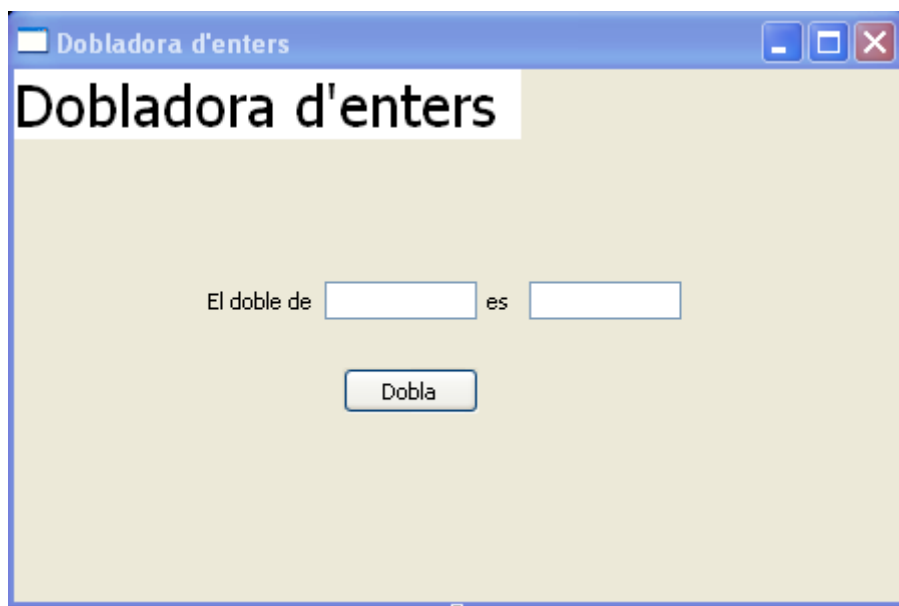


## CFGS Desenvolupament d'Aplicacions Multiplataforma (DAM) CFGS Desenvolupament d'Aplicacions Web (DAW)

### Mòdul 5 – Entorns de Desenvolupament. UF1 – Desenvolupament de programari

#### Exemple amb WindowBuilder pas a pas



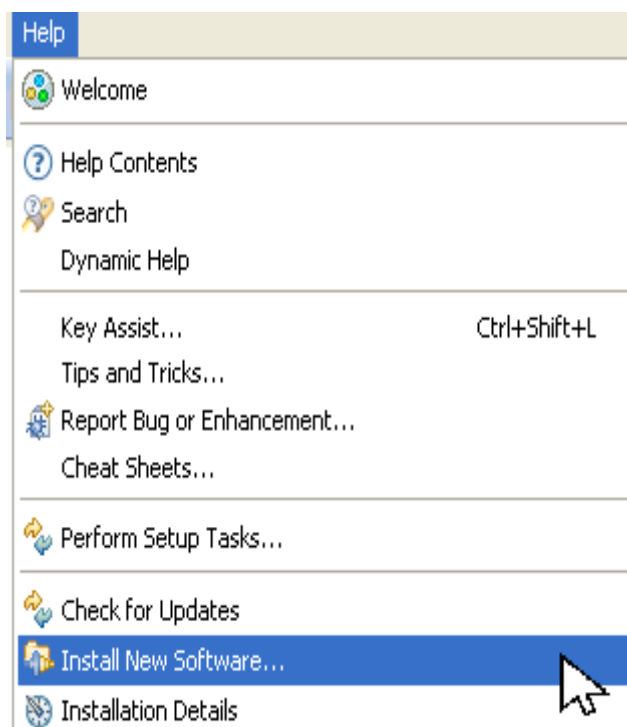
### Índex de continguts

1. Instal·lació de WindowBuilder.....	3
2. Exercici.....	5
3. Creació del projecte.....	6
4. Creació de la classe que implementa la finestra.....	8
5. Afegint comportament al botó.....	13
6. Execució.....	15

