//CLexplorateur.h

#pragma once

using namespace System::IO;

using namespace System;

namespace NS\_composants

{

ref class CLfichier

{

public:

array<String^> explorerUnDossier(String^);

void effacer(String^);

void copier(String^,String^);

};

}

//CLimages.h

#pragma once

using namespace System;

namespace NS\_composants

{

public ref class CLimages

{

public:

System::Drawing acquisitionImage(String^);

};

}

//CLservicesFichier.h

#pragma once

#include "CLexplorateur.h"

using namespace System;

namespace NS\_services

{

ref class CLserviceFichier

{

private:

NS\_composants::CLfichier^ fichier;

public:

CLserviceFichier(void);

void copier(array<String^>^, String^);

void copierEffacer(array<String^>^, String^);

void effacer(array<String^>^);

array<String^>^ explorer(String^);

};

}

//CLserviceImage.h

#pragma once

#include "CLimages.h"

namespace NS\_services

{

ref class CLserviceImage

{

private:

NS\_composants::CLimages^ image;

public:

CLserviceImage(void);

System::Drawing::Bitmap^ lireImage(String^);

};

}

//myForm.h

#pragma once

#include "CLserviceFichier.h"

#include "CLserviceImage.h"

namespace testWin {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

using namespace System::IO;

/// <summary>

/// Description résumée de MyForm

/// </summary>

public ref class MyForm

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: ajoutez ici le code du constructeur

//

}

protected:

/// <summary>

/// Nettoyage des ressources utilisées.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Button^ btn\_first;

private: System::Windows::Forms::Button^ btn\_previous;

private: System::Windows::Forms::Button^ btn\_next;

private: System::Windows::Forms::Button^ btn\_last;

private: System::Windows::Forms::PictureBox^ pctBx;

private: System::Windows::Forms::Label^ lbl\_vision;

private: System::Windows::Forms::Button^ btn\_source;

private: System::Windows::Forms::Button^ btn\_destination;

private: System::Windows::Forms::TextBox^ txt\_source;

private: System::Windows::Forms::TextBox^ txt\_destination;

private: System::Windows::Forms::GroupBox^ gpBox\_typeProcessus;

private: System::Windows::Forms::RadioButton^ rdb\_effacer;

private: System::Windows::Forms::RadioButton^ rdb\_coiperEffacer;

private: System::Windows::Forms::RadioButton^ rdb\_copier;

private: System::Windows::Forms::Button^ btn\_proceder;

private: System::Windows::Forms::TextBox^ txt\_rslt;

private: System::Windows::Forms::Label^ lbl\_resultats;

private: System::Windows::Forms::FolderBrowserDialog^ fld\_source;

private: System::Windows::Forms::FolderBrowserDialog^ fld\_destination;

private:

/// <summary>

/// Variable nécessaire au concepteur.

/// </summary>

System::ComponentModel::Container ^components;

private: array<String^>^ listeDeFichier;

private: NS\_services::CLserviceImage ^ image;

private: NS\_services::CLserviceFichier^ svcFichier;

private: int index;

private: int nombreImages;

#pragma region Windows Form Designer generated code

/// <summary>

/// Méthode requise pour la prise en charge du concepteur - ne modifiez pas

/// le contenu de cette méthode avec l'éditeur de code.

/// </summary>

void InitializeComponent(void)

{

this->btn\_first = (gcnew System::Windows::Forms::Button());

this->btn\_previous = (gcnew System::Windows::Forms::Button());

this->btn\_next = (gcnew System::Windows::Forms::Button());

this->btn\_last = (gcnew System::Windows::Forms::Button());

this->pctBx = (gcnew System::Windows::Forms::PictureBox());

this->lbl\_vision = (gcnew System::Windows::Forms::Label());

this->btn\_source = (gcnew System::Windows::Forms::Button());

this->btn\_destination = (gcnew System::Windows::Forms::Button());

this->txt\_source = (gcnew System::Windows::Forms::TextBox());

this->txt\_destination = (gcnew System::Windows::Forms::TextBox());

this->gpBox\_typeProcessus = (gcnew System::Windows::Forms::GroupBox());

this->rdb\_effacer = (gcnew System::Windows::Forms::RadioButton());

this->rdb\_coiperEffacer = (gcnew System::Windows::Forms::RadioButton());

this->rdb\_copier = (gcnew System::Windows::Forms::RadioButton());

this->btn\_proceder = (gcnew System::Windows::Forms::Button());

this->txt\_rslt = (gcnew System::Windows::Forms::TextBox());

this->lbl\_resultats = (gcnew System::Windows::Forms::Label());

this->fld\_source = (gcnew System::Windows::Forms::FolderBrowserDialog());

this->fld\_destination = (gcnew System::Windows::Forms::FolderBrowserDialog());

