

Alumnes: James Chen, Albert Villanueva

Pràctica 3 Part 2

Continguts:

[Descripció de les classes](#)

[Diagrama de classes](#)

[Pregunta 1](#)

[Pregunta 2](#)

[Pregunta 3](#)

[Pregunta 4](#)

[Pregunta 5](#)

[Proves de funcionament](#)

[Observacions generals](#)

Descripció de les classes

Model:

El model contindrà totes les classes que modelen les dades que s'han de gestionar dins de l'aplicació.

- Dades: La classe Dades és la classe principal del paquet del model, atès que conté i gestiona totes les dades de l'aplicació.
- La classe Bitacola ha de tenir un atribut de tipus ArrayList per guardar objectes de tipus PaginaBitacola. A més, la classe ha d'implementar la interfície InBitacola i tenir el mètode toString que generi una representació com a String de totes les pàgines de la bitàcola separades per salts de línia ('\n')

- La classe BombaRefrigerant ha d'implementar la interfície InBombaRefrigerant. Cada bomba refrigerant ha de tenir un identificador numèric i dos booleans per indicar si està activada o no i si està fora de servei.
- El generador de vapor transmet la calor transportat pel sistema de refrigeració (input) al circuit secundari (output) amb una determinada eficiència, fixada en 0.8 al nostre cas. Es considera 30°C la temperatura ambient.
- Aquesta classe implementa la classe base d'altres tres classes usades per representar pàgines de bitàcola (vegeu Apartat 2.4). La classe PaginaBitacola conté un atribut de tipus enter per guardar el nombre del dia al qual es refereix la pàgina.
- Aquesta classe també hereta de PaginaBitacola i ha de tenir el atributs necessaris segons el que es va a explicar a l'Apartat 2.4.1.
- La classe PaginaEstat hereta de PaginaBitacola i ha de tenir el atributs necessaris segons el que es va a explicar a l'Apartat 2.4.2.
- La classe PaginaIncidencies hereta de PaginaBitacola i conté una llista de String descrivint incidències, segons es va explicar a l'Apartat 2.4.3.
- El reactor rep com a input el grau d'inserció de les barres de control en percentatge i genera com a output una determinada quantitat de graus Celsius que es transmet a l'aigua del circuit primari.
- El sistema de refrigeració té com a input el calor generat pel reactor i genera com a output la quantitat de calor que és capaç de transportar des del reactor ns al generador de vapor. Aquesta quantitat depèn de la quantitat de bombes refrigerants que estiguin activades. A la nostra central hi haurà un total de 4 bombes refrigerants.
- La turbina converteix els graus del generador de vapor en unitats de potència. Això només es podrà fer si la temperatura generada com a output pel generador de vapor és superior a 100 graus (la temperatura de vaporització de l'aigua).
- VariableUniforme: aquesta classe es fa servir per determinar aleatoriament si una bomba refrigerant estarà fora de servei al començar un nou dia. Només es declara un objecte d'aquesta classe dins del constructor de la classe Dades i es proporciona com a paràmetre al constructor de la classe BombaRefrigerant. La classe BombaRefrigerant utilitza el mètode seguentValor per generar un nombre aleatori que determinarà si la bomba es queda fora de servei.

Adaptador:

El paquet adaptador únicament contindrà la classe Adaptador, que funcionarà com a mediador entre la vista i el model. La vista només podrà utilitzar aquesta classe per accedir a la informació del model.

- La classe Adaptador és el mediador entre la vista i el model. Concretament, connecta la classe Dades del model i la classe CentralUB de la vista. Aquesta classe també conté els mètodes de persistència de dades.