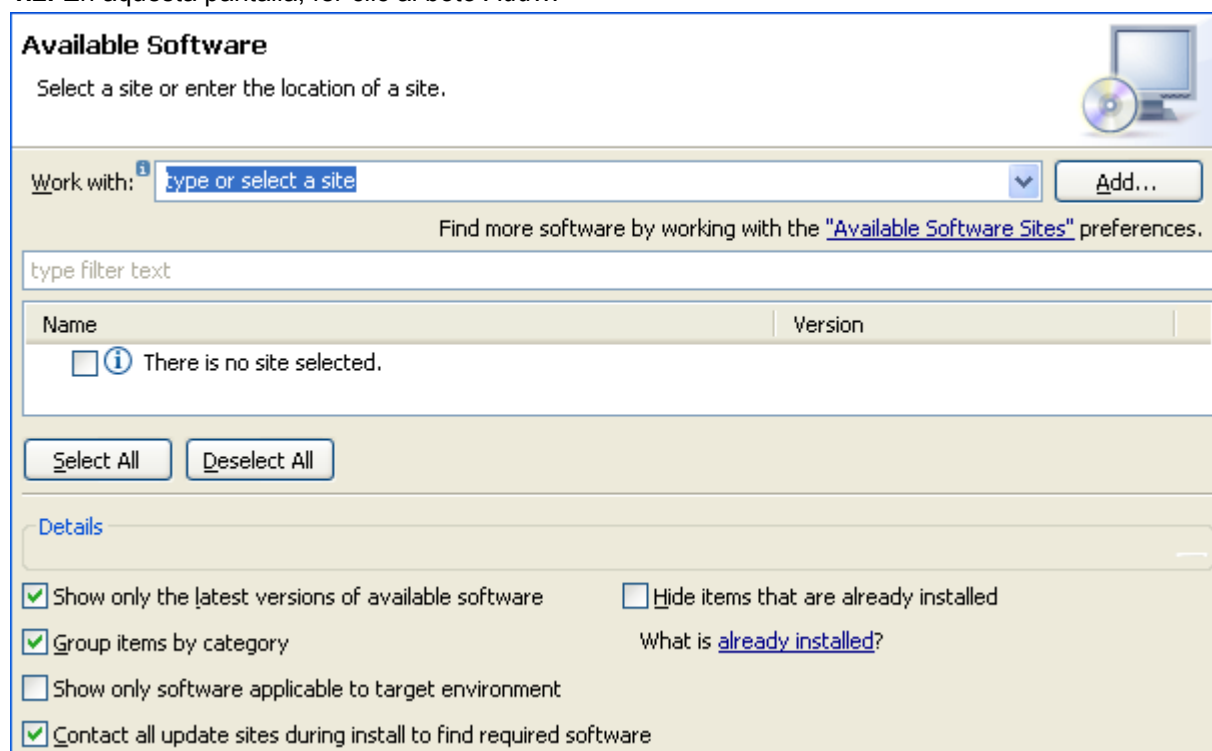
### 1. Instal·lació de WindowBuilder

Aquesta operació només cal fer-la una vegada (no per a cada projecte).

#### 1.1. Seleccionar l'opció *Help* → *Install New Software...*



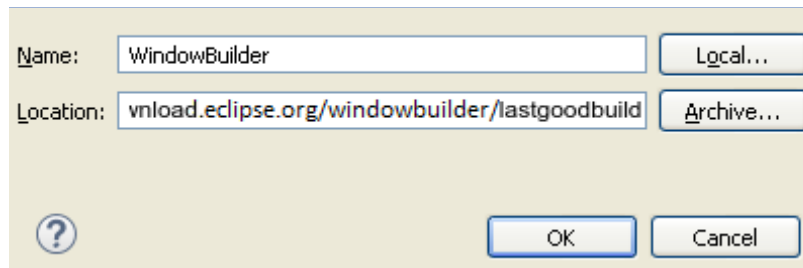
#### 1.2. En aquesta pantalla, fer clic al botó *Add...*



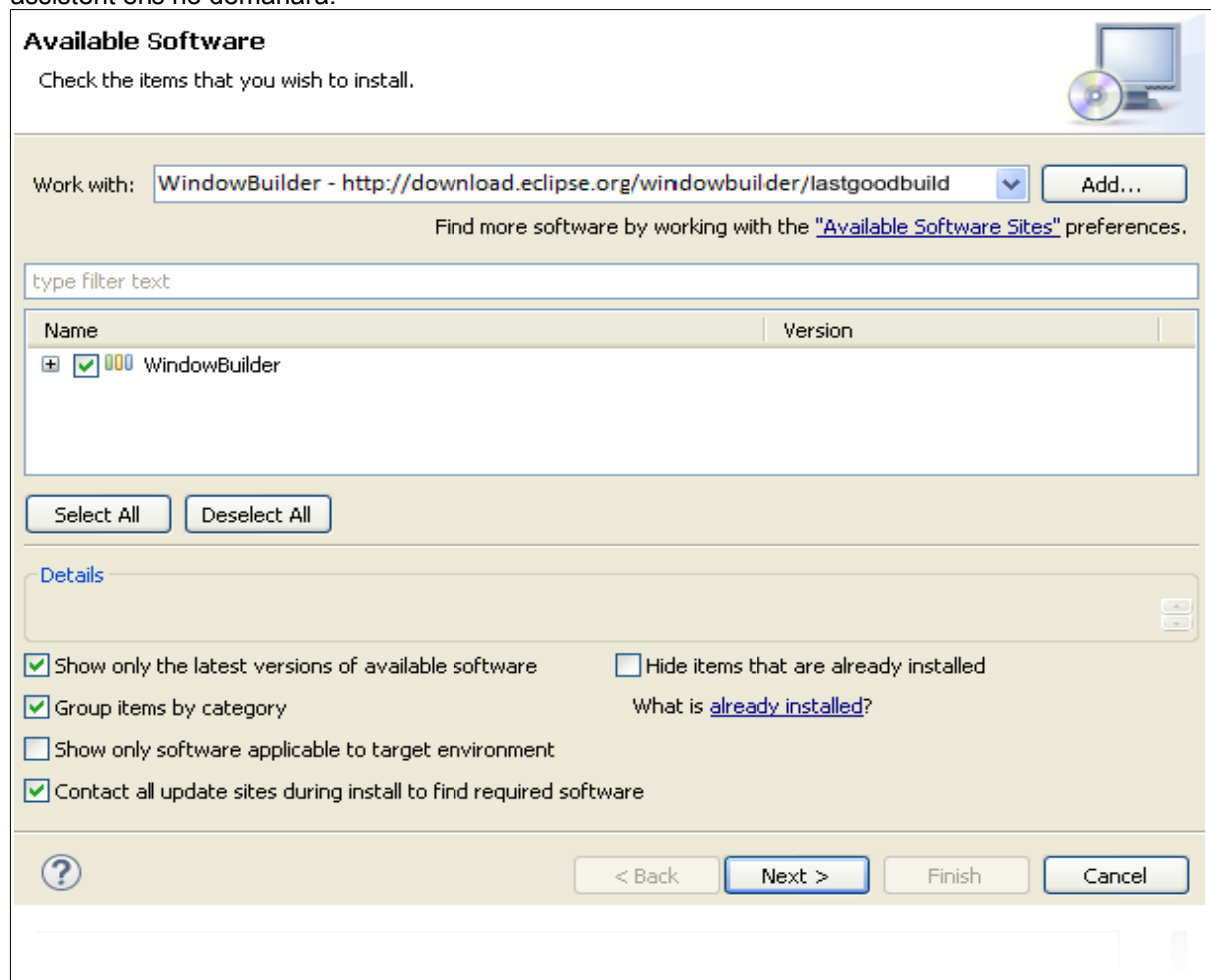
**1.3.** A la finestra que apareix posarem *WindowBuilder* a l'apartat *Name* i, a l'apartat *Location*, <http://download.eclipse.org/windowbuilder/lastgoodbuild> , que és l'adreça des d'on s'instal·la el connector.

Després farem clic a *Ok*.