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pctBx))->BeginInit();

this->gpBox\_typeProcessus->SuspendLayout();

this->SuspendLayout();

//

// btn\_first

//

this->btn\_first->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_first->Location = System::Drawing::Point(10, 337);

this->btn\_first->Name = L"btn\_first";

this->btn\_first->Size = System::Drawing::Size(75, 23);

this->btn\_first->TabIndex = 0;

this->btn\_first->Text = L"<<";

this->btn\_first->UseVisualStyleBackColor = true;

//

// btn\_previous

//

this->btn\_previous->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_previous->Location = System::Drawing::Point(91, 337);

this->btn\_previous->Name = L"btn\_previous";

this->btn\_previous->Size = System::Drawing::Size(75, 23);

this->btn\_previous->TabIndex = 1;

this->btn\_previous->Text = L"<";

this->btn\_previous->UseVisualStyleBackColor = true;

this->btn\_previous->Click += gcnew System::EventHandler(this, &MyForm::btn\_previous\_Click);

//

// btn\_next

//

this->btn\_next->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_next->Location = System::Drawing::Point(172, 337);

this->btn\_next->Name = L"btn\_next";

this->btn\_next->Size = System::Drawing::Size(75, 23);

this->btn\_next->TabIndex = 2;

this->btn\_next->Text = L">";

this->btn\_next->UseVisualStyleBackColor = true;

this->btn\_next->Click += gcnew System::EventHandler(this, &MyForm::btn\_next\_Click);

//

// btn\_last

//

this->btn\_last->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_last->Location = System::Drawing::Point(253, 337);

this->btn\_last->Name = L"btn\_last";

this->btn\_last->Size = System::Drawing::Size(75, 23);

this->btn\_last->TabIndex = 3;

this->btn\_last->Text = L">>";

this->btn\_last->UseVisualStyleBackColor = true;

this->btn\_last->Click += gcnew System::EventHandler(this, &MyForm::btn\_last\_Click);

//

// pctBx

//

this->pctBx->BorderStyle = System::Windows::Forms::BorderStyle::FixedSingle;

this->pctBx->Location = System::Drawing::Point(10, 27);

this->pctBx->Name = L"pctBx";

this->pctBx->Size = System::Drawing::Size(317, 302);

this->pctBx->SizeMode = System::Windows::Forms::PictureBoxSizeMode::StretchImage;

this->pctBx->TabIndex = 4;

this->pctBx->TabStop = false;

//

// lbl\_vision

//

this->lbl\_vision->AutoSize = true;

this->lbl\_vision->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->lbl\_vision->Location = System::Drawing::Point(10, 8);

this->lbl\_vision->Name = L"lbl\_vision";

this->lbl\_vision->Size = System::Drawing::Size(69, 13);

this->lbl\_vision->TabIndex = 5;

this->lbl\_vision->Text = L"Visionnage";

//

// btn\_source

//

this->btn\_source->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_source->Location = System::Drawing::Point(333, 27);

this->btn\_source->Name = L"btn\_source";

this->btn\_source->Size = System::Drawing::Size(110, 20);

this->btn\_source->TabIndex = 6;

this->btn\_source->Text = L"Source";

this->btn\_source->UseVisualStyleBackColor = true;

this->btn\_source->Click += gcnew System::EventHandler(this, &MyForm::btn\_source\_Click);

//

// btn\_destination

//

this->btn\_destination->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Italic, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_destination->Location = System::Drawing::Point(333, 53);

this->btn\_destination->Name = L"btn\_destination";

this->btn\_destination->Size = System::Drawing::Size(110, 20);

this->btn\_destination->TabIndex = 7;

this->btn\_destination->Text = L"Destination";

this->btn\_destination->UseVisualStyleBackColor = true;

this->btn\_destination->Click += gcnew System::EventHandler(this, &MyForm::btn\_destination\_Click);

//

// txt\_source

//

this->txt\_source->Location = System::Drawing::Point(449, 27);

this->txt\_source->Name = L"txt\_source";

this->txt\_source->ReadOnly = true;

this->txt\_source->Size = System::Drawing::Size(307, 20);

this->txt\_source->TabIndex = 8;

//

// txt\_destination

//

this->txt\_destination->Location = System::Drawing::Point(449, 53);

this->txt\_destination->Name = L"txt\_destination";

this->txt\_destination->ReadOnly = true;

this->txt\_destination->Size = System::Drawing::Size(307, 20);

this->txt\_destination->TabIndex = 9;

//

// gpBox\_typeProcessus

//

this->gpBox\_typeProcessus->Controls->Add(this->rdb\_effacer);

this->gpBox\_typeProcessus->Controls->Add(this->rdb\_coiperEffacer);

this->gpBox\_typeProcessus->Controls->Add(this->rdb\_copier);

this->gpBox\_typeProcessus->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold,

