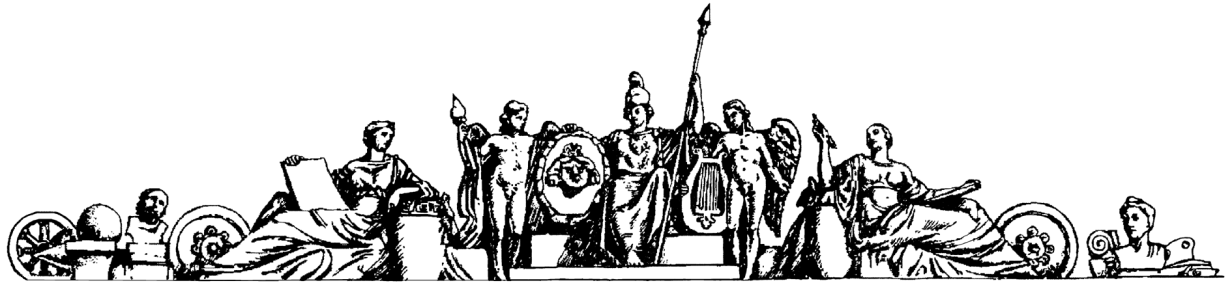
**Министерство образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

**Алгоритмические языки программирования**

#### **Дз-1, Джойнер. Пояснительная записка**

**Студент**: Пряхин В.А.

**ИУ8-31**

Москва 2016

**Задание:** Создать джойнер для операционной системы Windows. Программирование должно вестись на языке C++

**Ход решения:** Проект был разделен на 2 части. Создание программы, запускающей файлы. И создание билдера.

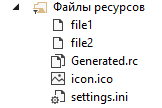
**1.Создание программы, запускающей файлы.**

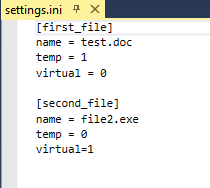
**Среда разработки:** Visual Studio 2015.

**План работы программы**:

1. Загрузка настроек
   1. Распаковка 1 и 2 файла.

Уделено много времени на работу с ресурсами. В ресурсы программы было загружено 4 файла:

* File1 – Первый запускаемый файл
* File2 - Второй запускаемый файл
* Icon.ico – Иконка приложения
* Settings.ini - Настройки программы.

Сначала проводится загрузка настроек. Примерное содержание settings.ini:

**[first\_file],[second\_file]** –настройки для первого и второго файла соответственно.

**name** – имя распаковываемого файла

**temp** - распаковывать файл в папке Temp (1 – да, распаковывать в Temp, 0 – нет, распаковывать в папке с программой)

**virtual** – запуск программы из памяти(1 – да, 0 – нет, сохранять на диск и уже запускать с диска)

**Основные методы**

|  |  |
| --- | --- |
| Прототип метода | Содержание метода |
| BOOL extract(DWORD numb, LPCSTR name) | Служит для выгрузки файлов из ресурсов, и их запуска |
| void get\_setings() | Загрузка настроек из файла settings.ini, находящегося в ресурсах |
| void RunFromMemory(LPSTR szFilePath, LPVOID pFile) | Запуск файла из памяти. За основу взята функция RunPE- https://github.com/m0n0ph1/Basic-File-Crypter/blob/master/runPE.h |

**Блок-схема, демонстрирующая работу программы**:

**C:\Users\pryah\Desktop\ДЗ отчеты\Диаграмма.png**

**2.Создание билдера.**

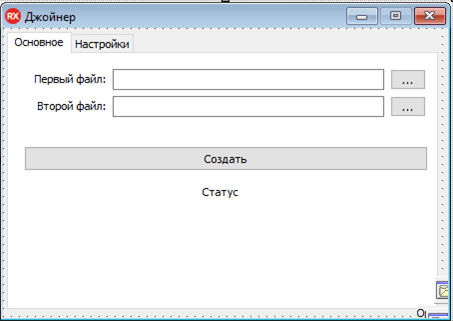
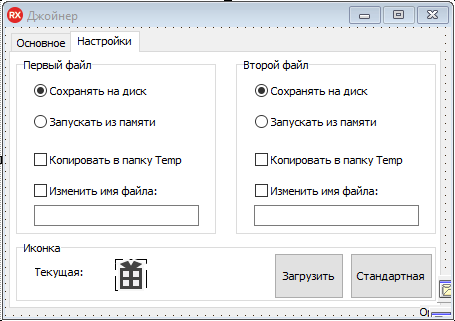
**Среда разработки:** C++ Builder 10.1 Berlin. Изначально билдер был написан в Visual Studio, но после был перенесен в C++ Builder.

**План работы программы:**

1. Загрузка пользователем данных
2. Выбор настроек, создание файла settings.ini
3. Выгрузка из ресурсов первой программы
4. Замена ресурсов первой программы в соответствии с настройками пользователя

**Разработка пользовательского интерфейса**

Для удобства и лаконичности было принято решение не пихать всё в одну кучу, а разбить на разделы. Разбиение было выполнено на разделы “Основное” и “Настройки” с помощью стандартного элемента TPageControl



Так как размеры разные, то чтобы избежать небрежности в виде большого белого пустого пространства, была написана функция, меняющая размеры формы при смени вкладок.

На вкладке “Основное” было размещено самое главное, выбор склеиваемых файлов и кнопка запуска склейки. Во вкладке “Настройки” были помещены собственно сами настройки, на основе которых генерируется settings.ini, а так же загрузка иконки.

**Основные методы**

|  |  |
| --- | --- |
| Прототип метода | Содержание метода |
| bool extract(DWORD numb) | Служит для выгрузки первой программы из ресурсов |
| std::vector<unsigned char> GetFile(wchar\_t\* name) | Запись файла с диска в vector<unsigned char> |
| BOOL ChangeExeIcon(LPCTSTR lpIconFile, HANDLE hUpdate) | Замена иконки в exe файле. За основу взята функция с форума Microsoft - https://social.msdn.microsoft.com/Forums/vstudio/en-US/d1fc85aa-d296-43b8-b0df-d5da7d94dcb5/the-ico-file-and-its-struct-?forum=vcgeneral |
| std::string TForm1::createini(); | Создание settings.ini |
| void \_\_fastcall TForm1::Button3Click(TObject \*Sender); | Button3 – та самая кнопка “Создать”, по нажатию на которую и происходит вся магия. |

**Блок-схема** второй программы будет приложена(“Блок-схема 2.png”)

**3.Создание документации к программе.**

После завершения работы над техническим заданием была выполнена работа над созданием документацией. В документацию входят пояснительная записка, блок-схемы алгоритмов работы программ и руководство пользователя.

**4.Ввод программы в действие.**

Программа была продемонстрирована в действии на защите домашнего задания по курсу «Алгоритмические языки программирования» третьего семестра.