**Atenció:** no té sentit accedir a aquesta adreça des del navegador.



**1.4.** A la finestra que apareix, cal fer clic a *Select All* (es marcaran totes les opcions, tal com apareix a la imatge) i, a continuació, fer clic a *Next >* i seguir l'assistent. Caldrà que reiniciem Eclipse. El mateix assistent ens ho demanarà.

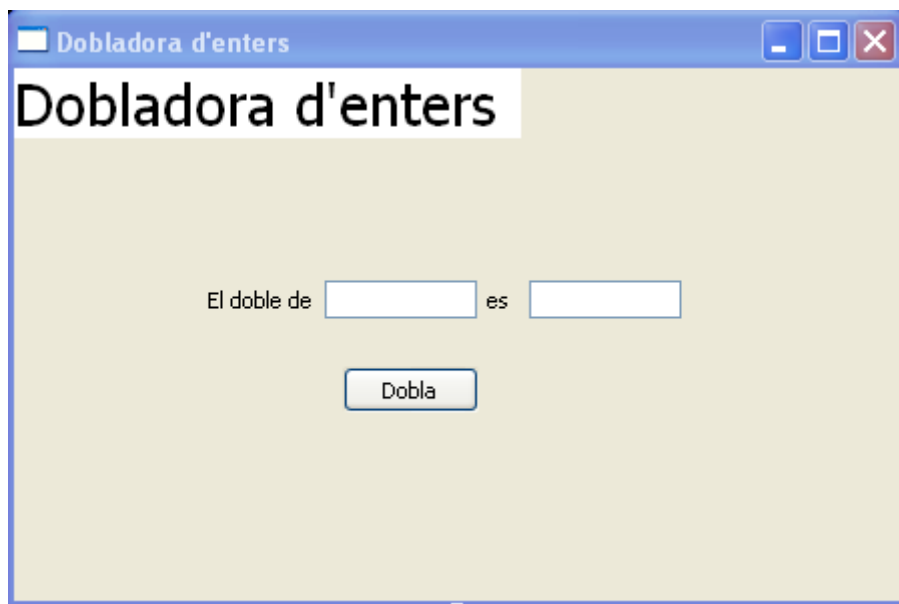


### 2. Exercici

Crearem una mini aplicació gràfica que tindrà dos quadres de text i un botó. En el primer quadre de text es podrà entrar un valor enter. Fer clic al botó provocarà que el programa calculi i escrigui al segon quadre de text el doble del valor entrat al primer quadre de text. Aquest segon quadre de text **no** serà editable.

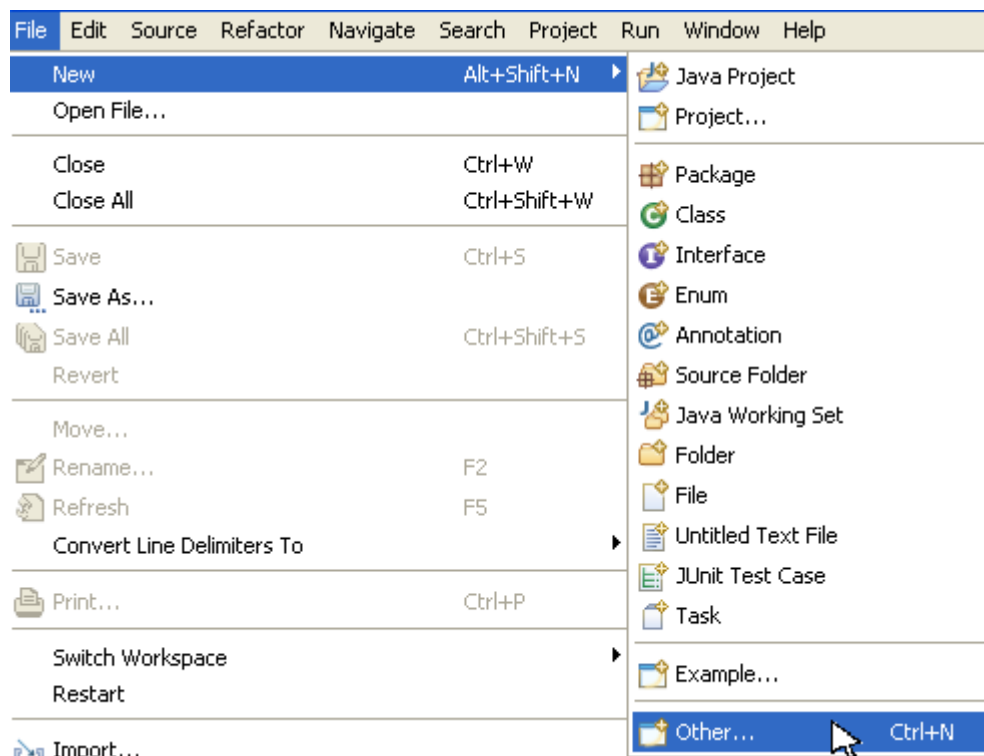
En cas que es premi el botó i al primer quadre no hi hagi un valor enter correcte, el programa obrirà una finestra emergent per informar de l'error i no escriurà cap resultat.