System::Drawing::GraphicsUnit::Point, static\_cast<System::Byte>(0)));

this->gpBox\_typeProcessus->Location = System::Drawing::Point(336, 88);

this->gpBox\_typeProcessus->Name = L"gpBox\_typeProcessus";

this->gpBox\_typeProcessus->Size = System::Drawing::Size(419, 44);

this->gpBox\_typeProcessus->TabIndex = 10;

this->gpBox\_typeProcessus->TabStop = false;

this->gpBox\_typeProcessus->Text = L"Type de processus";

//

// rdb\_effacer

//

this->rdb\_effacer->AutoSize = true;

this->rdb\_effacer->Location = System::Drawing::Point(357, 19);

this->rdb\_effacer->Name = L"rdb\_effacer";

this->rdb\_effacer->Size = System::Drawing::Size(66, 17);

this->rdb\_effacer->TabIndex = 2;

this->rdb\_effacer->TabStop = true;

this->rdb\_effacer->Text = L"Effacer";

this->rdb\_effacer->UseVisualStyleBackColor = true;

//

// rdb\_coiperEffacer

//

this->rdb\_coiperEffacer->AutoSize = true;

this->rdb\_coiperEffacer->Location = System::Drawing::Point(204, 19);

this->rdb\_coiperEffacer->Name = L"rdb\_coiperEffacer";

this->rdb\_coiperEffacer->Size = System::Drawing::Size(120, 17);

this->rdb\_coiperEffacer->TabIndex = 1;

this->rdb\_coiperEffacer->TabStop = true;

this->rdb\_coiperEffacer->Text = L"Copier et effacer";

this->rdb\_coiperEffacer->UseVisualStyleBackColor = true;

//

// rdb\_copier

//

this->rdb\_copier->AutoSize = true;

this->rdb\_copier->Location = System::Drawing::Point(113, 19);

this->rdb\_copier->Name = L"rdb\_copier";

this->rdb\_copier->Size = System::Drawing::Size(61, 17);

this->rdb\_copier->TabIndex = 0;

this->rdb\_copier->TabStop = true;

this->rdb\_copier->Text = L"Copier";

this->rdb\_copier->UseVisualStyleBackColor = true;

//

// btn\_proceder

//

this->btn\_proceder->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 15.75F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->btn\_proceder->ForeColor = System::Drawing::Color::Blue;

this->btn\_proceder->Location = System::Drawing::Point(336, 141);

this->btn\_proceder->Name = L"btn\_proceder";

this->btn\_proceder->Size = System::Drawing::Size(420, 61);

this->btn\_proceder->TabIndex = 11;

this->btn\_proceder->Text = L"Procéder";

this->btn\_proceder->UseVisualStyleBackColor = true;

this->btn\_proceder->Click += gcnew System::EventHandler(this, &MyForm::btn\_proceder\_Click);

//

// txt\_rslt

//

this->txt\_rslt->Location = System::Drawing::Point(337, 227);

this->txt\_rslt->Multiline = true;

this->txt\_rslt->Name = L"txt\_rslt";

this->txt\_rslt->ReadOnly = true;

this->txt\_rslt->Size = System::Drawing::Size(419, 133);

this->txt\_rslt->TabIndex = 12;

//

// lbl\_resultats

//

this->lbl\_resultats->AutoSize = true;

this->lbl\_resultats->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 8.25F, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(0)));

this->lbl\_resultats->Location = System::Drawing::Point(334, 211);

this->lbl\_resultats->Name = L"lbl\_resultats";

this->lbl\_resultats->Size = System::Drawing::Size(60, 13);

this->lbl\_resultats->TabIndex = 13;

this->lbl\_resultats->Text = L"Résultats";

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(766, 373);

this->Controls->Add(this->lbl\_resultats);

this->Controls->Add(this->txt\_rslt);

this->Controls->Add(this->btn\_proceder);

this->Controls->Add(this->gpBox\_typeProcessus);

this->Controls->Add(this->txt\_destination);

this->Controls->Add(this->txt\_source);

this->Controls->Add(this->btn\_destination);

this->Controls->Add(this->btn\_source);

this->Controls->Add(this->lbl\_vision);

this->Controls->Add(this->pctBx);

this->Controls->Add(this->btn\_last);

this->Controls->Add(this->btn\_next);

this->Controls->Add(this->btn\_previous);

this->Controls->Add(this->btn\_first);

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->MaximizeBox = false;

this->Name = L"MyForm";

this->Text = L"[EI] [A2 MININF] Bloc Programmation orientée objet- Prosit 5";

this->Load += gcnew System::EventHandler(this, &MyForm::MyForm\_Load);

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->pctBx))->EndInit();

this->gpBox\_typeProcessus->ResumeLayout(false);

this->gpBox\_typeProcessus->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void MyForm\_Load(System::Object^ sender, System::EventArgs^ e)

