

Tutorial de développement par l'exemple

Développement d'une application de gestion de compte

Table des matières

1 Objectifs de l'application.....	3
2 Création des tables.....	3
2.1 les tables.....	3
2.2 Les colonnes.....	4
3 Développement avec Devprogi Application.....	7
3.1 Lancement de l'application & connexion.....	7
3.2 Création des tables sous Devprogi Application.....	7
3.3 Création d'écran sous Devprogi Application.....	10
1 Évolution fonctionnelle de l'application.....	30
1.1 Ajout de listes de valeurs.....	30
1.2 Modification de la police.....	31
1.3 Modification de la longueur des champs.....	31

Vous apprendrez avec ce document à utiliser Devprogi Application à partir d'un exemple concret.

Le développement de cette application nous a permis de tester Devprogi Application et de déboguer de nombreux points. De nombreuses fonctionnalités restent encore à développer ou à faire évoluer. L'application de gestion résultante restera cependant basic.

1 Objectifs de l'application

Pour commencer, nous allons déterminer les différents objectifs auquel devra répondre notre application.

Une application de gestion doit permettre d'enregistrer les différentes opérations du jour et de connaître le solde qui en résulte. Lors de la saisie d'une opération, nous saisirons une date d'opération, le montant, le compte sur lequel est imputé l'opération, le mode de paiement (Carte bancaire, prélèvement, virement,...), le tiers débiteur ou créateur, le type de dépense et éventuellement un commentaire. Nous optons pour deux écrans :

- un écran permettant la saisie de l'opération
- un écran permettant de visualiser les différentes opérations saisies avec le solde final.

Par ailleurs, dans l'écran de visualisation des mouvements, nous devons intégrer l'information « opérations pointées/non pointées »

L'application doit par ailleurs permettre de paramétrer les différentes informations saisies. Nous aurons donc quelques écrans supplémentaires :

- écran de saisie des tiers
- écran de saisie des comptes
- écran de saisie des types de dépenses
- et enfin un écran de saisie des modes de paiement.

2 Création des tables

2.1 *les tables*

Nous aurons donc en premier lieu à créer les tables :

Description	Nom des tables
Table des comptes	TAB_ACCOUNT
Table des tiers	TAB_TIERS
Table des modes de paiement	TAB_MODE_PAIE
Table des types de dépenses	TAB_CATEGORIE
Tables des mouvements	TAB_MVT

2.2 Les colonnes

Nous allons maintenant définir les différentes colonnes pour chacune des tables

Table des comptes : TAB_ACCOUNT

Un compte est constitué d'un code banque, d'un code guichet, d'un numéro de compte et d'une clé RIB permettant de valider le numéro de compte. Par ailleurs, on peut aussi attribuer un intitulé au numéro de compte. Ensuite, il est aussi utile de définir le type de compte. Il peut s'agir d'un compte épargne, ou d'un compte courant par exemple. Ensuite, à la création de compte, il faut initialiser le montant initial.

Colonnes	Description	Type	Taille
ACC_ID	Identifiant compte	Numérique	
ACC_NAME	Nom du compte	Texte	50
ACC_TYPE	Type de compte	Texte	4
ACC_INI_AMOUNT	Solde initial	Numérique	10
ACC_CD_BANK	Code banque	Texte	5
ACC_CD_SORT	Code guichet	Texte	5
ACC_NUM	Numéro de compte	Texte	11
ACC_KEY	Clé RIB	Texte	2
ACC_BANK_NAME	Nom de la banque	Texte	100

Dans cette version simplifiée, nous n'effectuerons pas de contrôle sur la saisie du compte bancaire.

Table des tiers : TAB_TIERS

Colonnes	Description	Type	Taille
TIERS_ID	Identifiant tiers	Numérique	
TIERS_LAST_NAME	prénom	Texte	100
TIERS_FIRST_NAME	nom	Texte	100
TIERS_COMPLET_NAME	Nom complet ou raison social	Texte	200
TIERS_TYPE	Type tiers (part., Société,...)	Texte	100
TIERS_COMMENT	Commentaire	Texte	1000
TIERS_ADDR_LINE	Ligne d'adresse	Texte	1000
TIERS_CITY	Ville	Texte	200
TIERS_CAT	Catégories	Texte	

Table des modes de paiement : TAB_MODE_PAIE

Colonnes	Description	Type	Taille
MP_ID	Identifiant mode de paiement	Numérique	
MP_NAME	Nom du type de paiement	Texte	100
MP_CODE	Code type de paiement	Texte	4
MP_DESC	Description	Texte	1000

Table des types de catégorie de dépenses : TAB_CATEGORIE

Colonnes	Description	Type	Taille
TYPE_CAT_ID	Identifiant	Numérique	
TYPE_CAT_NAME	Nom de catégories de depense	Texte	100
TYPE_CAT_CODE	Code type de dépense	Texte	4
TYPE_CAT_DESC	Description	Texte	1000

Tables des mouvements : TAB_MVT

Colonnes	Description	Type	Taille
MVT_ID	Identifiant	Numérique	
MVT_AMOUNT	Nom du type de depense	Numérique	10,2

Colonnes	Description	Type	Taille
MVT_CAT	Type de dépense	Texte	4
MVT_MODE_PAIE	Mode de paiement	Texte	4
MVT_DATE	Date de mouvement	Date	
MVT_COMPTA	Si comptabilisé ou non	booleen	

3 Développement avec Devprogi Application

3.1 Lancement de l'application & connexion

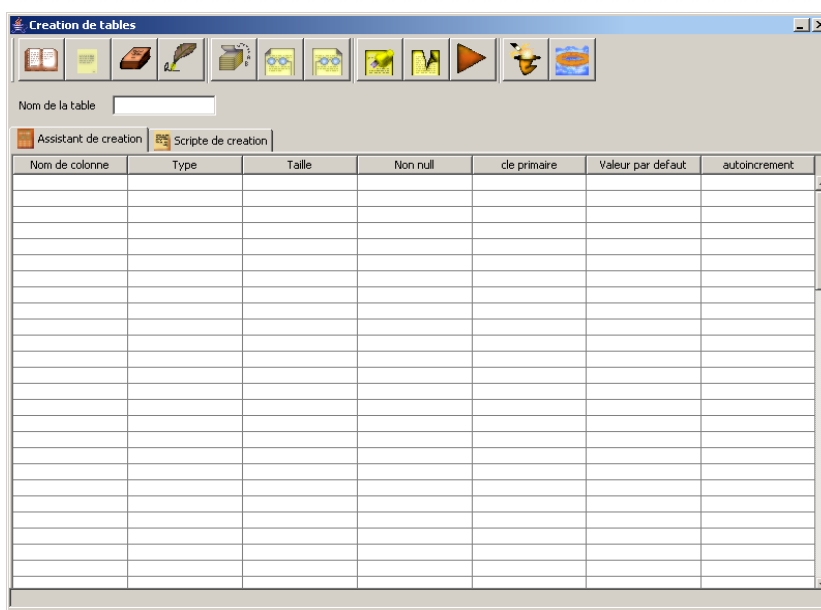
Après le lancement de l'application, on propose de suite la connexion en tant qu'administrateur (tout au moins tant que la version Devprogi Application sera une version Beta).



Après ouverture de la fenêtre MDI, nous allons maintenant créer les tables. Pour se faire, nous utiliserons l'écran d'assistance à la création des tables.