L'aspecte de la pantalla serà similar a aquest:

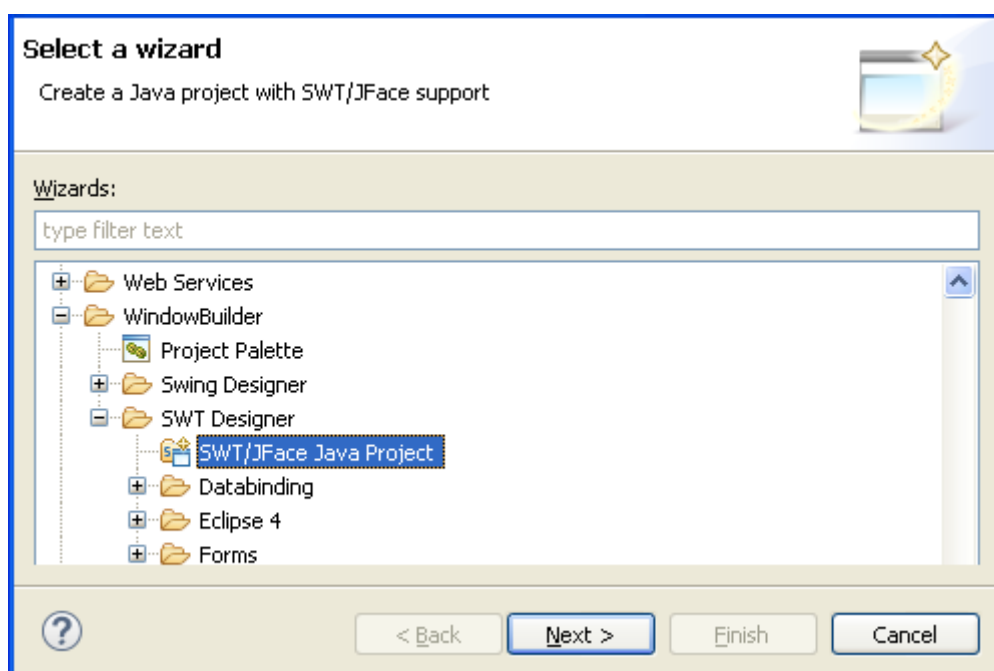


## 3. Creació del projecte

### 3.1. Cal seleccionar l'opció del menú *File* → *New* → *Other...*



### 3.2. A la pantalla següent, cal seleccionar *WindowBuilder* → *SWT Designer* → *SWT/Face Java Project* i fer clic a *Next* >



Ens apareixerà aquesta pantalla:

**Create a Java Project**

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:

**JRE**

☒ Use an execution environment JRE:

☐ Use a project specific JRE:

☐ Use default JRE (currently 'jre7') [Configure JREs...](#)

**Project layout**

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

**Working sets**

☐ Add project to working sets

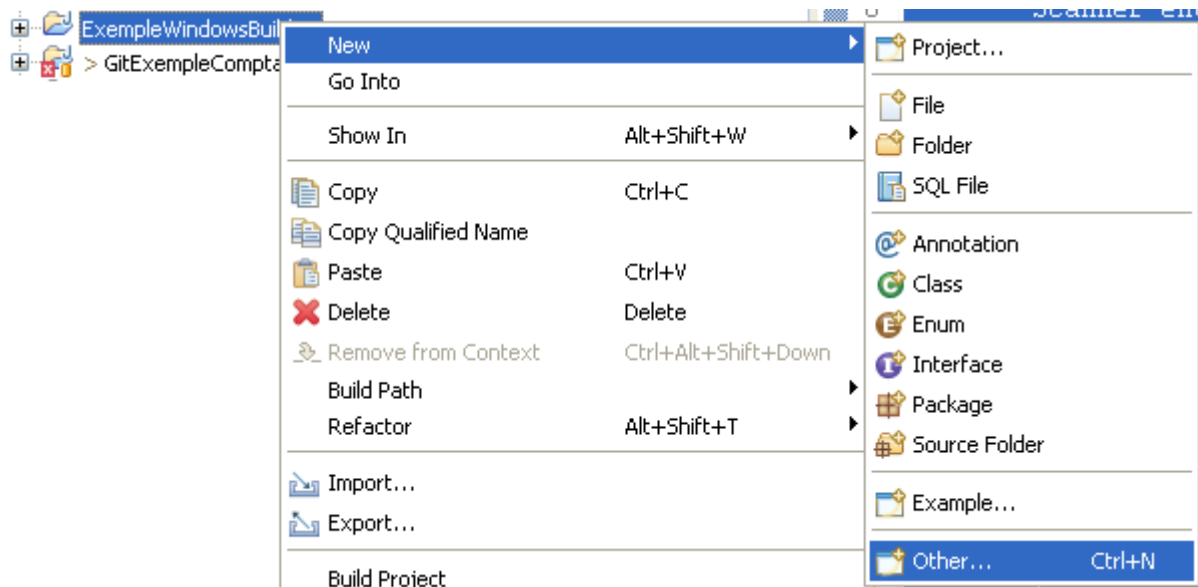
Working sets:

Cal escriure-hi el nom del projecte, *ExempleWindowBuilder*. La resta d'opcions poden deixar-se com surten per defecte. Un cop fet això, podeu fer clic a *Finish*.

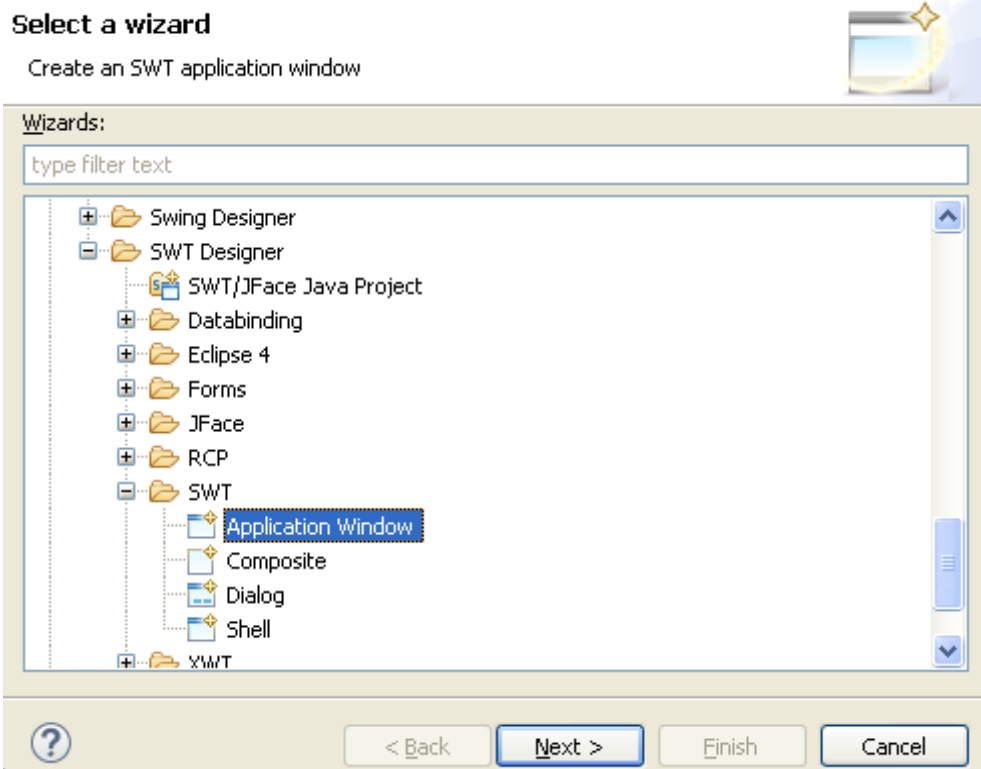
### 4. Creació de la classe que implementa la finestra

