama de classes i alhora d'estructures compostes simplificat i més adient per a determinades tecnologies de programació.
- **Diagrama de desplegament**, que descriu la configuració en temps d'execució d'un programari especificat, normalment, per un diagrama de components.

- **Diagrama de perfil**, permet adaptar o personalitzar el model amb construccions que són específiques d'un domini en particular, d'una determinada plataforma, o d'un mètode de desenvolupament de programari...

Diagramas dinámicos(7 tipos):

- **Diagrama de casos d'ús**, que considera els comportaments d'un sistema principalment des del punt de vista de les interaccions que té amb el món exterior.
- **Diagrama d'estats**, en el qual es descriuen les diferents situacions -estats- des del punt de vista dels seus comportaments, de les instàncies d'un classificador, alhora que les causes, condicions i conseqüències dels canvis d'una situació a una altra.
- **Diagrama d'activitats**, en què es descriu un comportament complex en forma de seqüències condicionals d'activitats components.

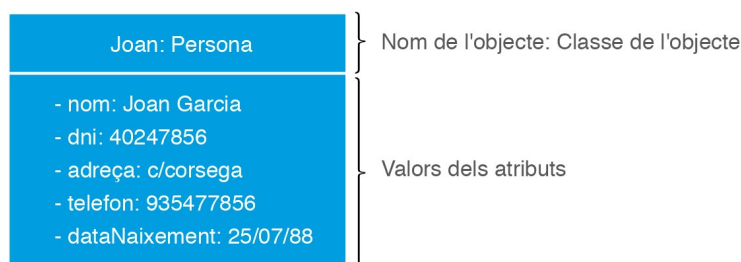
Diagrama d'interacció, que descriu comportaments emergents i té les variants següents:

- **Diagrama de seqüències**, que fa èmfasi en la seqüència temporal de les participacions de les diferents instàncies.
- **Diagrama de comunicacions**, que es basa directament en una estructura composta.
- **Diagrama de visió general de la interacció**, que dona una visió resumida del comportament emergent.
- **Diagrama temporal**, que es basa en un diagrama d'estats previ o més d'un i alhora posa èmfasi en els canvis d'estat al llarg del temps.

Diagrama de Clases:

Un diagrama de classes representa les classes que seran utilitzades dins el sistema i les relacions que existeixen entre elles.

Representación en diagrama de clases de un objeto:



Una **associació** és un classificador que defineix una relació entre diversos classificadors, que estableix connexions amb un cert significat entre les instàncies respectives.

Classificador:

Un classificador és un tipus els valors del qual tenen en comú característiques estructurals i de comportament, que són elements que es defineixen a nivell del classificador. Cadascun d'aquests valors és una instància del classificador.

Tipos de relaciones:

1. La **relació d'associació** es representa mitjançant una línia contínua sense fletxes ni cap altre símbol als extrems. És el tipus de relació (anomenada de forma genèrica associació) que s'ha estat explicant fins al moment. És un tipus de relació estructural que defineix les connexions entre dos o més objectes, la qual cosa permet associar objectes que instancien classes que col·laboren entre si.
2. Una **relació d'associació d'agregació** és un cas especial d'associació entre dos o més objectes. Es tracta d'una relació del tipus **tot-part**. Aquest tipus de relació implica **dos tipus d'objectes**, l'objecte anomenat base i l'objecte que estarà inclòs a l'objecte base. La relació indica que l'objecte base necessita de l'objecte inclòs per poder existir i fer les seves funcionalitats. Si desapareix l'objecte base, el o els objectes que es troben inclosos en l'objecte base no desapareixeran i podran continuar existint amb les seves funcionalitats pròpies. La relació d'associació d'agregació es representa mitjançant una línia contínua que finalitza en un dels extrems per un rombe buit, sense omplir. El rombe buit s'ubicarà a la part de l'objecte base. A la figura.13 es pot observar un exemple de relació d'associació d'agregació. L'objecte base és l'objecte anomenat Fruiteria. Els objectes inclosos a la fruiteria són: Pomes, Taronges, Peres i Maduixes. S'estableix una relació entre aquestes classes del tipus tot-part, on les fruites són part de la fruiteria. Això sí, si la fruiteria deixa d'existir, les fruites continuen existint.
3. Una **relació d'associació de composició** és també un cas especial d'associació entre dos o més objectes. És una relació del tipus **tot-part**. És una relació molt semblant a la relació d'associació d'agregació, amb la diferència que hi ha una dependència d'existència entre l'objecte base i l'objecte (o els objectes) que hi està inclòs. És a dir, si deixa d'existir l'objecte base, deixarà d'existir també el o els objectes inclosos. El temps de vida de l'objecte inclòs depèn del temps de vida de l'objecte base. La relació d'associació de composició es representa mitjançant una línia contínua finalitzada en un dels extrems per un rombe pintat, omplert. El rombe pintat s'ubicarà a la part de l'objecte base. A la