3.2 Création des tables sous Devprogi Application

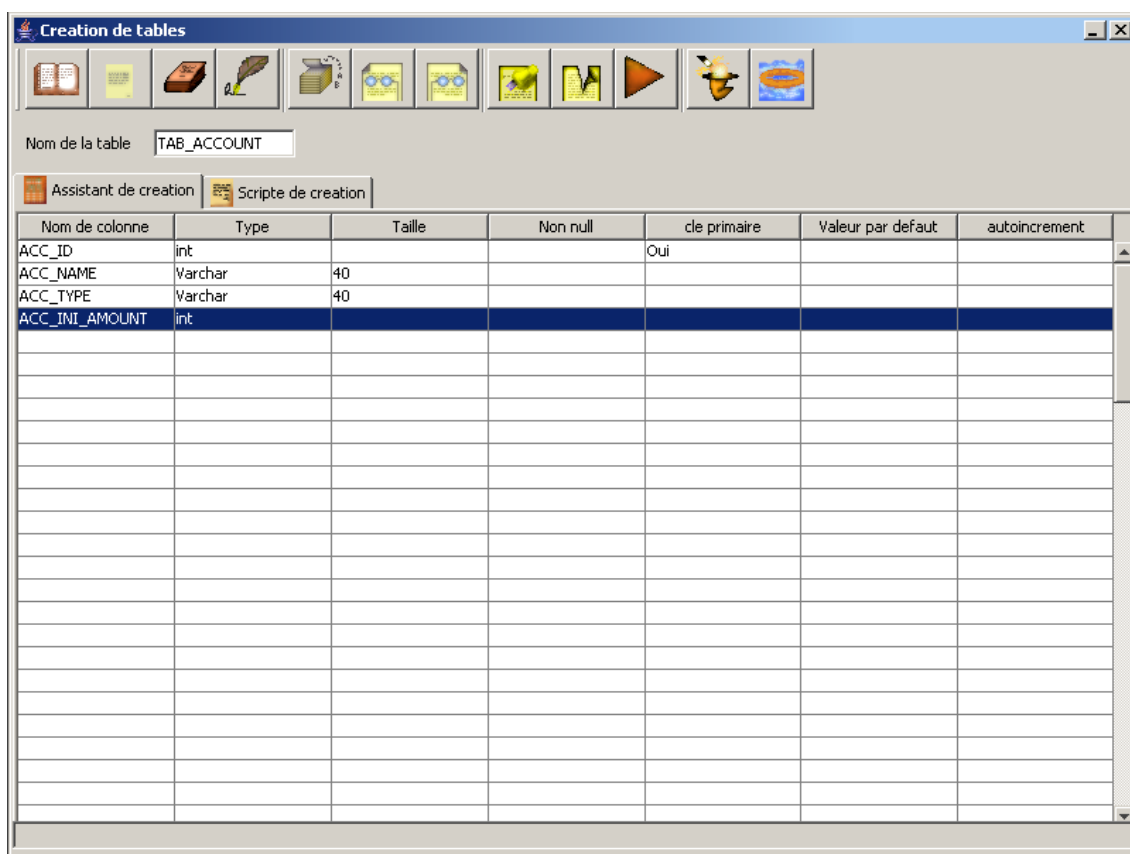
Menu développeur --> Créer une table



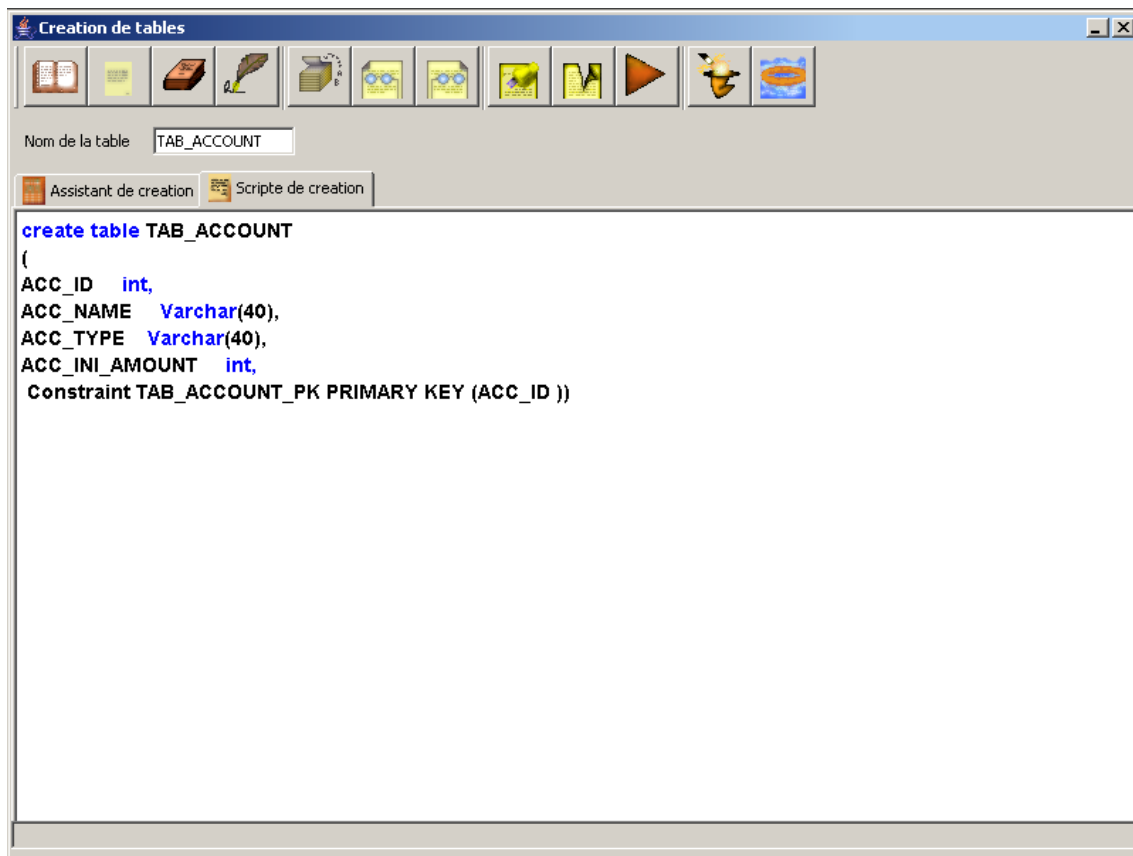
Remarque

L'écran ne permet actuellement que de créer une table mais ne peut éditer et modifier une table. C'est dans le cadre d'un futur développement que je rendrais possible ces dernières fonctionnalités. Si vous devez modifier une table, vous devez la supprimer puis ouvrir le script de création et modifier la table et enfin exécuter le script.

Il est clair que c'est un développement prioritaire sachant que supprimer la table est plutôt une aberration surtout si des données sont présentes.



Vous pouvez voir le script de création sous l'onglet « Scripte de creation ».



Sauvegarder le script en appuyant sur le bouton avant d'exécuter la création de la table. (C'est un point qui devrait être revu dans les prochains développements)



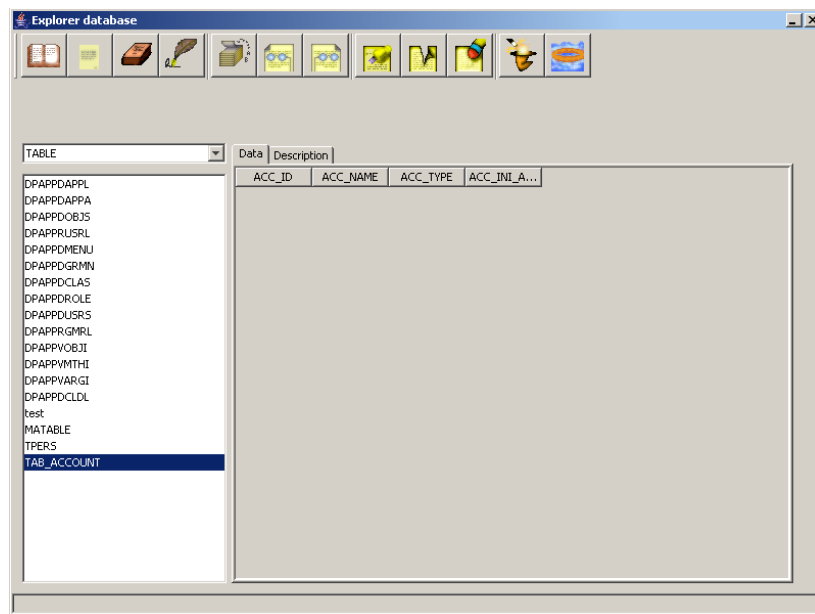
Pour lancer la création de la table, vous n'avez plus qu'à cliquer sur le bouton



Pour vérifier que la création s'est bien effectuée, vous pouvez aller dans le menu « Menu développeur/Browser Database ». **Il sera aussi spécifié dans le prochain développement un message de confirmation.**

Au moment où ce document est écrit, nous apportons certaines modifications. Par

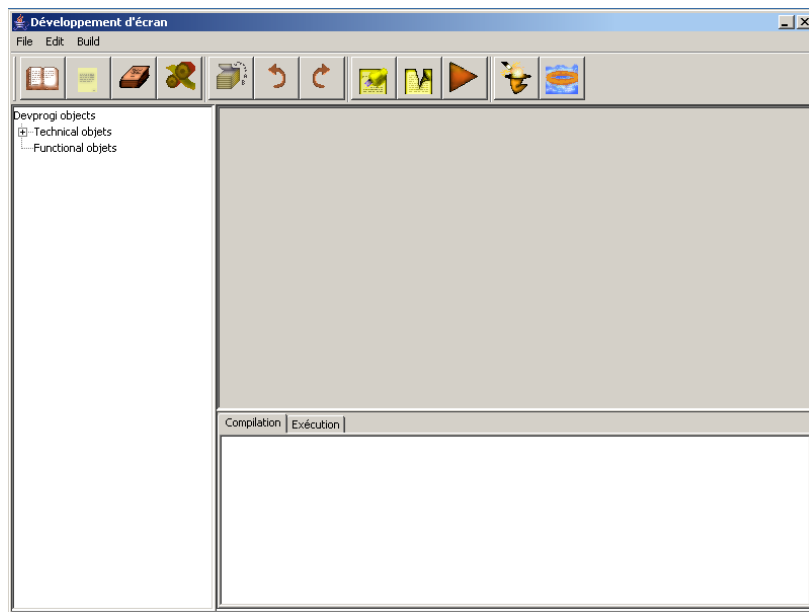
exemple, il n'y a pas encore la notion de projets. De plus, il n'y a pas de répertoire sql configuré comme pour les classes et les sources java.



3.3 Création d'écran sous Devprogi Application

Nous allons maintenant créer les écrans. Pour commencer, nous allons ouvrir l'éditeur de classes.

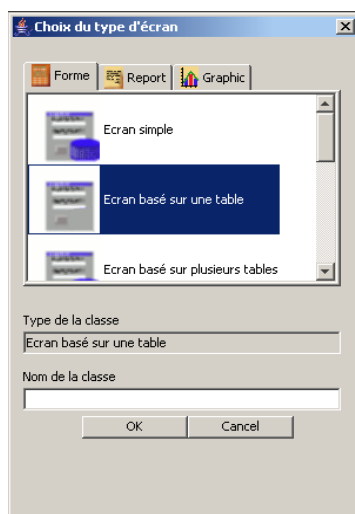
Menu développeur --> Editeur



La plupart des icones dans l'application sont tous identiques dans les écrans et ont une signification. Cliquez sur le bouton « nouveau ».