En aquest cas el programa estarà format només per una classe que crearà la finestra i contindrà també el mètode *main*. L'assistent ens facilitarà la generació de la majoria del codi.

**4.1.** Creació de la classe: des de la icona que representa el projecte (fent-hi clic amb el botó secundari -el botó dret en el cas dels destres-) cal seleccionar *New* → *Other...*

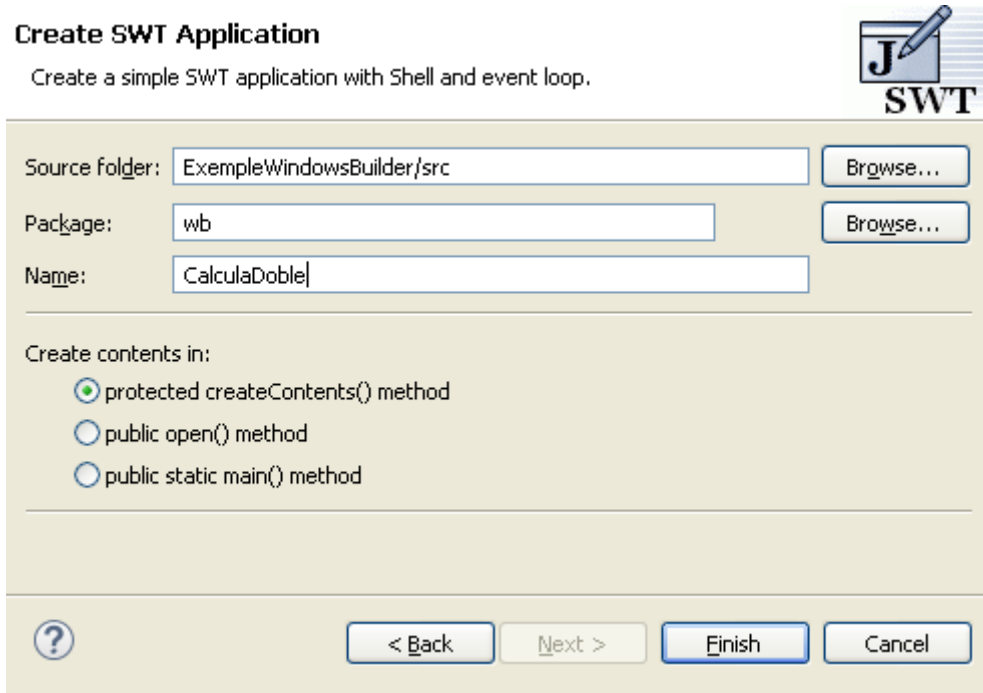


**4.2.** A la pantalla que apareix, cal seleccionar *WindowBuilder* → *SWT Designer* → *SWT* → *Application Window* i fer clic a *Next >*





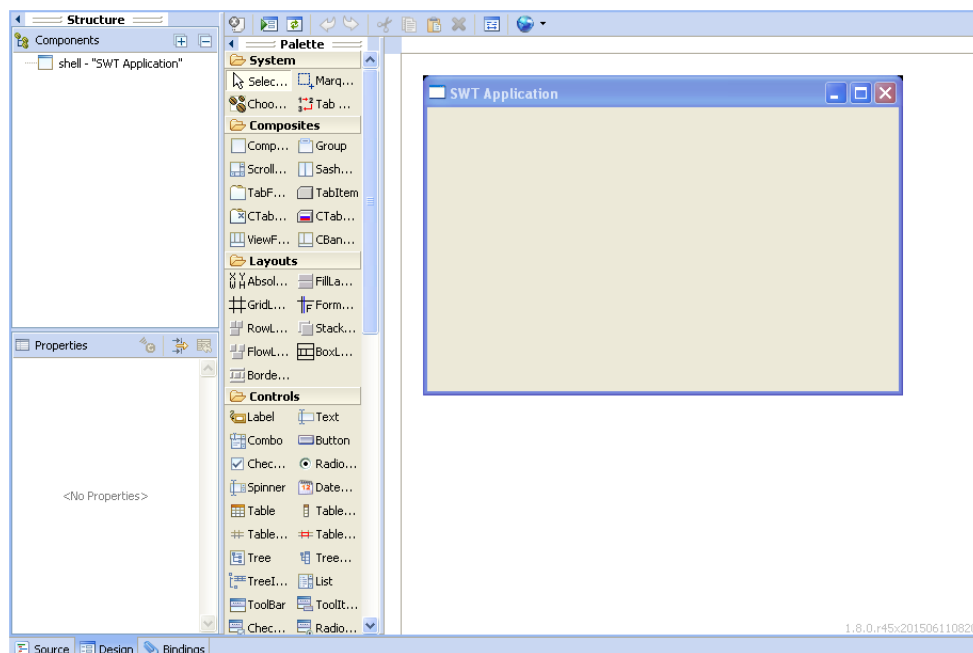
4.3. A la següent pantalla cal escriure: a l'apartat *Name*, el nom de la classe i, a l'apartat *Package*, el nom del paquet. Al nostre cas, la classe s'anomenarà *CalculaDoble* i el paquet, *wb*. La resta d'opcions poden deixar-se per defecte.



