In diesem Tutorial werden wir euch vorstellen, wie man eine C++ Klasse zu einer C# Klasse (auch mit den Methoden) marschlen. Das Marschal (engl. Marshaling) bedeutet die Überweisung von einer Klasse aus einem C++ Projekt zu einem C# Project.

Wir haben schon erklärt in der vorherigen Teile wie die Interoperabilität zwischen C++ und C# funktioniert. Wir haben eine Projectmappe entwickelt und konfiguriert.

Wir werden weiter das Projekt entwickeln, indem eine C++ Klasse („Wishlist“) und marschlien sie zur unseren C# Anwendung. Um zu kurz und bündig zu sein, werden wir nicht unnötigen Details bieten, z.B (wie viele Methoden es gibt, was die Methoden machen unsw.) sondern, was den konkreten technischen Schritten sind.

Schritt 1. Bitte füngt in den C++ Project (NativeLibray in unserm Fall) die folgenden Datei hinzu: WishList.cpp, WishList.h und WishListWrapper.cpp. Der Quellecode muss nicht abschreiben werden. Eine Repository gibt es schon.

WishList.h (hier deklariert man die Methoden)

#pragma once

#include <string>

#include <vector>

class WishList {

public:

WishList(std::string name) : \_name(name), \_items() { }

std::string getName();

void setName(std::string name);

int countItems();

void addItem(std::string item);

void removeItem(std::string item);

void print();

private:

std::string \_name;

std::vector<std::string> \_items;

};

WishList.cpp (hier ungesetzt man die Methoden)

#include "WishList.h"

#include <iostream>

std::string WishList::getName() {

return \_name;

}

void WishList::setName(std::string name) {

\_name = name;

}

void WishList::addItem(std::string item) {

\_items.push\_back(item);

}

void WishList::removeItem(std::string item) {

for (int i = 0; i < \_items.size(); ++i) {

if (\_items[i] == item) {

\_items.erase(\_items.begin() + i);

}

}

}

int WishList::countItems() {

return \_items.size();

}

void WishList::print() {

for (std::string item : \_items) {

std::cout << item << std::endl;

}

}

WishListWrapper.cpp (hier exportiert man die Methoden)

#include "WishList.h"

#include <comdef.h>

#include <comutil.h>

#include <iostream>

#include <cstdio>

#include <fstream>

extern "C" \_\_declspec(dllexport) WishList \* CreateWishList(const char\* name) {

return new WishList(name);

}

extern "C" \_\_declspec(dllexport) void DeleteWishList(WishList \* wishList) {

delete wishList;

}

extern "C" \_\_declspec(dllexport) BSTR GetWishListName(WishList \* wishList) {

return \_com\_util::ConvertStringToBSTR(wishList->getName().c\_str());

}

extern "C" \_\_declspec(dllexport) void SetWishListName(WishList \* wishList, const char\* name) {

wishList->setName(name);

}

extern "C" \_\_declspec(dllexport) void AddWishListItem(WishList \* wishList, const char\* name) {

wishList->addItem(name);

}

extern "C" \_\_declspec(dllexport) void RemoveWishListItem(WishList \* wishList, const char\* name) {

wishList->removeItem(name);

}

extern "C" \_\_declspec(dllexport) int CountWishListItems(WishList \* wishList) {

return wishList->countItems();

}

extern "C" \_\_declspec(dllexport) void PrintWishList(WishList \* wishList) {

wishList->print();

}

Schritt 2. Bitte füngt in den C# die folgende Datei hinzu: WhishList.cs:

using System;

using System.Runtime.InteropServices;

namespace ConsoleApplication.Models

{

public class WishList

{

[DllImport("NativeLibrary.dll")]

private static extern IntPtr CreateWishList(string name);

[DllImport("NativeLibrary.dll")]

private static extern void DeleteWishList(IntPtr wishListPointer);

[DllImport("NativeLibrary.dll")]

[return: MarshalAs(UnmanagedType.BStr)]

private static extern string GetWishListName(IntPtr wishListPointer);

[DllImport("NativeLibrary.dll")]

private static extern void SetWishListName(IntPtr wishListPointer, string name);

[DllImport("NativeLibrary.dll")]

private static extern void AddWishListItem(IntPtr wishListPointer, string name);

[DllImport("NativeLibrary.dll")]

private static extern void RemoveWishListItem(IntPtr wishListPointer, string name);

[DllImport("NativeLibrary.dll")]

private static extern int CountWishListItems(IntPtr wishListPointer);

[DllImport("NativeLibrary.dll")]

private static extern void PrintWishList(IntPtr wishListPointer);

private readonly IntPtr \_wishListPointer;

public string Name

{

get

{

return GetWishListName(\_wishListPointer);

}

set

{

SetWishListName(\_wishListPointer, value);

}

}

public int Count => CountWishListItems(\_wishListPointer);

public WishList(string name)

{

\_wishListPointer = CreateWishList(name);

}

~WishList()

{

DeleteWishList(\_wishListPointer);

}

public void AddItem(string name)

{

AddWishListItem(\_wishListPointer, name);

}

public void RemoveItem(string name)

{

RemoveWishListItem(\_wishListPointer, name);

}

public void Print()

{

PrintWishList(\_wishListPointer);

}

}

}

Schritt 3. Bearbeite bitte die Program.cs Datei wie unten gezeigt:

using ConsoleApplication.Models;

using System;

using System.Runtime.InteropServices;

namespace ConsoleApplication

{

class Program

{

[DllImport("NativeLibrary.dll")]

[return: MarshalAs(UnmanagedType.BStr)]

public static extern string GetName();

[DllImport("NativeLibrary.dll")]

public static extern void BuyShoe(Shoe shoe);

[DllImport("NativeLibrary.dll")]

public static extern Shoe CreateShoe(double shoeSize);

static void Main(string[] args)

{

WishList wishList = new WishList("Life");

wishList.Name = "Birthday";

wishList.AddItem("Lamp");

wishList.AddItem("Graphics Card");

wishList.AddItem("Chipotle Burrito Bowl");

wishList.RemoveItem("Lamp");

Console.WriteLine($"{wishList.Name}: {wishList.Count} items");

wishList.Print();

Console.ReadLine();

}

}

}