Приложение 2.

Решение задачи регистрации ресурсов, вариант V1, на языке C#

using System.IO;

namespace ResRegV1cons

{

class ResAreBusy : Exception { }

class ResIdInvalid : Exception { }

class UnRecommended : Exception { }

class ResIsBusy : Exception { }

class ResWasFree : Exception { }

static class SetUp

{

public static string Path; //путь к файлу, сохраняющему модель

private static void ClearModel()

{

Console.WriteLine("Укажите количество ресурсов:");

try

{

Model.vRes\_s = new string[Convert.ToInt32(Console.ReadLine())];

for (int i = 0; i < Model.vRes\_s.Length; i++) Model.vRes\_s[i] = "F"; }

catch

{

Console.WriteLine("Введено некорректное число!");

ClearModel();

}

}

private static void GetModel()

{

Console.WriteLine("Обновить файл?");

if (Console.ReadLine().ToUpper() == "Y") ClearModel();

else

{

Model.vRes\_s = File.ReadAllLines(Path);

}

}

public static bool On()

{

try

{

if (File.Exists(Directory.GetCurrentDirectory() + @"\Resmod00"))

{

Console.WriteLine("Использовать существующий стандартный файл Resmod00?");

if (Console.ReadLine().ToUpper() == "Y")

{

Path = Directory.GetCurrentDirectory() + @"\Resmod00";

GetModel();

return true;

}

}

else

{

Console.WriteLine("Создать стандартный файл?");

if (Console.ReadLine().ToUpper() == "Y")

{

Path = Directory.GetCurrentDirectory() + @"\Resmod00";

ClearModel();

return true;

}

};

Console.WriteLine("Введите полный адрес нестандартного файла:");

Path = Console.ReadLine();

if (File.Exists(Path))

{

GetModel();

return true;

}

else

{

ClearModel();

return true;

}

}

catch (IOException) { Console.WriteLine("Файл не открылся."); return false; }

catch (Exception) { Console.WriteLine("Ошибка ввода-вывода."); return false; }

}

}

static class Model

{

public static string[] vRes\_s;//Модель набора ресурсов

public static void Occupy(string cn)

{

if ((Convert.ToInt16(cn) > vRes\_s.Length) | (Convert.ToInt16(cn) < 0)) throw new ResIdInvalid();

if(vRes\_s[Convert.ToInt16(cn) - 1] == "B")throw new ResIsBusy();

vRes\_s[Convert.ToInt16(cn) - 1] = "B";

}

public static void Free(string cn)

{

if ((Convert.ToInt16(cn) > vRes\_s.Length) | (Convert.ToInt16(cn) < 0)) throw new ResIdInvalid();

if (vRes\_s[Convert.ToInt16(cn) - 1] == "F") throw new ResWasFree();

vRes\_s[Convert.ToInt16(cn) - 1] = "F";

}

public static string Request()

{

for(int i=0;i<vRes\_s.Length;i++)

{

if(vRes\_s[i] == "F") return Convert.ToString(i + 1);

}

throw new ResAreBusy(); ;

}

}

class Program

{

static void Main(string[] args)

{

string Command;

while (!SetUp.On());

do

{

File.WriteAllLines(SetUp.Path, Model.vRes\_s);//сохранение модели

Console.WriteLine("Введите команду:");

Command = Console.ReadLine();

Command = Command.ToUpper();

try

{

if (Command == "REQUEST") Console.WriteLine(Model.Request());

if (Command == "OCCUPY")

{

Console.WriteLine("Введите номер ресурса:");

Model.Occupy(Console.ReadLine());

Console.WriteLine("Ресурс стал занятым.");

};

if (Command == "FREE")

{

Console.WriteLine("Введите номер ресурса:");

Model.Free(Console.ReadLine());

Console.WriteLine("Ресурс освобождён.");

};

}

catch (OverflowException) { Console.WriteLine("Такого ресурса нет."); }

catch (FormatException) { Console.WriteLine("Такого ресурса нет."); }

catch (ResIdInvalid) { Console.WriteLine("Такого ресурса нет."); }

catch (ResWasFree) { Console.WriteLine("Ресурс был свободен."); }

catch (ResAreBusy) { Console.WriteLine("Все ресурсы заняты."); }

catch (ResIsBusy) { Console.WriteLine("ресурс уже занят."); }

}

while(Command != "");

}

}

}

Приложение 3.

Решение задачи регистрации ресурсов, вариант V2, на языке C++

// ResRegV2cpp.cpp: главный файл проекта.