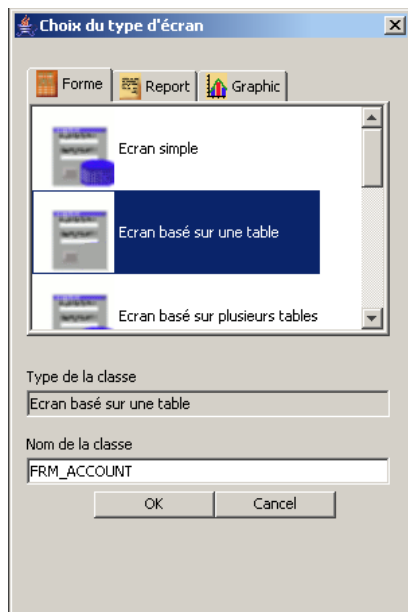
Nouvelle classe

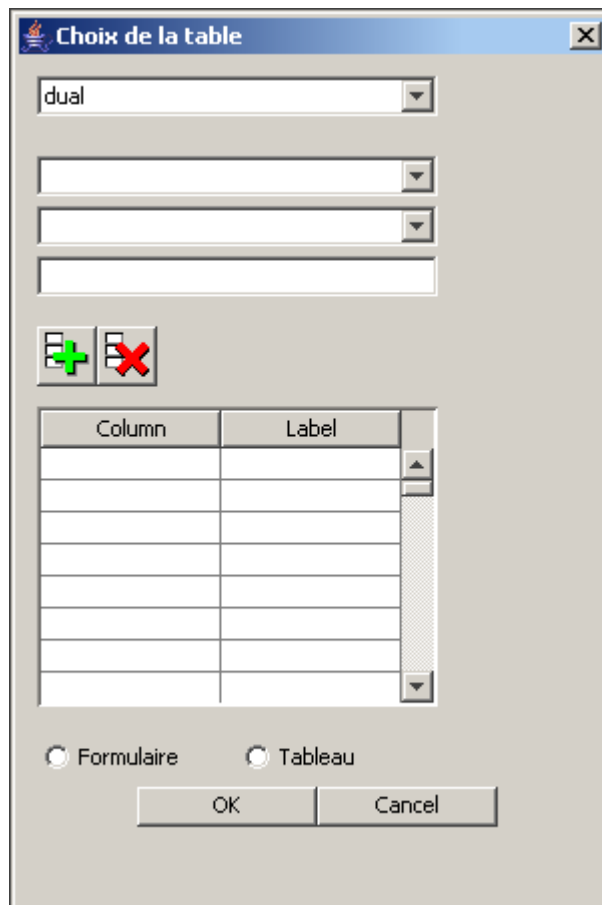


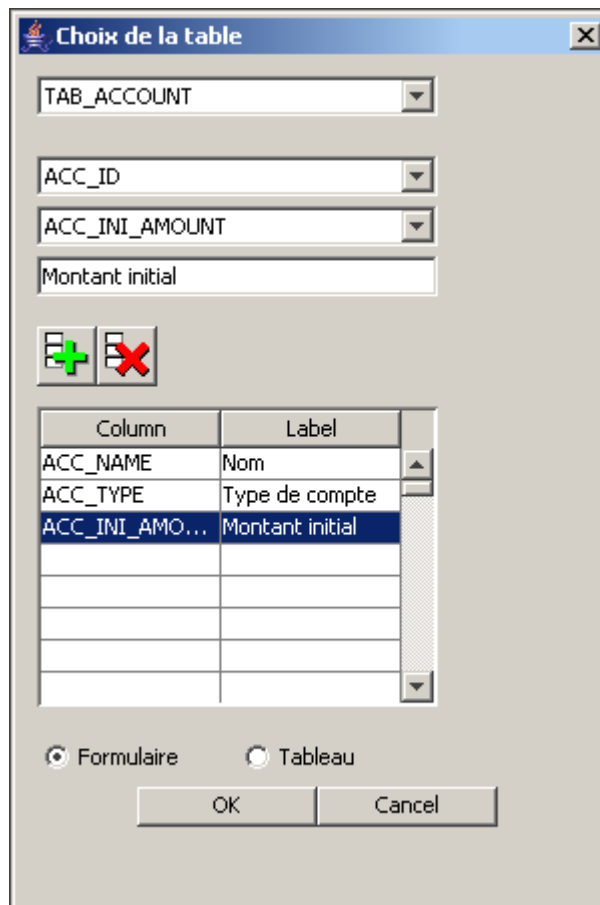
Un écran est alors ouvert et vous permet de choisir un template.

Remarque

Actuellement, les templates sont « figés » dans l'application. Une future évolution devrait permettre de pouvoir créer et enregistrer des templates extérieurs.



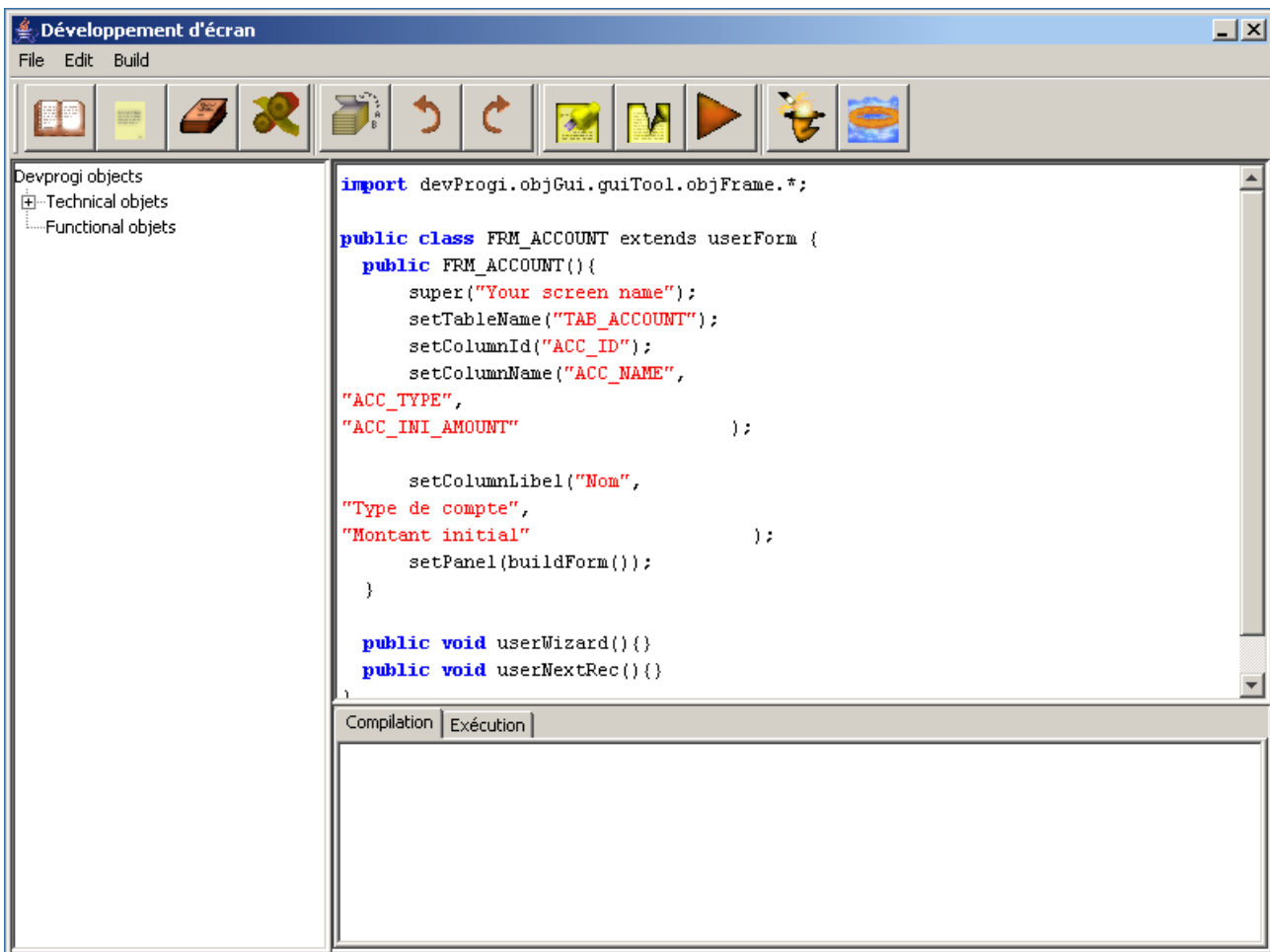




possibilité de saisir de différentes façon

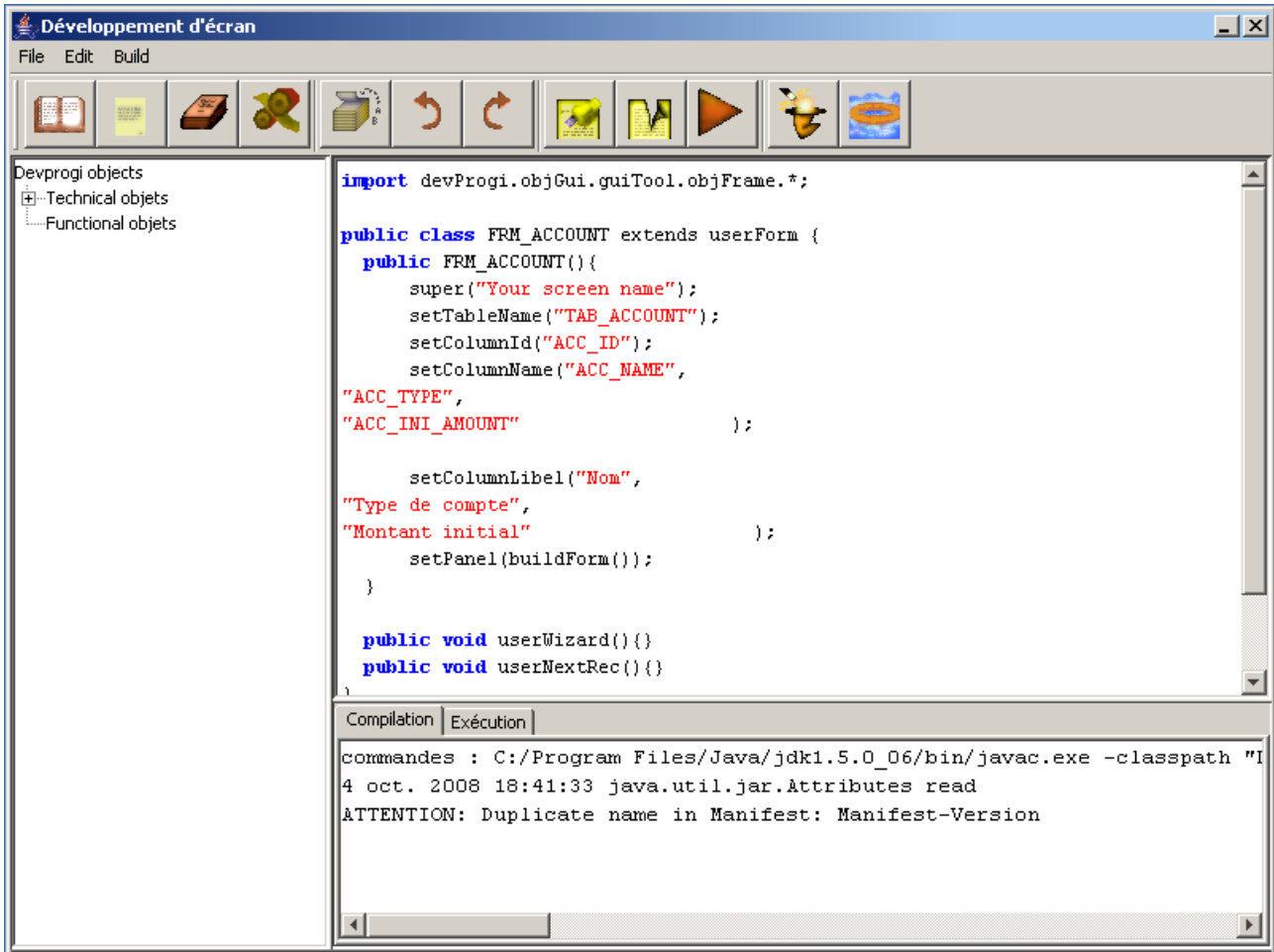
Remarque

Il est évident que l'objectif final est de pouvoir construire l'écran de façon graphique. C'est l'étude de future développements.



