

Министерство образования Республики Беларусь
Учреждение образования «Барановичский государственный университет»
Студенческое научное общество БарГУ

СОДРУЖЕСТВО НАУК. БАРАНОВИЧИ-2015

Материалы XI Международной
научно-практической конференции
молодых исследователей

(Барановичи, 21—22 мая 2015 года)

В трёх частях

Часть 2

Барановичи
БарГУ
2015

В части 2 сборника материалов XI Международной научно-практической конференции молодых исследователей «Содружество наук. Барановичи-2015» представлены результаты исследований в области информационных систем и технологий в образовании, науке и технике, а также рассмотрены актуальные проблемы в области физики и математики. Особое внимание уделено результатам исследований современных тенденций в технологиях и материалах машиностроительного и сельскохозяйственного производств.

Сборник адресован научным работникам, аспирантам, магистрантам и студентам инженерных специальностей вузов.

Редакционная коллегия:

А. В. Никишова (гл. ред.), Ю. Е. Горбач (отв. ред.), О. И. Наранович,
А. К. Гавриленя, В. А. Дремук, Е. Н. Кирюхова

Рецензенты:

кандидат технических наук А. М. Милюкова,
кандидат физико-математических наук, доцент Д. А. Ционенко

А. И. Калько, А. А. Масло, О. И. Наранович

Учреждение образования «Барановичский государственный университет», Барановичи

ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ DLL-БИБЛИОТЕК

Введение. С развитием IT-индустрии, совершенствованием аппаратной части ЭВМ и в связи со стремительным процессом старения программных комплексов ПЭВМ, большое внимание уделяется созданию универсальных, соответствующих требованиям пользователя программ. На сегодняшний день при разработке программных продуктов основным влиянием служит человеческий фактор. Иными словами это непосредственное вмешательство человека в процесс дальнейшей работы итогового продукта. Поскольку большинство программных пакетов и комплексов разработано на обычном уровне программирования, с применением структурирования, то возникает проблема универсальности данного программного продукта, ещё более актуальной она становится спустя некоторое время, когда продукт устаревает. Как известно, структурированное программирование со временем было вытеснено объектно-ориентированным.

Объектно-ориентированное программирование (ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов. Класс — абстрактный тип данных в объектно-ориентированном программировании, задающий общее поведение для группы объектов, модели объекта [1].

Объектно-ориентированное программирование стало наиболее актуальной концепцией при разработке программных продуктов. Одно из его преимуществ — создание более сложного, решающего различное множество задач программного обеспечения. Одним из способов построения многофунк-

ционального, универсального пользовательского обеспечения и его альтернативным решением является создание программных комплексов с использованием DLL (Dynamic Link Library) библиотек.

Основная часть. DLL-библиотеки представляют собой набор подпрограмм, которые могут использоваться в приложении или другой DLL-библиотекой. Иными словами, DLL-библиотеки — это просто наборы функций, собранные в библиотеки.

С самого рождения операционная система Windows использовала библиотеки динамической компоновки, в которых содержались наиболее часто применяемых функций. Наследники Windows — NT и Windows 95, а также OS/2 — тоже зависят от библиотек DLL в плане обеспечения значительной части их функциональных возможностей. Практически невозможно создать приложение Windows, в котором не использовались бы библиотеки DLL. В DLL содержатся все функции Win32 API и большое количество других функций операционных систем Win32.

$$y = -6x_1 + 2x_2^2 - 2x_1x_2 + 2x_2^2 \rightarrow \min$$

$$x_1 + x_2 \leq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Рисунок 1 — Математическая модель оптимизационной задачи

Рассмотрим применение DLL-библиотек на примере разработки приложения для решения оптимизационной задачи нелинейного программирования (рисунок 1).

Цель исследования состояла в оценке достоинств и недостатков использования DLL-библиотек при разработке приложения решения оптимизационных задач следующими методами: метод прямого перебора, метод Монте-Карло, метод Хука-Дживса, метод штрафных функций, метод случайного линейного и нелинейного поиска, эволюционный бионический метод.

Разработанный в среде Microsoft Visual Studio 2012 на языке программирования C# проект содержит DLL-библиотеки, в которых реализованы функции оптимизационных методов.

Нами установлено, что создание приложения с использованием DLL-библиотек обладает рядом преимуществ:

1) *экономия оперативной памяти используемой приложением.* Это обусловлено тем, что выделение динамической памяти происходит во время выполнения определённой функции, которая принадлежит соответствующей DLL-библиотеке, а не во время компиляции программы (рисунок 2). Использование только одного файла DLL-библиотеки даёт экономию 2% оперативной памяти, что существенно при работе больших приложений;

2) *время выполнения программы с DLL-библиотекой быстрее обычного.* Это объясняется тем, что выделение динамической памяти требует не только использование оперативной памяти, но и времени под её выделение и освобождение. Нами было установлено, что при запуске приложения, использующего DLL-библиотеку метода прямого перебора, приложение работает в 2 раза быстрее (1,06 мс), чем без использования DLL-библиотеки (2,29 мс);

3) *существенное уменьшение программного кода в проекте.* Чтобы не создавать громоздкие проекты, в которых потом трудно разбираться, можно разбить его на различные функции и создать DLL-библиотеки. Такой подход можно назвать декомпозицией проекта, т. е. разбиение проекта на мелкие блоки. Данное решение очень полезно для командной разработки программного обеспечения (рисунок 3).

