

Организация комплекса программ и данных помощи обучаемым программированию

08, август 2015

Алексеев Ю. Е.^{1*}, Куров А. В.¹

УДК: 681.3.069

¹Россия, МГТУ им. Н.Э. Баумана

^{*}lekarru@mail.ru

Введение

Современные системы программирования предоставляют пользователям широкий спектр услуг, позволяющих существенно облегчить разработку программ. Это выявление ошибок времени компиляции, компоновки и выполнения, предоставление шаблонов программ, контекстные подсказки и исчерпывающие справочные системы по языкам программирования и библиотекам подпрограмм, средства обработки исключений и средства отладки.

Для изучающего основы программирования весь этот арсенал услуг представляется весьма сложным, сообщения об ошибках и исключениях не всегда понятны, да и поиск необходимой справки может занять много времени. Кроме того, вопросы алгоритмизации задач, разработки структур данных и проверки соответствия программы заданию на её разработку вообще не относятся к системам программирования.

Назначение комплекса программ и данных помощи обучаемым (сокращенно КПДПО или просто комплекс) – автоматизация различных проверок соответствия разрабатываемых обучаемым приложений на лабораторных и контролях работах и помощи в виде контекстно-зависимых справок и пособий, предоставляемых в файлах различных типов.

Приводится пример использования комплекса при обучении основам программирования на языке C в среде VS C++. Для проверки соответствия проекта заданию используется тестирование, а справочные файлы описывают используемый приём программирования и дают наглядное представление о предметной области.

Технология применения

(основные шаги, подробности – позже).

- 1) Обучаемый - пользователь, получив задание на выполнение лабораторной или контрольной работы, создаёт проект со стандартным шаблоном и готовит его замену на новый шаблон, подготовленный заранее для каждого варианта задания.
- 2) Запускает приложение, в результате чего программы комплекса заменяют шаблон и размещают в папке проекта справочные файлы и файлы, необходимые для выполнения тестирования разрабатываемого приложения.
- 3) Обучаемый разрабатывает программу, используя средства отладки для устранения различного рода ошибок и, при необходимости, справочные файлы комплекса.
- 4) Обучаемый готовит исходный текст программы для автоматического тестирования и запускает приложение.
- 5) Программы комплекса выполняют тестирование по заранее подготовленным исходным данным. При обнаружении несоответствия или соответствия результатов работы приложения пользователем ожидаемым по заданию, обучаемый получает соответствующее сообщение, а работа приложения завершается. В случае несоответствия приложения заданию обучаемый должен продолжить отладку, иначе - представить результат своей работы к защите.
- 6) Преподаватель должен принять защиту, оценить и проставить отметку в файле учета выполнения лабораторной или контрольной работы.

Структура комплекса

Для реализации комплекса выбрана древовидная структура на основе файловой системы Windows. Это соответствует природе её составляющих, привычно для преподавателей и не требует использования специальных программ. Представленное далее описание структуры комплекса построено на отдельных описаниях его папок. Формально описанием папки являются сведения о содержащихся в ней файлах и подчинённых папках, а также сведения о родительской папке. Используемые имена файлов и папок не имеют значения, главное - какие реальные объекты они представляют. Принадлежащие папке файлы с информацией для пользователей могут иметь произвольные расширения (.txt, .doc, .xls, .bmp и т.д.), принятые для документов соответствующих редакторов, но расширениями исполняемых и заголовочных файлов должны быть .exe и .h соответственно. Каждая папка должна может иметь файл с именем "оПапке" (возможно любое расширение) со справкой о назначении папки и её месте в дереве папок комплекса.

Корневая папка имеет имя КПДПО. Будем для примера считать, что она, как и проекты обучаемых, находится в папке D:\temp\. Ей подчинены папки с именами EXE, Н и папки с именами вида УчГодNN-ММ, например, УчГод13-14. Папка с наибольшим ММ представляет текущий учебный год, а остальные, если они есть, - архив. Далее будем пока считать, что архивных папок нет и что текущей является УчГод13-14.

Папки с именами EXE и Н не имеют подчинённых папок и предназначены для хранения исполняемых и заголовочных файлов соответственно, имеющих общее назначение в реализации комплекса. Кроме того, в каждой из них должен быть свой файл "оПапке.txt

", в первой строке которого содержится спецификация самой папки, и файл "Список.txt " со списком имён файлов .exe и файлов .h соответственно.