{

this->index = 0;

this->nombreImages = 0;

}

private: System::Void btn\_source\_Click

{

int i;

this->image = gcnew NS\_services::CLserviceImage();

this->fld\_source->ShowDialog();

this->txt\_source->Text = this->fld\_source->SelectedPath;

this->listeDeFichier = this->svcFichier->explorer(this->txt\_source->Text);

for (i = 0; i < this->listeDeFichier->Length; i++)

{

this->txt\_rslt->Text += this->listeDeFichier[i] + "\r\n";

}

this->pctBx->Image = image->lireImage(this->listeDeFichier[this->index]);

}

private: System::Void btn\_destination\_Click(System::Object^ sender, System::EventArgs^ e)

{

this->fld\_destination->ShowDialog();

this->txt\_destination->Text = this->fld\_destination->SelectedPath;

}

private: System::Void btn\_first\_Click(System::Object^ sender, System::EventArgs^ e)

{

this->index = 0;

this->pctBx->Image = image->lireImage(this->listeDeFichier[this->index]);

}

private: System::Void btn\_proceder\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (this->rdb\_copier->Checked == true)

{

this->svcFichier->copier(this->listeDeFichier, this->txt\_destination->Text);

this->txt\_rslt->Clear();

this->txt\_rslt->Text = "Les fichiers ont été copier dans le dossier : " + this->txt\_destination->Text;

}

else if (this->rdb\_coiperEffacer->Checked == true)

{

this->svcFichier->copierEffacer (this->listeDeFichier, this->txt\_destination->Text);

this->txt\_rslt->Text = "Les fichiers ont été copier dans le dossier : " + this->txt\_destination->Text;

}

}

private: System::Void btn\_next\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (this->index < this->listeDeFichier->Length - 1)

{

this->index++;

this->pctBx->Image = image->lireImage(this->listeDeFichier[this->index]);

}

}

private: System::Void btn\_previous\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (this->index > 0)

{

this->index--;

this->pctBx->Image = image->lireImage(this->listeDeFichier[this->index]);

}

}

private: System::Void btn\_last\_Click(System::Object^ sender, System::EventArgs^ e)

{

this->index = this->listeDeFichier->Length - 1;

this->pctBx->Image = image->lireImage(this->listeDeFichier[this->index]);

}

};

}

//CLexplorateur.cpp

#include "CLexplorateur.h"

namespace NS\_composants

{

array<String^>^ CLfichier::explorerUnDossier(String^ dossier)

{

return file;

}

void CLfichier::effacer(String^ fichier)

{

System::IO::File::Delete(fichier);

}

void CLfichier::copier(String^ fichierSource, String^ fichierDestination)

{

String^ nomDeFichierExtension;

nomDeFichierExtension = System::IO::Path::GetFileName(fichierSource);

System::IO::File::Copy(fichierSource, fichierDestination +"\\"+nomDeFichierExtension);

}

}

//CLimages.cpp

#include "CLimages.h"

using namespace System;

namespace NS\_composants

{

System::Drawing::Bitmap^ CLimages::acquisitionImage(String^ fichier)

{

return gcnew System::Drawing::Bitmap(fichier);

}

}

//CLserviceFichier.cpp

#include "CLserviceFichier.h"

namespace NS\_services

{

CLserviceFichier::CLserviceFichier(void)

{

this->fichier = gcnew NS\_composants::CLfichier();

}

void CLserviceFichier::copier(array<String^>^ fichiersSources, String^ destination)

{

int i;

i = 0;

for (i = 0; i < fichiersSources->Length; i++)

{

this->fichier->copier(fichiersSources[i], destination);

}

}

void CLserviceFichier::copierEffacer(array<String^>^ fichiersSources, String^ fichierDestination)

{

this->copier(fichiersSources, fichierDestination);

this->effacer(fichiersSources);

}

void CLserviceFichier::effacer(array<String^>^ fichiersSources)

{

int i;

i = 0;

for (i = 0; i < fichiersSources->Length; i++)

{

this->fichier->effacer(fichiersSources[i]);

}

}

array<String^>^ CLserviceFichier::explorer(String^ dossier)

{

return this->fichier->explorerUnDossier(dossier);

}

}

//CLserviceImage.cpp

#include "CLserviceImage.h"

namespace NS\_services

{

CLserviceImage::CLserviceImage(void)

{

this->image = gcnew NS\_composants::CLimages();

}

System::Drawing::Bitmap^ CLserviceImage::lireImage(String^ image)

{

return this->image->acquisitionImage(image);

}

}

//MyForm.cpp

#include "MyForm.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

int \_\_clrcall WinMain(array<String^>^ args )

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

testWin::MyForm monFormulaire;

}