Vous pouvez lancer la compilation dans un premier temps sans retoucher le code.





The screenshot shows the 'Développement d'écran' IDE interface. On the left is a project tree with 'Devprogi objects', 'Technical objets', and 'Functional objets'. The main editor displays the following Java code:

```
import devProgi.objGui.guiTool.objFrame.*;

public class FRM_ACCOUNT extends userForm {
    public FRM_ACCOUNT(){
        super("Your screen name");
        setTableName("TAB_ACCOUNT");
        setColumnId("ACC_ID");
        setColumnName("ACC_NAME",
"ACC_TYPE",
"ACC_INI_AMOUNT"           );
        setColumnLibel("Nom",
"Type de compte",
"Montant initial"         );
        setPanel(buildForm());
    }

    public void userWizard(){}
    public void userNextRec(){}
}
```

Below the code editor is a console window with the following output:

```
Compilation | Exécution
commandes : C:/Program Files/Java/jdk1.5.0_06/bin/javac.exe -classpath "I
4 oct. 2008 18:41:33 java.util.jar.Attributes read
ATTENTION: Duplicate name in Manifest: Manifest-Version
```

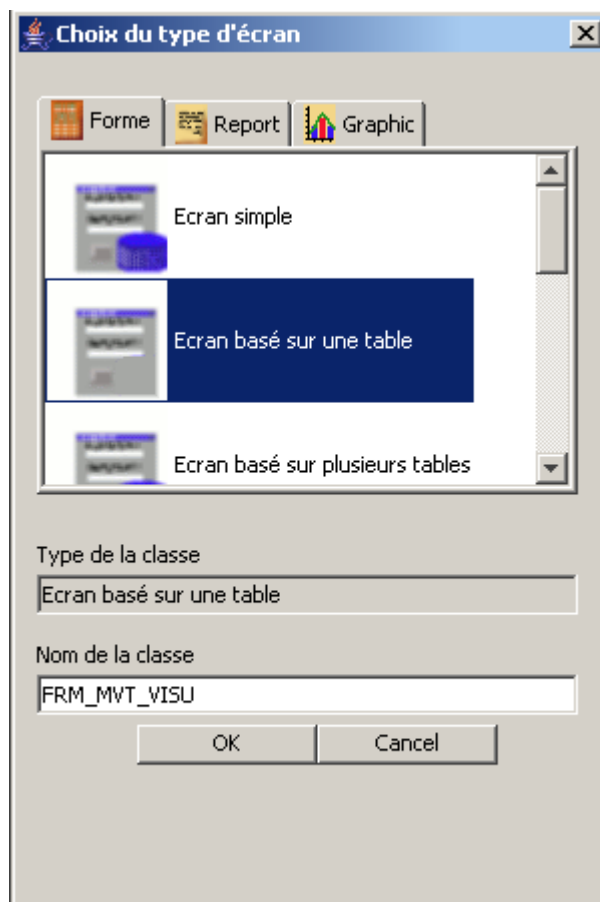


The screenshot shows a web browser window with the title "Your screen name". The browser's address bar and toolbar are visible at the top. The main content area of the browser displays a registration form with the following elements:

- A label "Nom" followed by a text input field.
- A label "Type de compte" followed by a text input field.
- A label "Montant initial" followed by a text input field.

A ce stade, nous nous préoccupons pas de la présentation. Nous verrons un peu plus loin comment mieux présenter l'écran.

Nous allons par ailleurs faire un écran afin de visualiser les opérations et de comptabiliser les opérations.



Choix de la table

TAB_MVT

MVT_ID

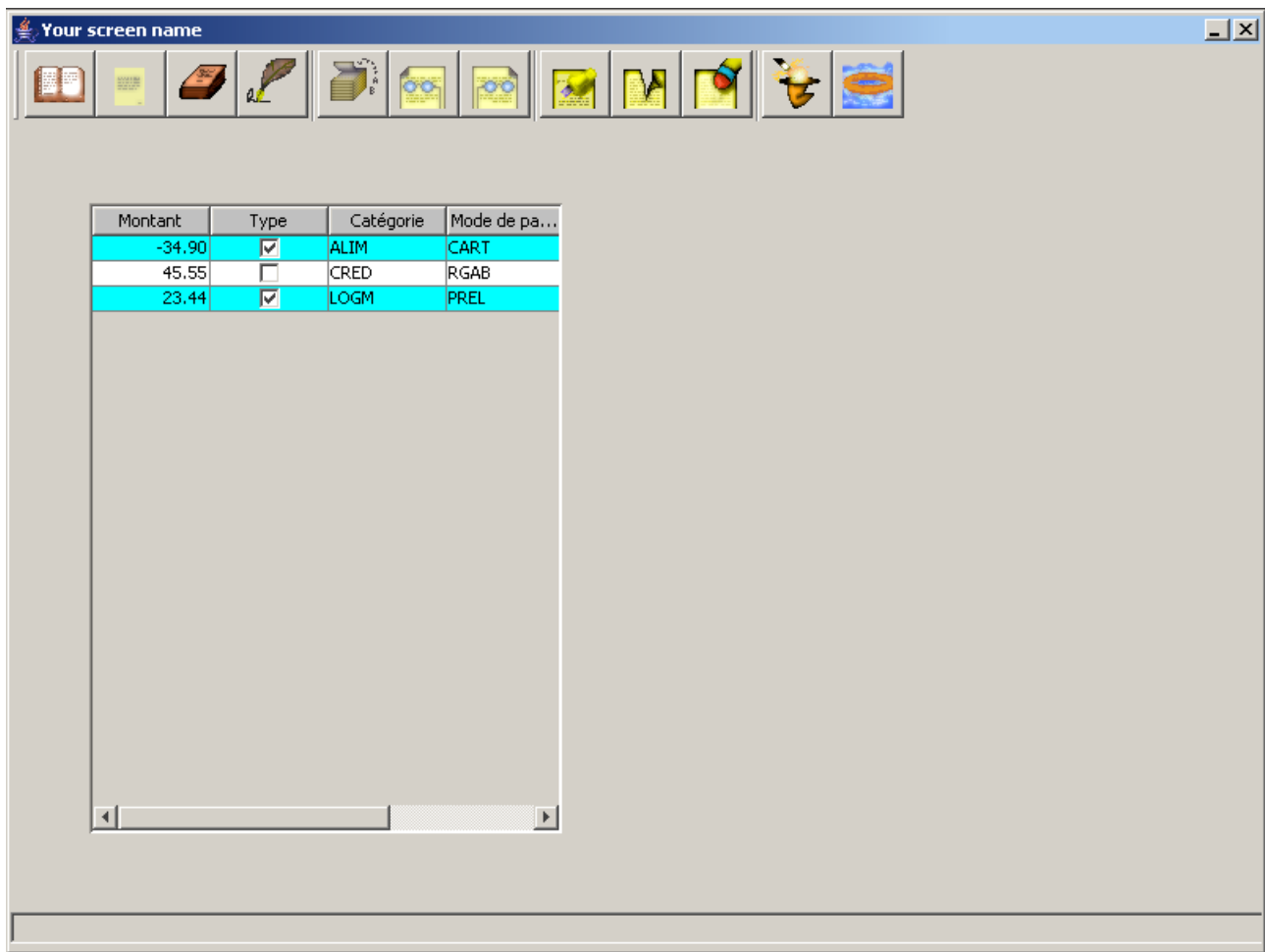
MVT_COMPTA

+ -

Column	Label
MVT_AMOUNT	Montant
MVT_TYPE	Type
MVT_TYPE_DEP	Catégorie
MVT_MODE_PAIE	Mode de paiem...
MVT_DATE	Date d'opération
MVT_COMPTA	Pointer

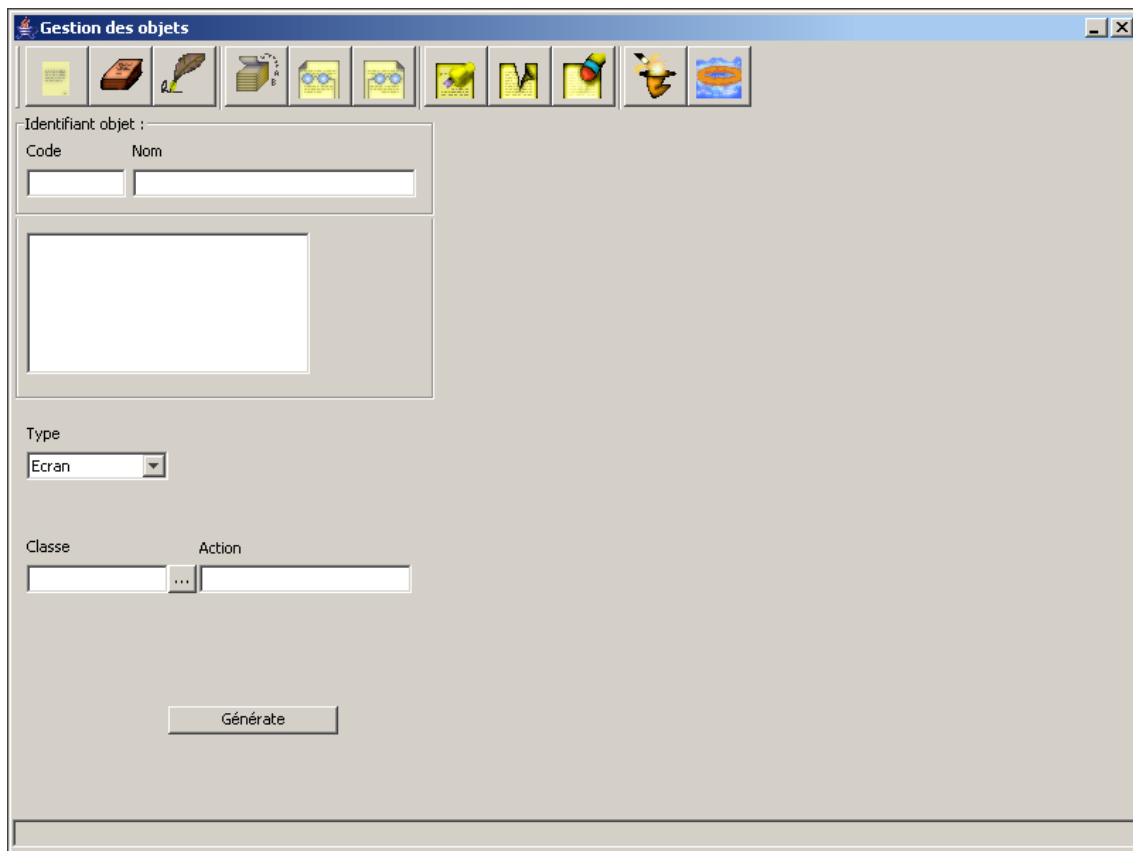
Formulaire Tableau

OK Cancel



A ce stade, nous pouvons commencer à configurer l'application afin de faire apparaître dans le menu les différents écrans.