В подчинении папке УчГод13-14 находятся папки ОСЕНЬ, представляющая первый семестр учебного года, и ВЕСНА, представляющая второй семестр.

Необходимость разделения комплекса по годам и семестрам определяется тем, что от года к году могут обновляться планы учебных дисциплин и/или объекты самого комплекса, но внутри этих периодов обновления запрещены.

Папка ОСЕНЬ (аналогично и папка ВЕСНА) содержат две подчинённых папки с именами КПД и ФАК. Папка КПД является корнем поддерева КПДПО, связанного с организацией учебного процесса, а папка ФАК представляет поддерево, отражающее структуру объектов учебного заведения от факультета до обучаемого и статистические данные по выполнению обучаемым различных видов работ (лабораторных, контрольных). Далее, в соответствующих разделах помимо описаний поддереьев КПД и ФАК для наглядного представления в виде рисунков даны возможные примеры фрагментов этих структур.

Поддерево КПД

Это поддерево (рис. 1 представляет пример его части) содержит методические материалы по различным учебным курсам. На этом рисунке полностью представлена только одна ветвь дерева (остальные имеют аналогичную структуру), файлы "оПапке" не показаны (они должны быть в каждой папке), многоточие обозначает, что список подчинённых папке объектов может быть продолжен. Аналогичные ограничения использовались и для показанного ниже рисунка поддерева ФАК.

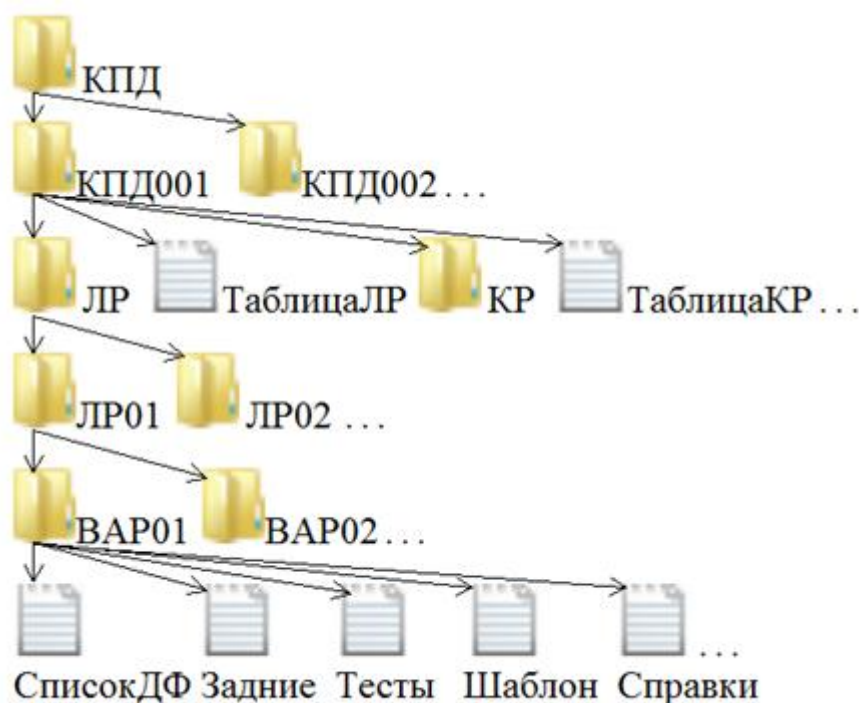


Рис.1. Пример части поддерева КПД

Папка КПД содержит папки с именами КПДnnn, где nnn - уникальный номер части комплекса программ и данных, связанного с конкретной программой учебной дисциплины (её частью - для многосеместровой программы).

Папки с именами КПДnnn в своих файлах "оПапке" должны содержать ссылки на соответствующие программы учебной дисциплины или их части. Например, части комплекса с именем КПД002 может соответствовать первой части программы учебной дисциплины "Информатика" для указанных в ней специальностей, языка и среды программирования (язык С, среда VS C++). Кроме того, для каждой из этих частей должен содержаться список имён групп обучаемых.