A continuació, farem clic a *Finish*.

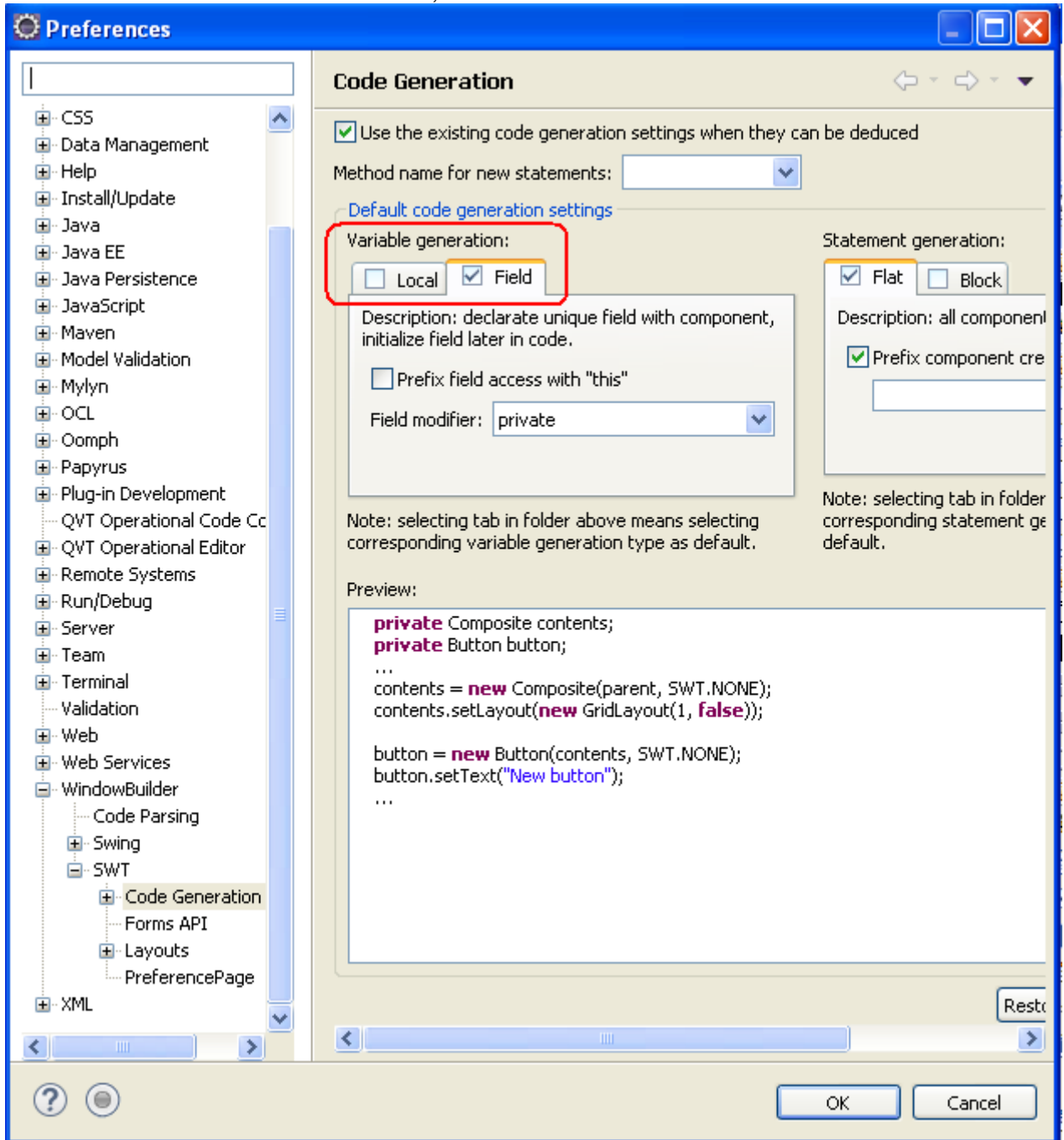
Ens apareix una finestra amb el codi generat. Ara només ens resta editar la interfície per incloure-hi el botó, els quadres de text i alguna etiqueta i definir el comportament del botó.




4.4. Seleccionem la pestanya *Design* (  Design ). Apareixerà una pantalla similar a la següent:

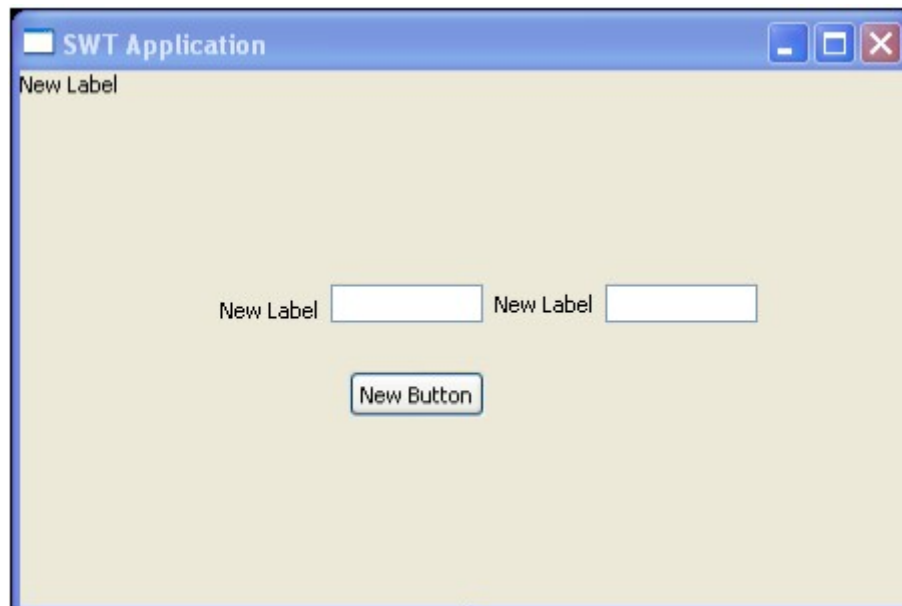


Ara caldrà anar situant els diferents elements sobre la finestra i canviant les seves característiques. Cadascun dels elements que afegim serà una variable del programa. Ens interessa que aquestes variables es defineixin com a camps de la classe. Per aconseguir-ho, abans de començar, cal fer el que ens indica el següent pas.