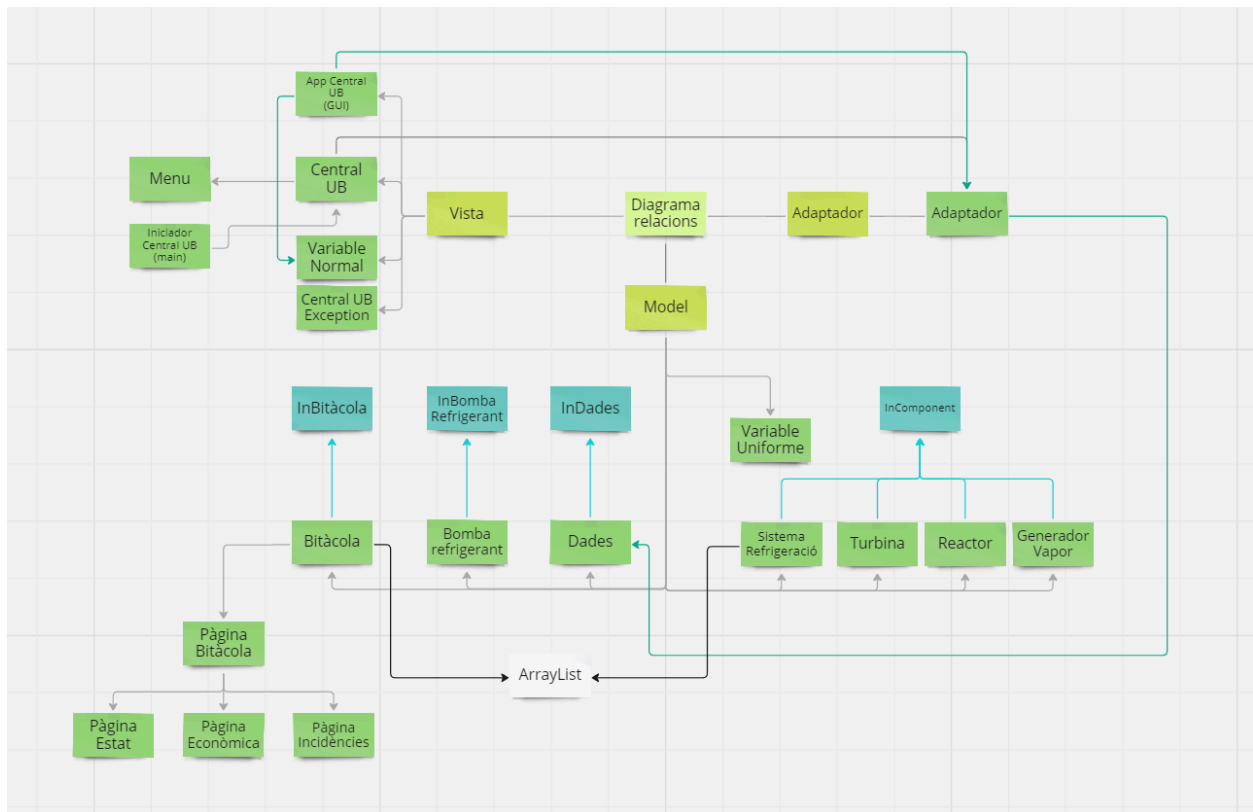
Vista:

La vista contindrà totes les classes relacionades amb el maneig de menú d'opcions. Entre elles, destaca la classe CentralUB, que implementarà el menú d'opcions ofert per l'aplicació i la classe AppCentralUB, que proporcionarà la interfície gràfica per poder interactuar amb l'aplicació.

- La classe AppCentralUB estarà dins del paquet Vista i correspondrà al formulari principal des d'on s'inicia l'aplicació. Aquest formulari serà de tipus JFrame. Des d'aquesta classe AppCentralUB haureu de poder realitzar les 4 funcionalitats principals detallades en l'apartat anterior. Haureu d'accedir a altres formularis per tal d'implementar la gestió de les components de la central, la visualització d'informació de la central, i haureu de poder finalitzar el dia i guardar i recuperar les dades. Per tal de realitzar aquestes funcionalitats haureu de considerar diverses opcions possibles, per exemple a través de botons, barra de menús, menú desplegable, etc.
- CentralUB: Deneix el menú d'opcions i la lògica de l'aplicació. És molt important ressaltar que des de CentralUB no es pot importar cap classe del paquet model. Només podem importar la classe Adaptador.
- CentralUBException: Aquesta classe serà utilitzada per gestionar excepcions.
- IniciadorCentralUB: Conté la classe principal de l'aplicació.
- Menú: Classe que serà utilitzada com a template per implementar un menú a CentralUB.
- VariableNormal: aquesta classe es fa servir dins de la classe CentralUB per generar la demanda de potència elèctrica de cada nou dia.

2. Diagrama de classes

- Els rectangles de color verd clar són els paquets - Els rectangles de color verd fosc són les classes
- Els rectangles de color blau són les interfícies
- El rectangle blanc és una classe predeterminada de Java
- Les fletxes de color gris significa que deriven de la classe a la qual són apuntades
- Les fletxes de color blau són les classes que implementen interfícies
- Les fletxes de color verd són unes connexions especials que permeten transmetre informació d'un paquet a l'altre



Pregunta 1

Imagina que un company teu et passés la seva implementació de la classe Dades perquè la substituïssis en el teu codi, quines modificacions serien necessàries en la vista i el adaptador? Com es relaciona la interfície InDades amb la pregunta anterior? Justifica la resposta.