Menu administrateur / Gestion des objets

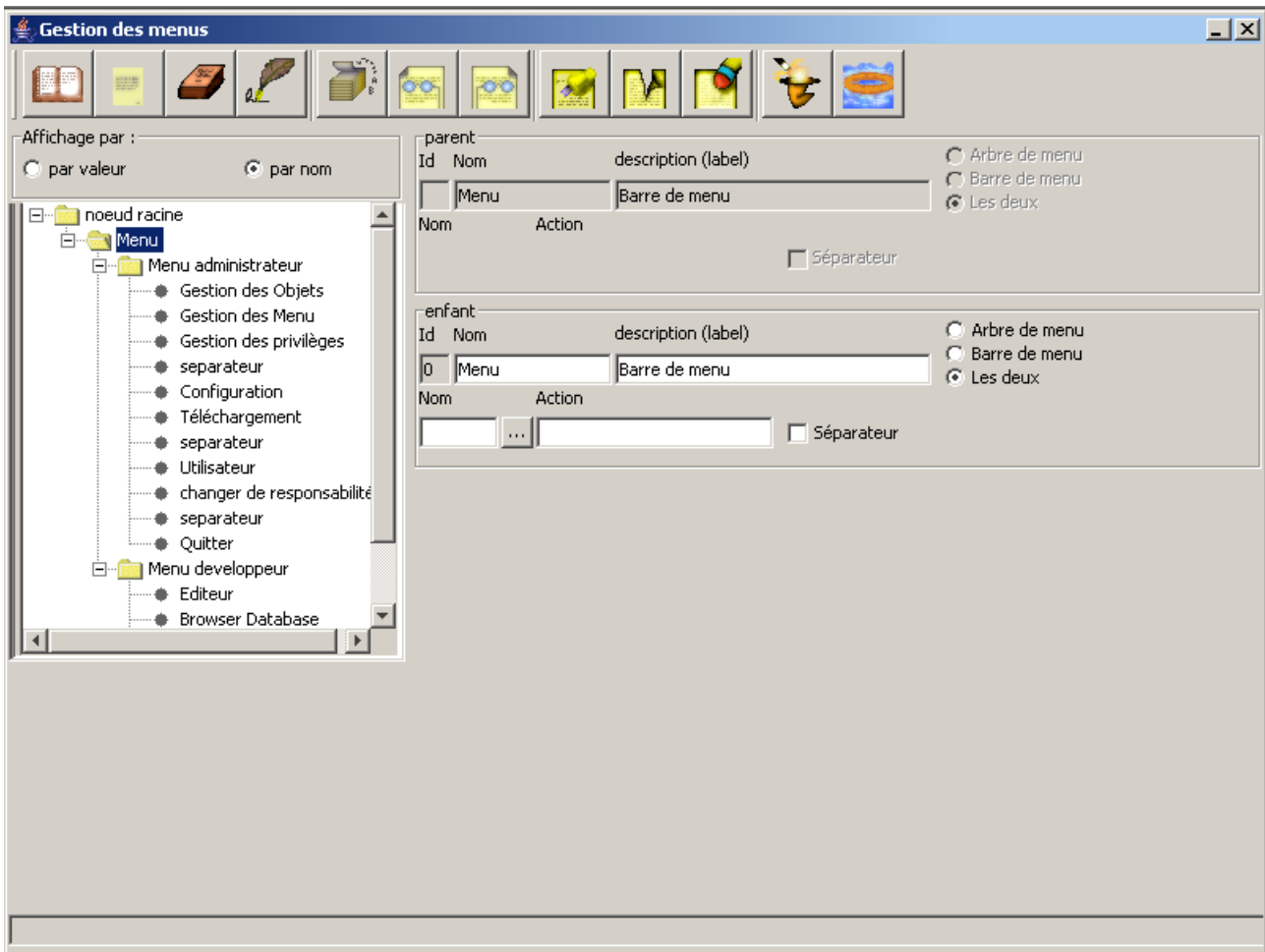


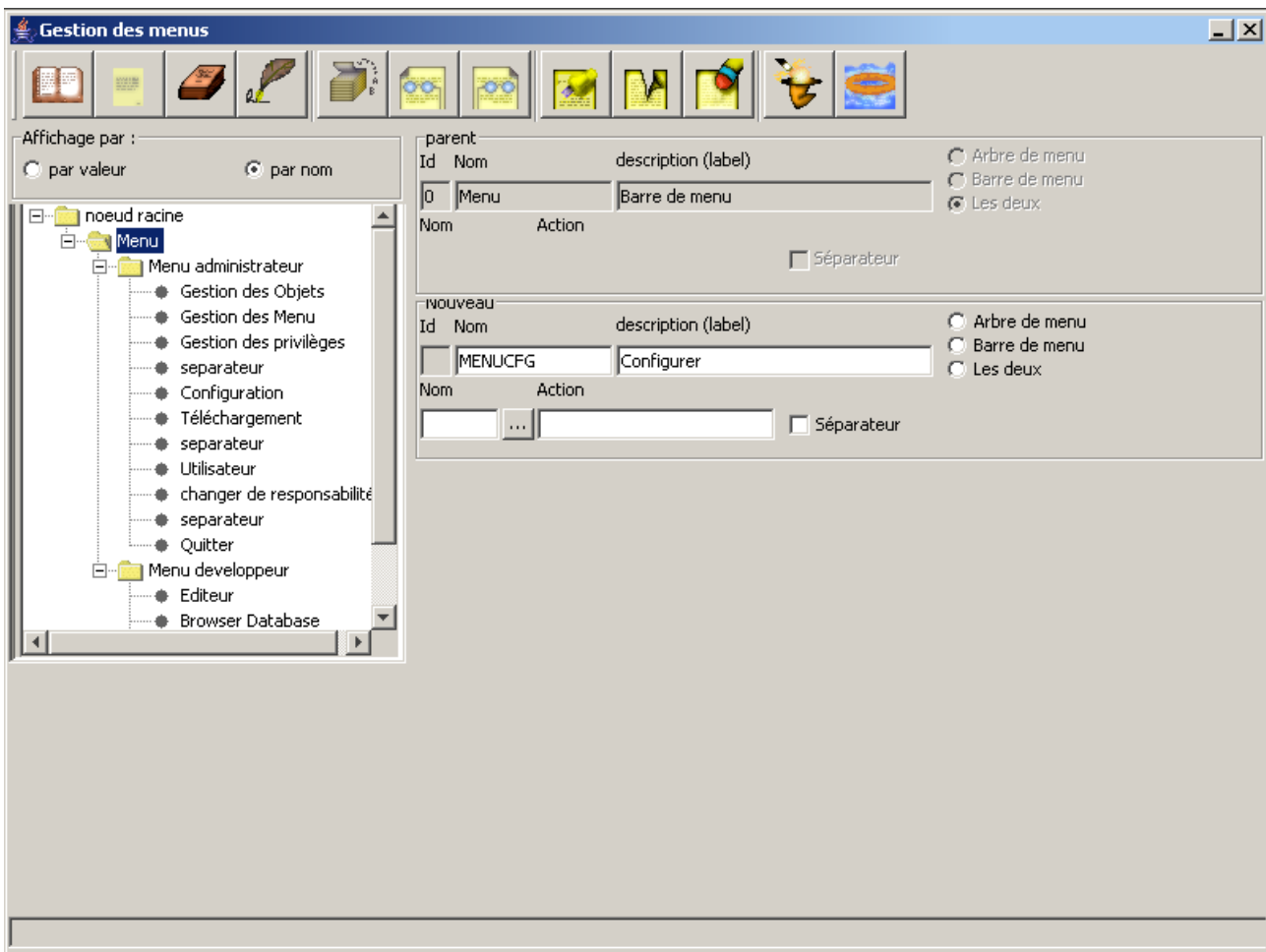
Attention, on ne génère l'action associée une fois après avoir sauvegarder. Si vous lancez la génération de l'action sans avoir sauvegarder, vous verrez apparaître un fichier action0.java dans le répertoire source.