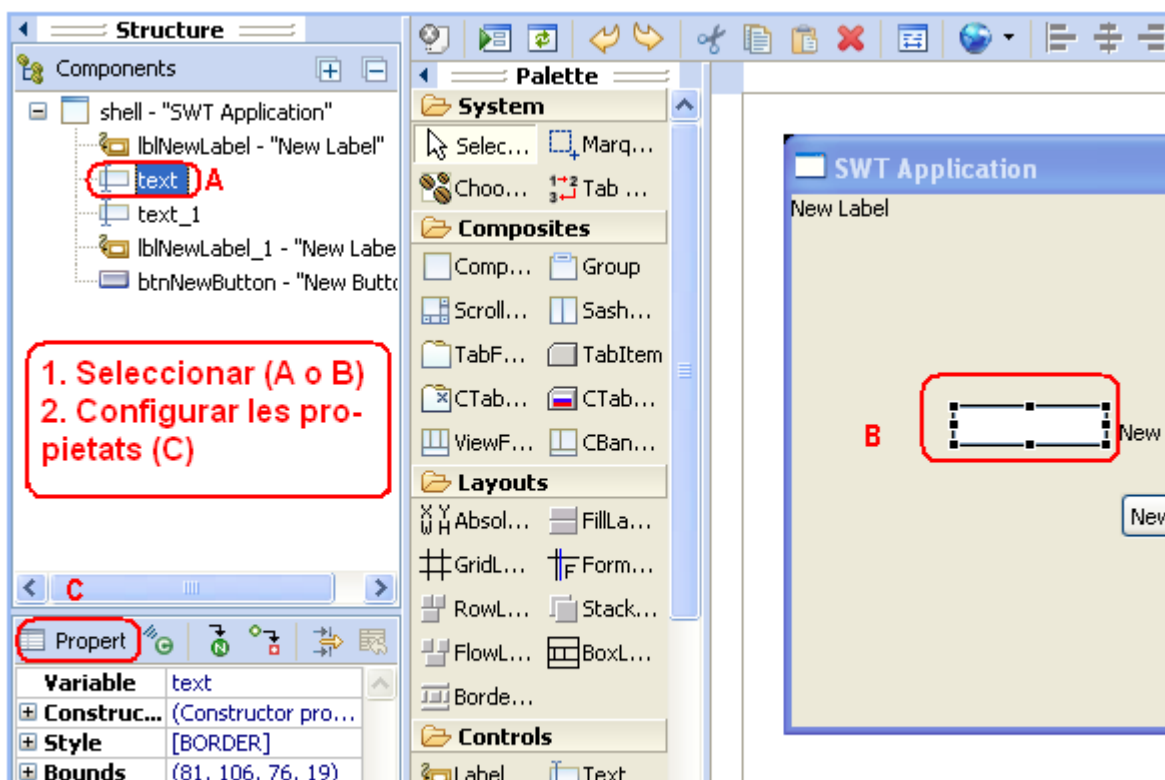
**4.5.** Seleccioneu l'opció del menú principal *Window* → *Preferences* i, a la finestra que apareix, a l'opció *WindowBuilder* → *SWT* → *Code Generation*, seleccioneu *Variable Generation* → *Field*.



4.6. Situeu 3 etiquetes (icona ) , dos quadres de text (icona ) i un botó (icona ) sobre la finestra perquè quedi de manera similar a la imatge que hi ha a continuació. Per fer-ho, només cal fer clic en primer lloc a la icona corresponent de la columna *Palette* i, a continuació, a sobre del punt de la finestra on volem situar l'element. Al següent pas, modificarem les propietats d'aquests elements.



4.7. Ara modificarem les propietats dels diferents elements. Per fer-ho, cal seleccionar l'element que volem modificar a la secció *Components* (A) o a la finestra de disseny (B) i actualitzar la propietat corresponent a la columna *Properties* (C)



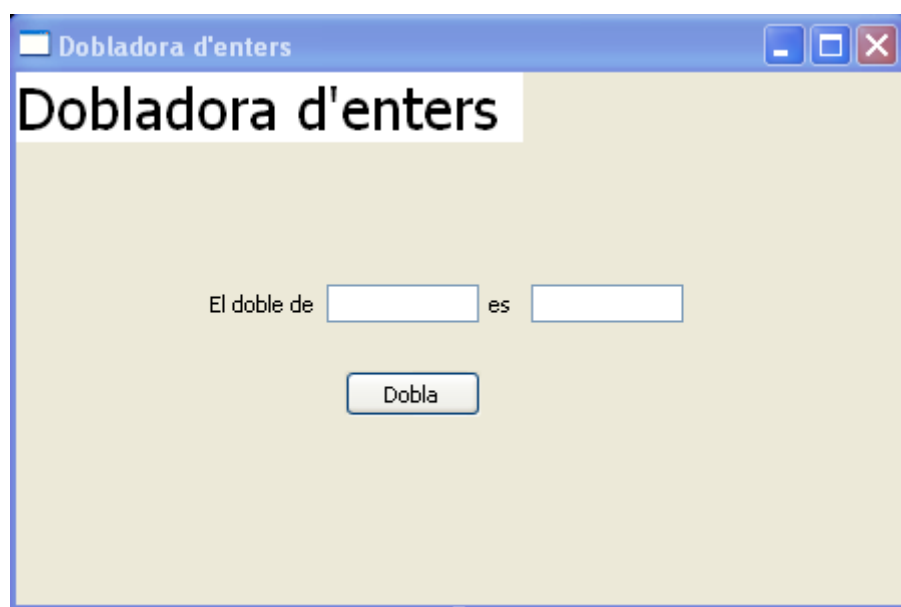
Concretament, les propietats que posarem seran (els requadres en gris indiquen que es deixen els valors per defecte):

	Finestra (variable shell)	Etiqueta superior	Etiqueta de l'esquerra	Etiqueta entre tots dos quadres de text	Primer quadre de text	Segon quadre de text	Botó
Variable (n'és el nom)		lblTitol	lblAbans	lblEnmig	textDada	textResultat	btnDobla
Text		Dobladora d'enters	El doble de	es			Dobla
		Font: Tahoma 22					
back ground		COLOR_WHITE				COLOR_WHITE	
editable						false	

Pot passar que algun canvi no es vegi immediatament. Si això passa, executeu l'aplicació com s'indica al darrer apartat i, a partir d'aquí, obtindreu una visualització correcta.