#include "stdafx.h"

using namespace System;

using namespace System::IO;

ref class ResAreBusy : Exception { };

ref class ResIsBusy : Exception { };

ref class ResIdInvalid : Exception { };

ref class ResWasFree : Exception { };

public ref class SetUp

{

public: static System::String ^Path;//путь к файлу, сохраняющему модель

private: static void ClearModel()

{

Console::WriteLine("Укажите количество ресурсов:");

try

{

vRes\_s = gcnew array<System::String^>(Convert::ToInt32(Console::ReadLine()));

for (int i = 0; i < vRes\_s->Length; i++) vRes\_s[i] = "F";

}

catch ( ... )

{

Console::WriteLine("Введено некорректное число!");

ClearModel();

}

}

private: static void GetModel()

{

Console::WriteLine("Обновить файл?");

if (Console::ReadLine()->ToUpper() == "Y") ClearModel();

else

{

vRes\_s = File::ReadAllLines(Path);

}

}

public: static bool On()

{

try

{

if (File::Exists(Directory::GetCurrentDirectory() + "Resmod00"))

{

Console::WriteLine("Использовать существующий стандартный файл Resmod00?");

if (Console::ReadLine()->ToUpper() == "Y")

{

Path = Directory::GetCurrentDirectory() + "Resmod00";

GetModel();

return true;

}

}

else

{

Console::WriteLine("Создать стандартный файл?");

if (Console::ReadLine()->ToUpper() == "Y")

{

Path = Directory::GetCurrentDirectory() + "Resmod00";

ClearModel();

return true;

}

};

Console::WriteLine("Введите полный адрес нестандартного файла:");

Path = Console::ReadLine();

if (File::Exists(Path))

{

GetModel();

return true;

}

else

{

ClearModel();

return true;

}

}

catch (const IOException^ e ) { Console::WriteLine("Файл не открылся."); return false; }

catch (const Exception^ e) { Console::WriteLine("Ошибка ввода-вывода."); return false; }

}

public: static array<System::String^> ^vRes\_s; //Модель набора ресурсов

public: static void Occupy(String^ cn)

{

if ((Convert::ToInt16(cn) > vRes\_s->Length) | (Convert::ToInt16(cn) < 0)) throw gcnew ResIdInvalid();

if (vRes\_s[Convert::ToInt16(cn) - 1] == "B")throw gcnew ResIsBusy();

vRes\_s[Convert::ToInt16(cn) - 1] = "B";

}

public: static void Free(String^ cn)

{

if ((Convert::ToInt16(cn) > vRes\_s->Length) | (Convert::ToInt16(cn) < 0)) throw gcnew ResIdInvalid();

if (vRes\_s[Convert::ToInt16(cn) - 1] == "F") throw gcnew ResWasFree();

vRes\_s[Convert::ToInt16(cn) - 1] = "F";

}

public: static String^ Request()

{

for (int i = 0; i<vRes\_s->Length; i++)

{

if (vRes\_s[i] == "F") return Convert::ToString(i + 1);

}

throw gcnew ResAreBusy();

}

};

int main(array<System::String ^> ^args)

{

String ^Command;

while ((!SetUp::On()));

do

{

File::WriteAllLines(SetUp::Path, SetUp::vRes\_s);//сохранение модели

Console::WriteLine("Введите команду:");

Command = Console::ReadLine();

Command = Command->ToUpper();

try

{

if (Command == "REQUEST") Console::WriteLine(SetUp::Request());

if (Command == "OCCUPY")

{

Console::WriteLine("Введите номер ячейки:");

SetUp::Occupy(Console::ReadLine());

Console::WriteLine("Ячейка стала занята.");

};

if (Command == "FREE")

{

Console::WriteLine("Введите номер ячейки:");

SetUp::Free(Console::ReadLine());

Console::WriteLine("Ячейка освобождена.");

};

}

catch (const OverflowException^ e) { Console::WriteLine("Такого ресурса нет."); }

catch (const FormatException^ e) { Console::WriteLine("Такого ресурса нет."); }

catch (const ResIdInvalid^ e) { Console::WriteLine("Такого ресурса нет."); }

catch (const ResWasFree^ e) { Console::WriteLine("Ресурс был свободен."); }

catch (const ResAreBusy^ e) { Console::WriteLine("Все ресурсы заняты."); }

catch (const ResIsBusy^ e) { Console::WriteLine("ресурс уже занят."); }

}

while ((Command != ""));

return 0;

}