figura.14 es mostra un exemple d'una relació d'agregació. L'objecte base Mà es compon dels objectes inclosos Menovell, Anular, Cor, Índex i Polze. Sense l'objecte Mà la resta d'objectes deixaran d'existir.

4. Un altre tipus de relació entre classes és la **relació de dependència**. Aquest tipus de relació es representa mitjançant una fletxa discontinua entre dos elements. L'objecte del qual surt la fletxa es considera un objecte dependent. L'objecte al qual arriba la fletxa es considera un objecte independent. Es tracta d'una relació semàntica. Si hi ha un canvi en l'objecte independent, l'objecte dependent es veurà afectat.
5. La **relació de generalització** es dona entre dues classes on hi ha un vincle que es pot considerar d'herència. Una classe és anomenada classe mare (o superclasse). L'altra (o les altres) són les anomenades classes filles o subclasses, que hereten els atributs i els mètodes i comportament de la classe mare. Aquest tipus de relació queda especificat mitjançant una fletxa que surt de la classe filla i que acaba a la classe mare. A la figura.15 es pot veure un exemple on l'element actor és independent. Hi ha una dependència normal de l'element pel·lícula en relació amb l'element actor, ja que actor es fa servir com a paràmetre als mètodes afegir i eliminar de pel·lícula. Si hi ha canvis a actor, l'objecte pel·lícula es veurà afectat.

Diagrama d'objectes:

El **diagrama d'objectes** només pot contenir instàncies i relacions entre objectes: enllaços i dependències que tinguin lloc entre instàncies. Els classificadors i les associacions respectives han d'haver estat definits prèviament en un diagrama de classes.

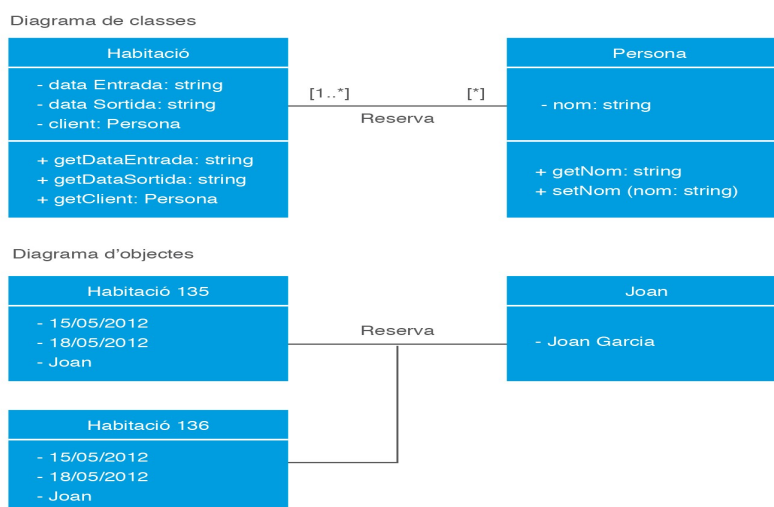


Diagrama de paquets:

El **diagrama de paquets** serveix per descriure l'estructura d'un model en termes de paquets interrelacionats.

Diagrama d'estructures compostes:

Una **estructura composta** és un conjunt d'elements interconnectats que col·laboren en temps d'execució per aconseguir algun propòsit. Es tracta d'un diagrama que complementa el diagrama de classes.

Diagrama de components:

Un component és una peça del programari que conforma el sistema, peça que pot ser reutilitzable i fàcilment substituïble. Un component sol fer ús d'una interfície que defineix les operacions que el component implementarà.

El diagrama de components mostra els components que conformen el sistema i com es relacionen entre si.

Diagrama de desplegament:

El **diagrama de desplegament** descriu la distribució de les parts d'una aplicació i les seves interrelacions, tot en temps d'execució.