Segurament també caldrà canviar la posició i/o la grandària d'algun dels elements per aconseguir una visualització correcta. Es fa fent-hi clic a sobre i arrossegant l'element (per moure) o un dels quadrets negres que hi apareixen (per canviar la mida).

El resultat pot ser semblant a aquest:



### 5. Afegint comportament al botó

Si fem doble clic a sobre del botó ens apareixerà un fragment de codi semblant a aquest:

```
btnDobla.addSelectionListener(new SelectionAdapter() {  
    @Override  
    public void widgetSelected(SelectionEvent e) {  
    }  
});
```

El comportament desitjat per al botó el posarem dins del mètode `widgetSelected`.

#### Ampliació

La gestió del botó es fa a través d'esdeveniments (*events* en anglès) que es gestionen seguint el patró **observador**.

Aquest patró treballa amb dues menes d'objectes: *observadors* i *observats*. Els *observadors* estan pendents d'alguns dels esdeveniments que succeeixen en els *observats* i, en resposta a aquests esdeveniments, realitzen alguna acció. Els *observats* s'encarreguen d'avisar als *observadors* quan els succeeix un esdeveniment pel qual s' "ha interessat" prèviament l'*observador*. Cada *observat* enregistra amb un mètode els *observadors* que ha d'avisar.

En aquest cas, l'*observat* és el botó `btnDobla`. Quan cridem al mètode `addSelectionListener` el que fem és enregistrar-hi que ha d'avisar a l'objecte que li passem com a paràmetre perquè aquest pugui actuar davant els seus esdeveniments.

Però, quin objecte li passem com a paràmetre? El que creem a la instrucció

```
new SelectionAdapter() {  
    @Override  
    public void widgetSelected(SelectionEvent e) {  
  
        // codi que tracta l'esdeveniment  
  
    }  
}
```

`SelectionAdapter` és una classe que implementa un observador de determinats esdeveniments, entre ells fer clic a un botó.

En fer `new SelectionAdapter() {...}` i sobrescriure el mètode `widgetSelected`, definim a la vegada una subclasse de `SelectionAdapter` i un objecte d'aquesta subclasse. Aquest objecte farà el tractament desitjat i és el que passem com a paràmetre a `addSelectionListener`. El tractament que farà quedarà detallat dins del mètode `widgetSelected` (la classe `SelectionAdapter` té aquesta especificació).

Veureu també que no posem nom a la subclasse. És **anònima**. Això és coherent perquè aquesta classe només s'utilitzarà aquí.

El comportament el posarem dins del mètode `widgetSelected`.

Pot quedar així:

```
btnDobla.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        int valorInicial;
        try{
            valorInicial=new Integer(textDada.getText());
        }catch (NumberFormatException ex){
            JOptionPane.showMessageDialog(null,
                "Enter erroni", "Avis", JOptionPane.WARNING_MESSAGE);
            return;
        }

        textResultat.setText(""+(valorInicial*2));
    }
});
```

La instrucció `new Integer(textDada.getText())` converteix el text del quadre *textDada* en un enter. El resultat s'assigna a la variable entera *valorInicial*. Per accedir al contingut del primer quadre de text posem el nom de la variable (*textDada*) i cridem al mètode dels quadres de text *getText*, que retorna el text que conté el quadre.

L'assignació és dins d'un *try/catch* per a capturar l'excepció que es produeix quan l'enter no té un format correcte. El tractament es fa dins del *catch*.

Dins del *catch* (s'hi entra quan hi ha un error al format d'entrada), es llença un avís i s'acaba el mètode (no cal fer res). El missatge es mostra amb el mètode *JOptionPane.showMessageDialog*. La classe *JOptionPane* és una classe estàndard de Java que permet mostrar diferents tipus de diàleg per pantalla. En el cas de *showMessageDialog* es mostra un missatge amb un text i només un botó per tancar el diàleg. Els paràmetres són, en aquest ordre:

- El formulari pare. En aquest cas, podem posar *null*.
- El missatge que volem mostrar.
- El títol que sortirà a la part de dalt de la finestra.
- El tipus de missatge. En aquest cas, és un missatge d'avís.

Després del *catch* (el codi s'executa quan el contingut del primer quadre de text és correcte), s'actualitza el text de la variable *textResultat* (és el segon quadre de text) amb el resultat obtingut. Per fer-ho s'utilitza el mètode *setText*. Es concatena el resultat amb la cadena buida perquè fer-ho és una manera ràpida de convertir un enter en una cadena de caràcters.

El connector *WindowBuilder* utilitza la biblioteca SWT. Podeu trobar la seva documentació aquí: <https://www.eclipse.org/swt/javadoc.php>. Trobareu especialment útil la documentació del paquet *org.eclipse.swt.widgets*.

A l'enllaç <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html> podeu trobar la documentació associada a la classe *JOptionPane*. Veureu que permet realitzar de manera molt senzilla diferents tipus de diàlegs.

## 6. Execució

Cal fer clic amb el botó secundari del ratolí (botó dret en el cas dels destres) a sobre del fitxer *CalculaDoble* i seleccionar l'opció *Run As → Java Application*.