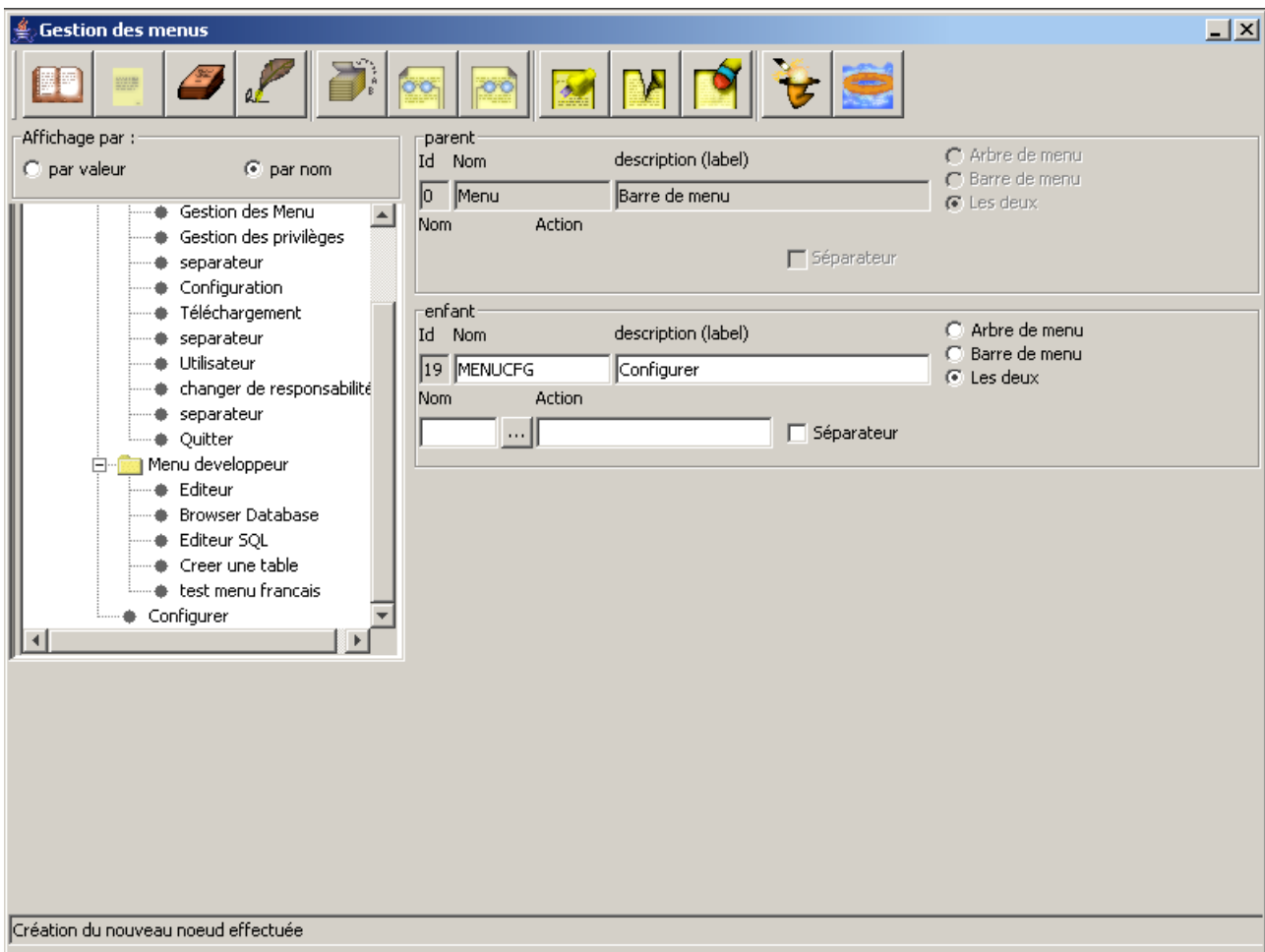
Dans de future développement, il est prévu qu'un même objet serve plusieurs fois (suivant les paramètres passés).

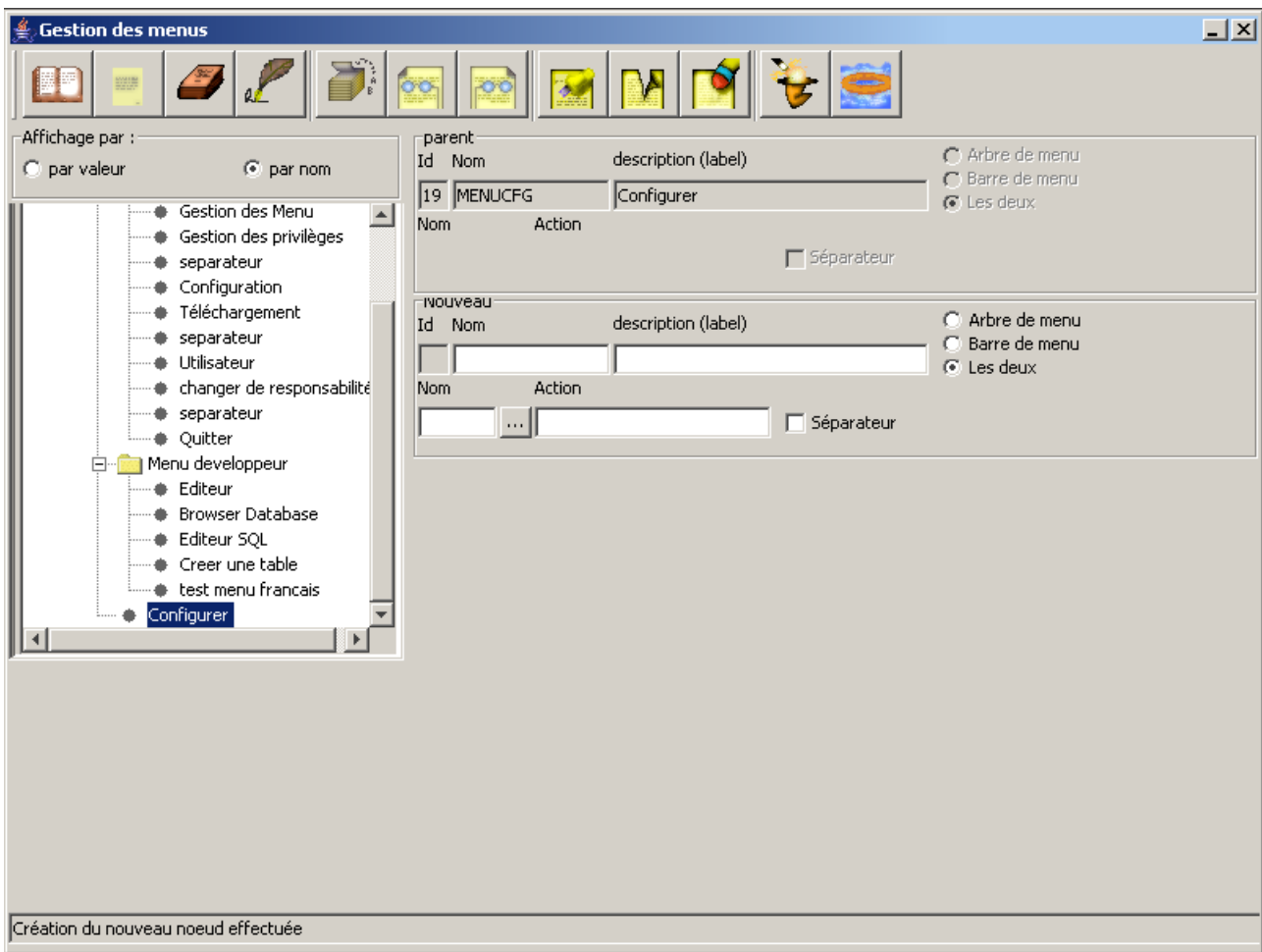
The screenshot shows a window titled "Gestion des objets" with a toolbar at the top. Below the toolbar, there are two input fields for "Identifiant objet": "Code" and "Nom", both of which are empty. A large text area for "Description" is also empty. The "Type" dropdown menu is set to "Ecran". The "Classe" and "Action" fields are empty, with a magnifying glass icon between them. A "Générer" button is located at the bottom.

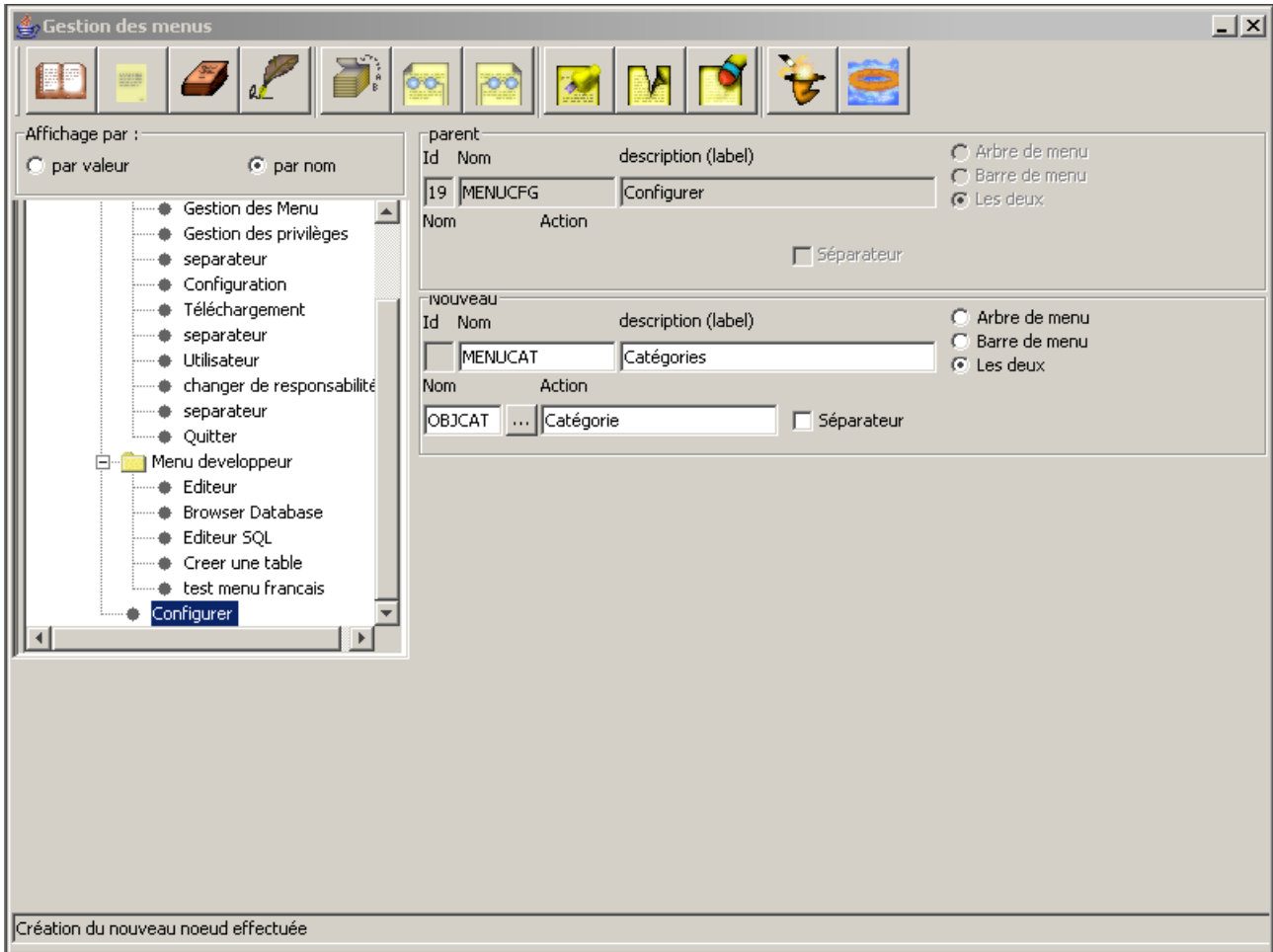
The screenshot shows the same "Gestion des objets" window, but with data entered. The "Code" field contains "OBJCAT" and the "Nom" field contains "FRM_CATEGORIE". The "Description" text area contains the text "Ecran de configuration des catégories". The "Type" dropdown menu remains set to "Ecran". The "Classe" field now contains "FRM_CATEGORIE", and the "Action" field is empty. The magnifying glass icon is still present between the "Classe" and "Action" fields. The "Générer" button is at the bottom.

