

```

/*****
 * Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
 * Copyright INRA, février 2006                                           *
 *****/

```

```

/*****
 * Licence :

```

Le logiciel portageMM2CPP est une réécriture en langage C++ du modèle écrit dans le fichier ModelMaker "modele2.mod" (situé dans le répertoire "leSourceModelMaker"). Voir informations dans ../laDocumentation/help.

Ce logiciel est régi par la licence CeCILL soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence CeCILL telle que diffusée par le CEA, le CNRS et l'INRIA sur le site "<http://www.cecill.info>".

En contrepartie de l'accessibilité au code source et des droits de copie, de modification et de redistribution accordés par cette licence, il n'est offert aux utilisateurs qu'une garantie limitée. Pour les mêmes raisons, seule une responsabilité restreinte pèse sur l'auteur du programme, le titulaire des droits patrimoniaux et les concédants successifs.

A cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs et des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité.

Le fait que vous puissiez accéder à cet en-tête signifie que vous avez pris connaissance de la licence CeCILL, et que vous en avez accepté les termes.

Les fichiers Licence\_CeCILL\_V2-fr.txt et Licence\_CeCILL\_V2-en.txt du répertoire ../LICENCE fournissent le texte de la licence, dans sa version 2, en langue française et en langue anglaise. (ces 2 fichiers sont aussi directement dans le répertoire courant)

```

*****/
/*****
 *

```

```
* Fichier      : mainPortageMM2CPP.cpp
*
* Auteur(s)    : Nathalie Rousse, Nathalie.Rousse@toulouse.inra.fr
*                de l'INRA - Institut National de la Recherche Agronomique -
*                ( UMR ARCHE / Plate-forme INRA-ACTA-ICTA ).
*
* Description  :
*
* Ce fichier fait partie du code source C++ du logiciel portageMM2CPP.
*
* Il contient une utilisation du modèle : configuration et appel des
* simulations. Voir ../laDocumentation/help, en particulier
* "Présentation générale" et "Spécification".
*
*****
* Historique  :
*
* 03/02/06, Nathalie Rousse : création du fichier.
*
*****/

/* modeDemploi
 * Instruction s'adressant à quelqu'un qui est en train de construire son propre logiciel C++ relativement à son propre
 * modèle ModelMaker : dans tout le code de ce fichier, consulter et suivre les consignes/indications marquées du label "
 * modeDemploi". Voir aussi "Mode d'emploi" dans le fichier "help".
 */

/* modeDemploi
 * Le code de ce fichier est à reprendre pour appeler la simulation comme voulu (paramètres d'appel de "simuler" ...).
 */

#include <iostream>
using namespace std;

#include "genericitePortageMM2CPP.h"
#include "specificitePortageMM2CPP.h"

////////////////////////////////////
///
/// appel du système
///
////////////////////////////////////
```

```

int main ( int argc, char *argv[] )
{
    /* modeDemploi
    *
    * Ecrire son propre appel de la fonction "simuler" (choix de valeurs de paramètres appropriées) en remplacement de l'appel codé ci-dessous. Les différentes alternatives d'appel proposées ci-dessous peuvent servir d'exemples.
    * Il s'agit notamment de choisir quelle méthode appeler parmi les différentes méthodes "simuler" existantes. Pour sa propre utilisation, on peut être amené à écrire une(des) nouvelle(s) méthode(s) "simuler" (voir "modeDemploiEvolutions" dans "genericitePortageMM2CPP.*"). De même par rapport aux formats de sorties souhaités, on peut être conduit à reconsidérer les fonctions qui restituent les résultats des simulations (voir "modeDemploiEvolutions" dans "specificitePortageMM2CPP.*") */

    /*
    * Une alternative d'appel de simulation
    */
    systemParticulier S;
    if ( S.pasDePbInitReleve() == VRAI ){
        cout << " SIMULATION : " << endl << endl;
        S.simuler( 100, 2.0 ); // 100 itérations, pas de simulation 2 s
    } else {
        cout << " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME : code erreur CrInitValue=" << S.getCrInit() << endl << endl;
    }

    /*
    * Une alternative d'appel de simulation
    */
    /*
    systemParticulier S;
    if ( S.pasDePbInitReleve() == VRAI ){
        cout << " SIMULATION : " << endl << endl;
        S.simuler(0.0, 200.0, 2.0 ); // debut 0 s, durée 200 s, pas de simulation 2 s
    } else {
        cout << " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME : code erreur CrInitValue=" << S.getCrInit() << endl << endl;
    }

    */

    /*
    * Une alternative d'appel de simulation
    */
    /*
    // avec configuration de l'horloge locale (instant initial et pas de temps)

```

```
    systemParticulier S( 37.0, 0.001);

    if ( S.pasDePbInitReleve() == VRAI ){
        cout << " SIMULATION : " << endl << endl;
        S.simuler( 100, 2.0 ); // 100 itérations, pas de simulation 2 s
    } else {
        cout << " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME : code erreur CrInitValue=" << S.getCrInit() << endl << endl;
    }

    */

};
```