4) *переносимость DLL-библиотеки в другие проекты, их использование в нескольких проектах.* Характерно тем, что нет необходимости полностью создавать приложение, если есть частичная реализация в другом проекте, а также возможность обновления проекта без полной переработки. Использование библиотеки, разработанной на одном языке программирования может применяться в проекте, строящемся в другом языке программирования. Разработанная DLL-библиотека может использоваться в различных языках программирования, например, в языке C++ (рисунок 4);

5) *мобильность и замена одних DLL-библиотек другими.* Иногда возникает необходимость изменить условие задачи. При этом достаточно будет заменить DLL-библиотеку, содержащую целевую функцию или ограничения;

6) *экономия места на диске.* Многие приложения совместно используют одну DLL-библиотеку, расположенную на диске, т. е. нет необходимости создавать копии DLL-библиотеки в каждом проекте. В каждом приложении, построенном на основе статически компоуемой библиотеки, код библиотеки, напротив, компоуется в исполняемый образ в виде отдельной копии.

Потенциальные недостатки использования DLL-библиотек состоят в следующем:

1) *приложение не является самодостаточным:* его выполнение определяется наличием того или иного библиотечного модуля DLL. Чаще всего проект из-за этого недостатка приходится архивировать для переноса на различные персональные компьютеры [2];

2) *наименьшая защищённость приложения.* Злоумышленник, используя легальное приложение, может подменить DLL-библиотеку, что в итоге при использовании приложения вызовет библиотеку злоумышленника, а не легальную библиотеку, в результате выполнится код злоумышленника.

Имя образа	Польза	ЦП	Память (частн...
opr.exe	Wm-pnn	00	5 348 КБ
opr_no_dll.exe	Wm-pnn	00	5 456 КБ

Рисунок 2 — Использование оперативной памяти приложения с применением DLL-библиотек и без них

```

using System;
using System.Threading.Tasks;
using function;
namespace opr_dll
{
    public class Methods
    {
        public struct Points
        {
            public Double x1, x2, y;
            public struct Point_Line
            {
                public Double x1, x2;
            }
            public Point_Line[] line;
        }
        public Points Perebor(Double x1min = 0, Double x1max = 0, Double x2min = 0, Double x2max = 0, Double _x1_ = 0, Double
        _x2_ = 0)
        {
            Points rezult_func;
            var func = function.function();
            rezult_func.y = 0; rezult_func.x1 = 0; rezult_func.x2 = 0;
            rezult_func.line = new Points.Point_Line[0];
            Double hagx1 = 0, hagx2 = 0;
            int i = 1;
            while (x1min + hagx1 * _x1_ <= x1max)
            {
                while (x2min + hagx2 * _x2_ <= x2max)
                {
                    double y = func.fun(x1min + hagx1 * _x1_, x2min + hagx2 * _x2_);
                    if (y < rezult_func.y)
                    {
                        rezult_func.y = y;
                        rezult_func.x1 = x1min + hagx1 * _x1_;
                        rezult_func.x2 = x2min + hagx2 * _x2_;
                    }
                    i++;
                    hagx2++;
                }
                hagx2 = 0;
                hagx1++;
            }
            return rezult_func;
        }
    }
}

```

Рисунок 3 — Метод прямого перебора в виде файла DLL-библиотеки

```

#include "stdafx.h"
#include "opr_dll.dll"
#include <iostream>

using namespace std;
void main()
{
    double x1min, x1max, x2min, x2max, _x1_, _x2_;
    cin >> x1min; cin >> x1max; cin >> x2min;
    cin >> x2max; cin >> _x1_; cin >> _x2_;
    opr_dll.Methods method = new opr_dll.Methods();
    opr_dll.Methods.Points answer;
    answer = method.Perebor(x1min, x1max, x2min, x2max, _x1_, _x2_);
    cout << "Минимальное значение функции = " << answer.y << "\n";
    cout << "x1 = " << answer.x1 << " ";
    cout << "x2 = " << answer.x2 << "\n";
}

```

Рисунок 4 — DLL-библиотека в C++

Заключение. Предложенный способ создания приложений является перспективным решением при проектировании программного обеспечения. Перечисленные преимущества показывают то, что использование DLL-библиотек решает проблему командной и рутинной разработки, экономии памяти и времени и даёт ряд дополнительных преимуществ.

Список цитируемых источников

1. *Sintes A. Sams Teach Yourself Object-Oriented Programming in 21 Days / A. Sintes.* — 672 s.
2. Преимущества использования DLLs [Электронный ресурс] / Microsoft Developer Network. — Режим доступа: <https://msdn.microsoft.com/ru-ru/library/dtba4t8b.aspx>. — Дата доступа: 03.02.2015. — Загл. с экрана.

Материал поступил в редакцию 25.02.2015 г.

Репозиторий Баргу