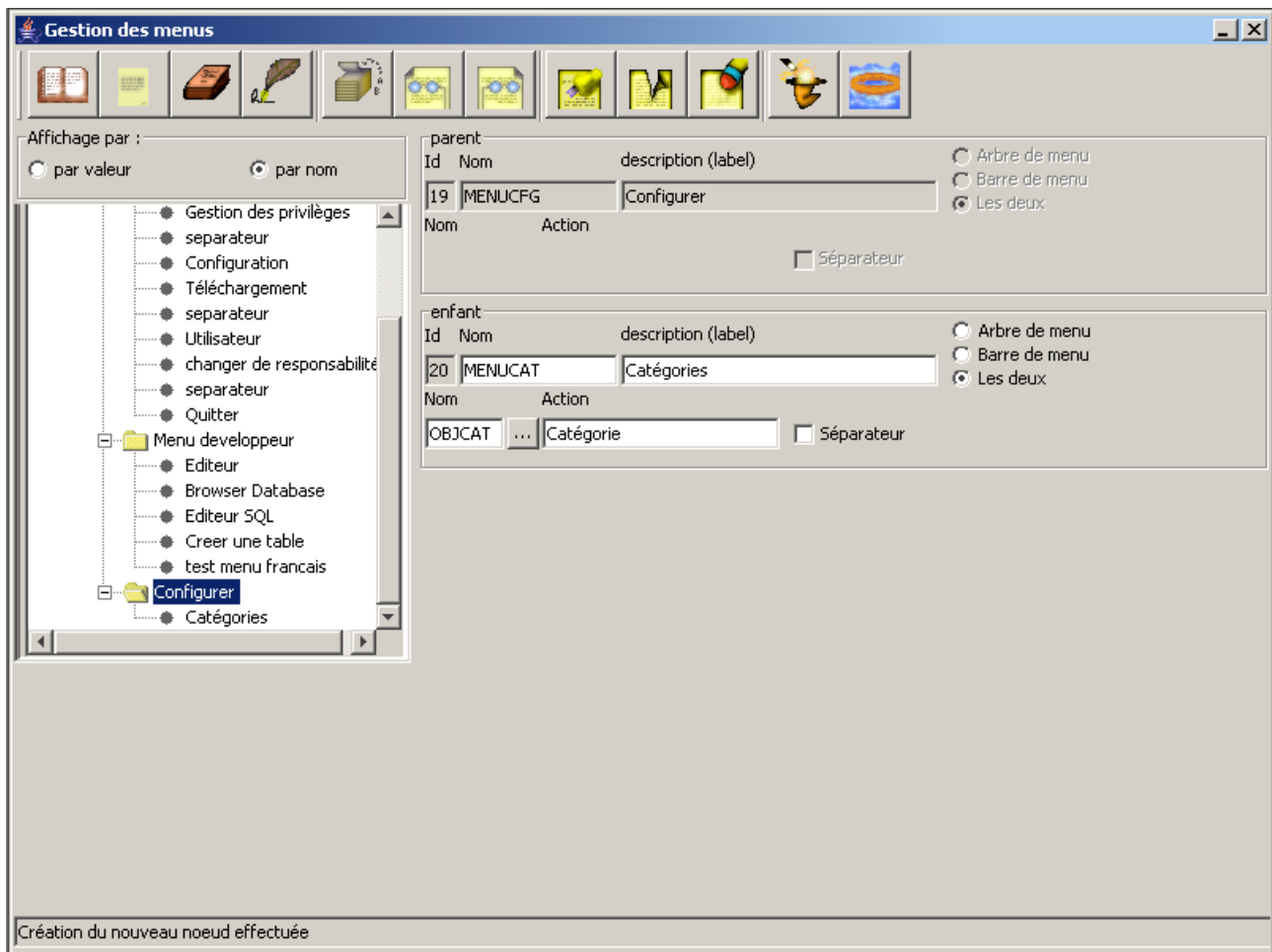








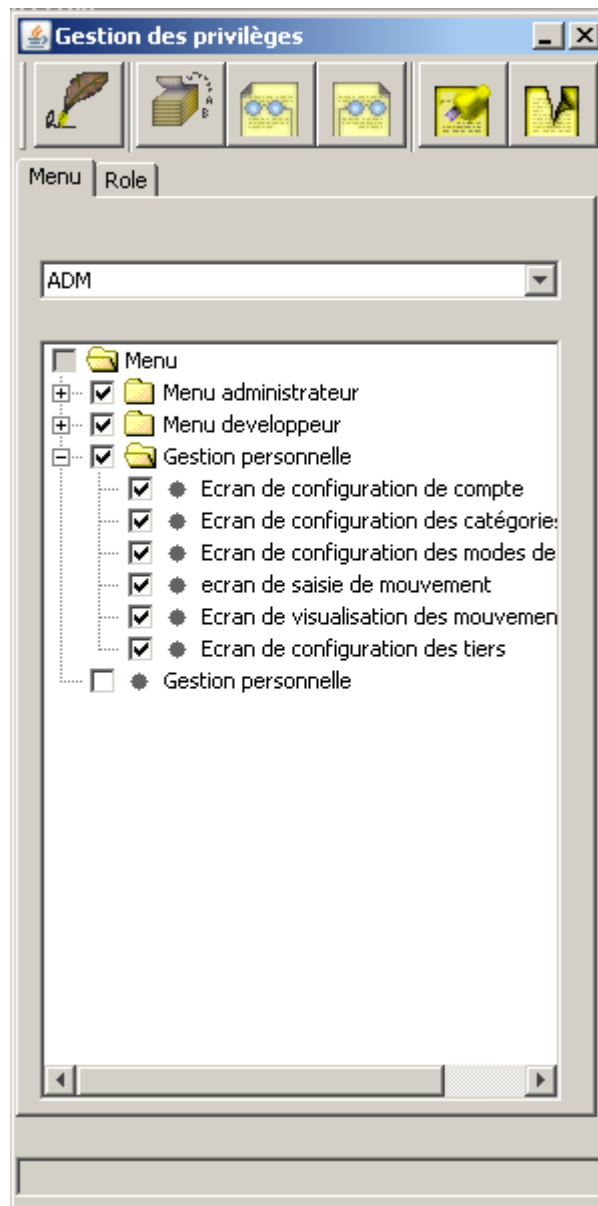




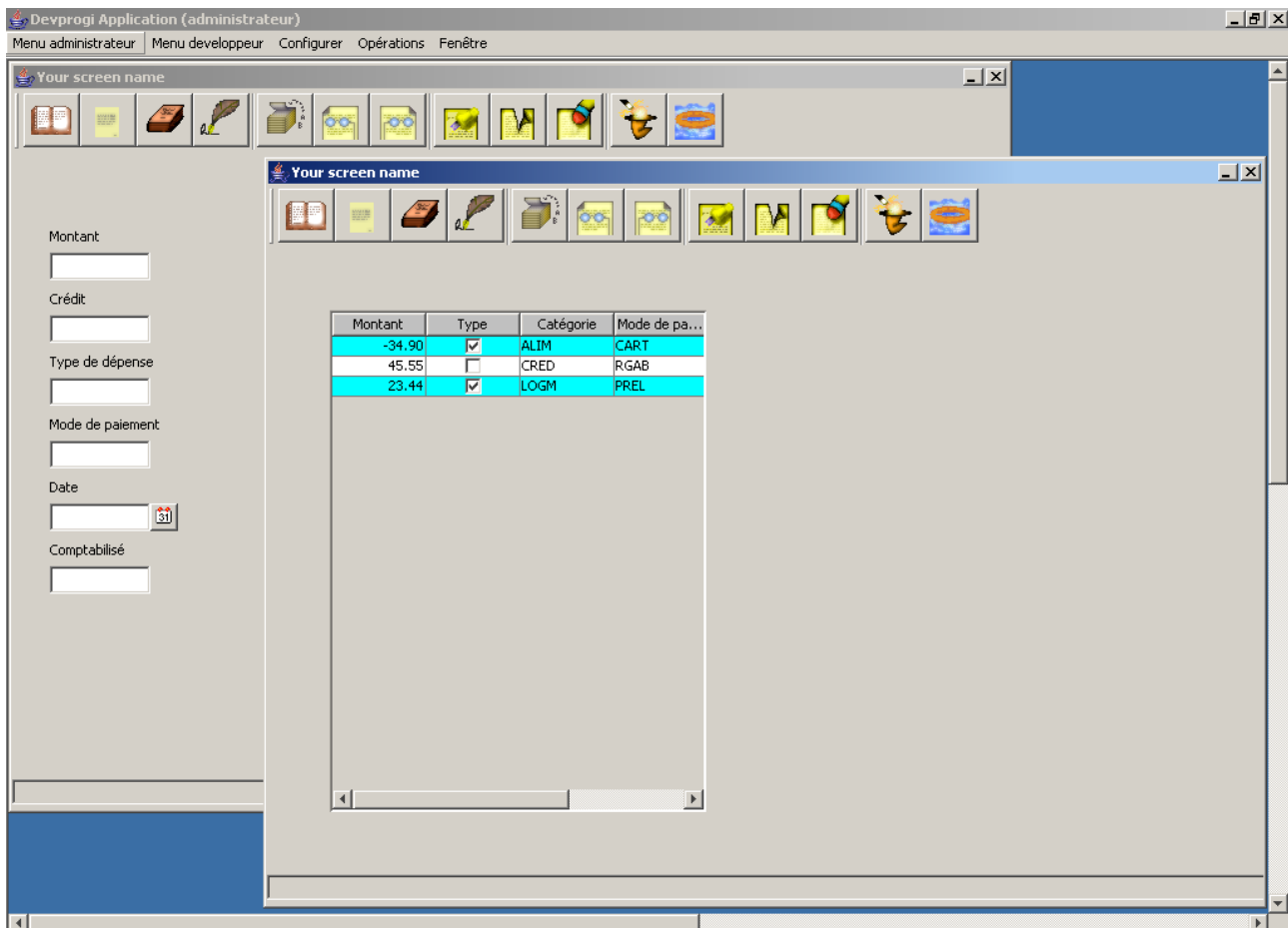
Remarque :

il faut bien déterminer à l'avance l'ordre des menus et si vous désirez ajouter un séparateur. C'est aussi une future évolution à prévoir que d'ajouter la possibilité de changer l'ordre. Il est aussi prévu la possibilité d'ajouter ou non une icône et la nature de l'item.