Si la implementació de la classe Dades conté tots els mètodes de la interfície InDades no caldria modificació addicional si s'executa l'aplicació des de IniciadorCentralUB. Si s'executa des de AppCentralUB, caldria afegir alguns mètodes a dades i a adaptador per poder accedir a les dades necessàries a AppCentralUB (que és el que hem fet).

Doncs que la interfície InDades no conté tots els mètodes necessaris per poder mostrar tota la informació a AppCentralUB, encara que alguns poden ser creats en adaptador.

Pregunta 2

Quines conseqüències tindria que el mètode mostralIncidencies de la classe Adaptador retornés una llista d'objectes de la classe Incidencia en lloc d'un String?

Doncs que en la implementació de AppCentralUB o InciadorCentralUB s'hauria de recórrer la llista i convertir cada element a String per poder mostrar la informació.

Pregunta 3

Expliqueu quines classes has pogut reutilitzar de la pràctica 3 part 1 per a fer la part 2. Quins canvis sobre les classes reutilitzades has necessitat fer i perquè.

Hem pogut reutilitzar totes les classes de la part 1. Hem hagut d'afegir mètodes a la classe adaptador per poder mostrar tota la informació en AppCentralUB.

Pregunta 4

Indiqueu quins tipus d'esdeveniments heu fet servir al vostre codi.

Hem utilitzat esdeveniments de tipus actionPerformed. Per exemple:

'btnGuardarDadesActionPerformed' o 'btnFinalitzarDiaActionPerformed'
Per modificar informació al produir-se una acció.

També de tipus stateChanged. Per exemple: 'sldBarresControlStateChanged' per canviar estats de components.

I també de tipus componentHidden. Per exemple:

'FrmVisualitzarInformacioCentralComponentHidden' o
'FrmGestioComponentsCentralComponentHidden'
Per amagar certs components.

Pregunta 5

Explica quins canvis hauríeu de realitzar en l'aplicació si es volgués afegir la funcionalitat següent: En iniciar l'aplicació cal obrir una subfinestra en la qual l'usuari ha d'introduir el seu nom i aquest quedi registrat.

Una manera d'assolir-ho seria en el constructor de l'aplicació afegir un JDialog amb la finestra en la qual s'ha d'introduir el nom (usuari i/o contrasenya). Si les dades són correctes es continua a la finestra del menú. Si no, apareix una finestra dient que les dades entrades no són vàlides.

Per que la informació quedi enregistrada es podria fer un botó d'enregistrament i un d'entrada a l'aplicació (register i login). Per enregistrar es pot crear una variable d'adaptador o en la classe AppCentralUB que sigui una llista String amb la informació d'accés. Aquestes dades després es poden emmagatzemar per separat en un arxiu o juntament amb la resta de les dades.

Proves de funcionament

En el vídeo adjuntat ensenyem el correcte funcionament de l'aplicació. Comprovant el funcionament de tots els components i funcionalitats demanades:

- o Modificar el percentatge d'inserció de les barres de control (75%).
- o Activar dues bombes refrigerants.
- o Activar el reactor.
- o Finalitzar el dia fins que aparegui una incidència en el Reactor.
- o Intentar activar el reactor.

- o Finalitzar el dia fins que aparegui una incidència en alguna de les bombes refrigerants.
- o Intentar activar la bomba refrigerant fora de servei.
- o Mostrar les incidències.
- o Mostrar l'estat de la central.
- o Mostrar el quadern de bitàcola.
- o Guardar les dades de la central.
- o Desactivar totes les bombes refrigerants (en cas que n'hi hagi alguna d'activa).
- o Finalitzar el dia.
- o Carregar les dades de la central.
- o Mostrar el quadern de bitàcola.
- o Mostrar el percentatge d'inserció de les barres de control i l'estat de les bombes refrigerants i el reactor.

Observacions generals

En la part 1 i 2 de la pràctica hem tingut errors diversos sobre el funcionament teòric de la central que ens ha costat entendre. Especialment requisits específics com les temperatures mínimes i el flux de dades de la reacció. Quant a la part 2 ens ha costat entendre el funcionament de la GUI entorn la part de disseny i codi. Creiem que la dificultat principal en aquesta part 2 ha sigut familiaritzar-se amb el funcionament i estructura dels components gràfics.