Un bug de dernière minute : il se peut que l'attribution d'un nœud enfant devienne un nœud père pour le nœud enfant suivant. Vous pouvez le temps que je corrige ce bug, utiliser le drag&drop pour attribuer au nœud enfant le nœud parent.



En vous reconnectant, vous pouvez vérifier que les menus apparaissent bien et que chaque écran se lance correctement.



Maintenant, quand vous irez dans l'éditeur modifier l'écran, vous pourrez le tester par le menu.

Remarque :

Un futur développement devrait prendre en compte les versions afin que au final, la version de l'écran lancé via le menu soit une version de prod et différente de la version dev de l'écran lancé par l'éditeur de classe.

1 Évolution fonctionnelle de l'application

Nous allons désormais améliorer l'application afin de rendre plus pratique la saisie des différentes informations.

1.1 Ajout de listes de valeurs

Nous allons commencer par ajouter une liste de valeur dans l'écran de saisie d'une opération.

Pour ce faire, vous allez ajouter 3 lignes de codes dans le code de l'écran de saisie des opérations après le « `setPanel(buildForm());` »

```
objButtonLov o = new objButtonLov("Catégories", "TAB_CATEGORIE",  
"TYPE_DEP_CODE" ,  
"TYPE_DEP_NAME" );  
o.setDescItemDisplayed(true);  
o.setItem(getField(2));  
o.addComponent(getPanel(),200);
```

explication

l'objet `objButtonLov` permet la création d'un bouton d'appel de liste de valeur.

La méthode permet de positionner le champ pour lequel la liste est créée.

Enfin, permet d'ajouter à l'écran ces derniers avec ici, 200, la longueur du champ descriptif.

Si `setDescItemDisplayed` n'est pas spécifiée, le champ descriptif ne sera pas affiché.

1.2 Modification de la police

Pour changer la police de tous les champs, deux possibilités :

```
setItemFont("police,style,taille","police,style,taille",....);
```

soit pour un champ uniquement :

```
getField(«numéro de colonne »).setItemFont(new Font("policepolice",Font.PLAIN,12));
```

1.3 Modification de la longueur des champs

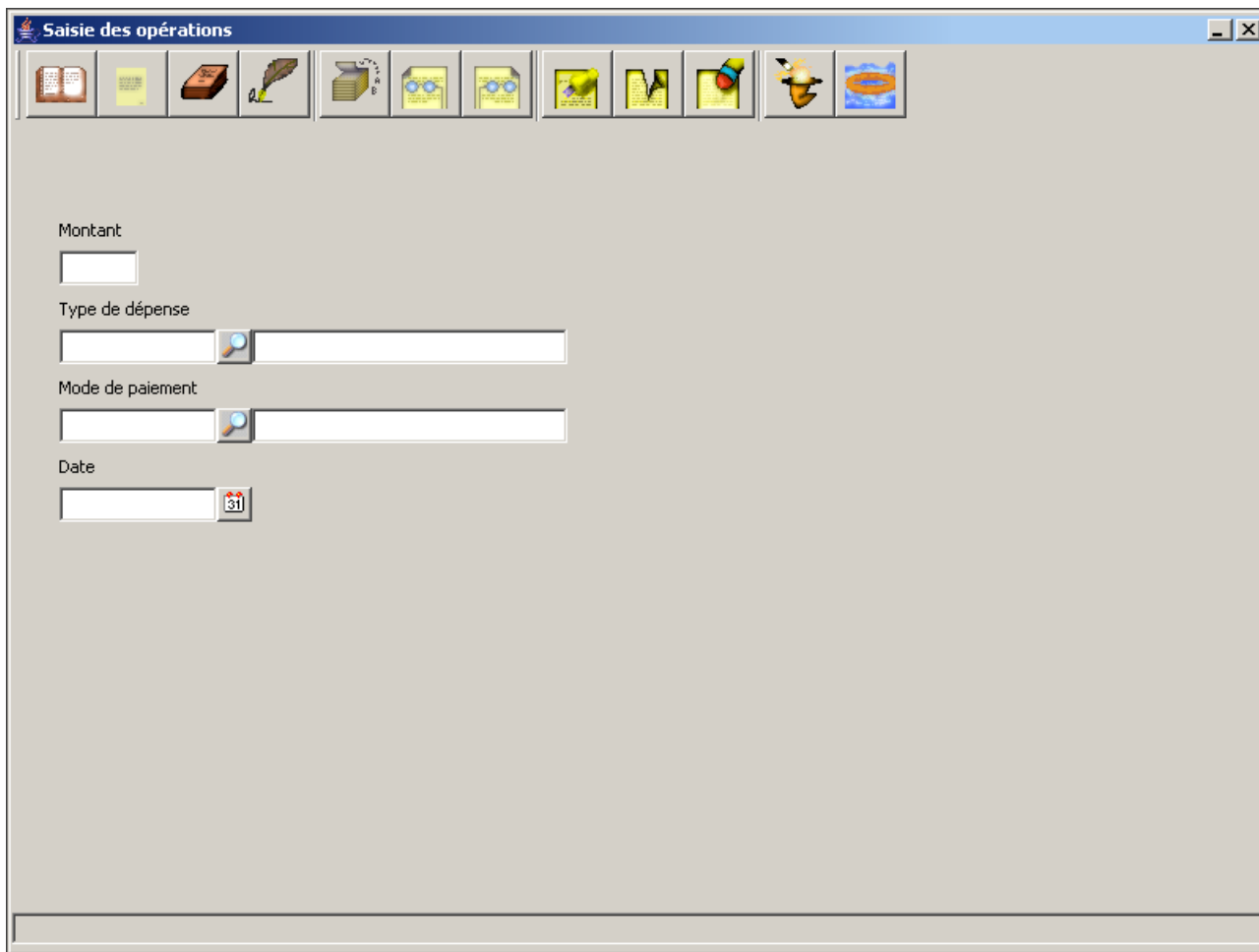
Le resultat final à ce stade est présenté par la copie d'écran suivante

```
setColumnDim(50,100,100,100,100,100);
```

The screenshot shows a web application window titled "Saisie des opérations". The window has a toolbar with various icons. The main content area contains a form with the following fields:

- Montant:
- Crédit:
- Type de dépense:
- Mode de paiement:
- Date:
- Comptabilisé:

Maintenant, nous allons aussi garder les champs qui nous intéressent. Le champs « Crédit » et le champs « Comptabilisé » sont mis en commentaire.

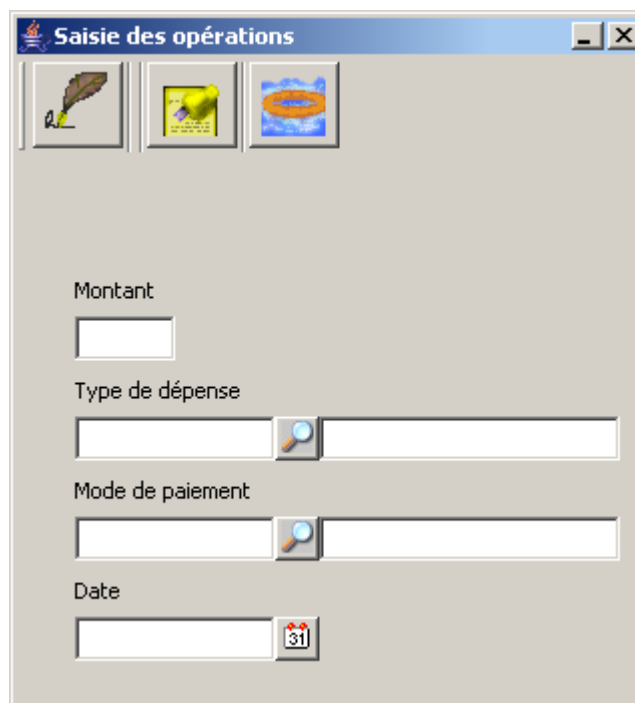


Il reste maintenant à revoir la taille de l'écran et ainsi que les boutons de la barre d'outils à garder.

Les boutons sont :

Nom	Fonction	afficher
openBt	Ouvrir un fichier	
newBt	Nouveau fichier	
closeBt	Fermeture du fichier	
saveBt	Sauvegarder le fichier ou les données	✓
execQueryBt	Exectuter une interrogation	
nxtRecBt	Enregistrement suivant	
prvRecBt	Enregistrement précédent	

Nom	Fonction	afficher
newRecBt	Nouvelle enregistrement	✓
delRecBt	Supprimer un enregistrement	
clearBt	Effacer l'enregistrement de l'écran	
wizardBt	Appel de l'assistant	
helpBt	Appel de l'aide	





The screenshot shows a window titled "Saisie des opérations" with a standard Windows-style title bar. Below the title bar are three small icons: a quill pen, a yellow envelope, and a blue envelope. The main area of the window is labeled "Saisie des opérations :" and contains the following fields:

Montant	solde
39	184.39

Type de dépense
SAL salaire

Mode de paiement
RGAB retrait

Date
2008-1-23

Remarques d'ordre générale

Il est prévu dans de futures développements d'intégrer la technologie bean.
Il est de même prévu d'inclure les raccourcis claviers.