Папки КПДnnn содержат также папки видов работ с именами ЛР (лабораторные работы) и КР (контрольные работы), а также файлы с именами "ТаблицаЛР" и "ТаблицаКР", являющиеся заготовками таблиц, копии которых для каждой учебной группы в папках КПДnnn поддерева ФАК будут использоваться для назначения обучаемым вариантов заданий по работам и учета их выполнения.

Папка ЛР (КР) содержит папки с именами ЛРnn (КРnn), где nn - уникальный номер лабораторной (контрольной) работы, а они, в свою очередь, содержат папки с именами ВАРmm, где mm - уникальный номер варианта задания на работу.

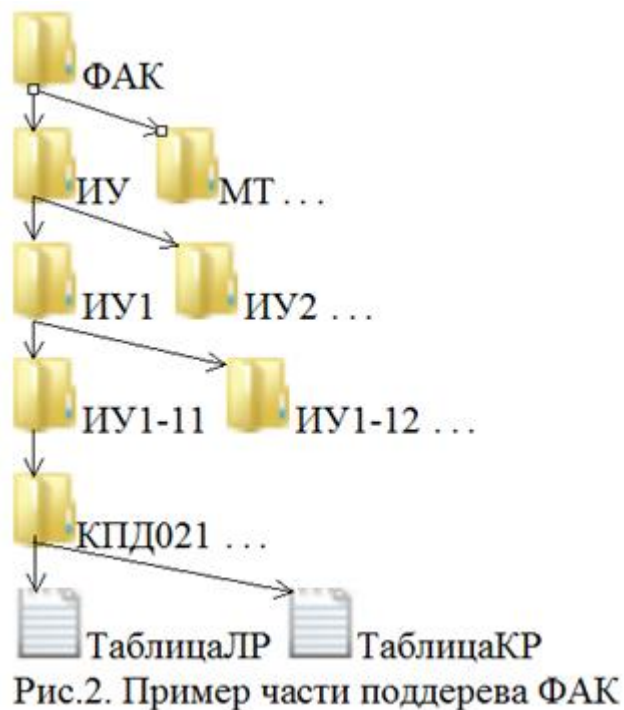
Папка ВАРmm, подчинённая папке ЛРnn, содержит файлы, которые должны быть добавлены в папку проекта обучаемого, выполняющего данный вариант данной лабораторной работы. Это файл с заданием на работу, файлы для тестирования подготовленной обучаемым программы, файл с новым шаблоном, который заменит исходный файл .сpp проекта, файл с контекстной справкой и, возможно, какие-либо другие файлы. Для выполнения копирования все эти файлы, а также путь доступа к содержащей их папке, перечислены в файле "СписокДФ" (список добавляемых файлов).

Аналогично, папка ВАРmm, подчинённая папке КРnn, содержит файлы, которые должны быть добавлены в папку проекта обучаемого, выполняющего данный вариант данной контрольной работы, но состав этих файлов ориентирован на проверку соответствия проекта заданию.

Поддерево ФАК

Это поддерево (рис. 2 представляет возможный пример его части) содержит сведения о каждом из обучаемых - их принадлежность структурным подразделениям учебного заведения, сведения о работах, которые должны быть выполнены обучаемым, а также оценки их выполнения, предоставляемые преподавателем.

Три верхних уровня поддерева ФАК представляют подразделения учебного заведения (факультет, кафедра, группа). Папка ФАК содержит имена факультетов, например, ИУ, МТ и т.д. Папки с именами факультетов, например, ИУ, содержат имена кафедр, например, ИУ1. Папки с именами кафедр, например, ИУ1, содержат имена групп, например, ИУ1-11.



Папки с именами групп содержат папки КПДnnn - копии папок поддерева КПД, представляющие только те части комплекса, которые должны использоваться для конкретной группы (см. описание папки КПДnnn в поддереве КПД).

Папки с именами КПДnnn содержат файлы "ТаблицаЛР" и "ТаблицаКР" - копии одноимённых файлов из папки КПДnnn в поддереве КПД, дополненные преподавателем фамилиями членов группы обучающихся и номерами вариантов для каждой лабораторной (контрольной) работы каждого обучающегося. После выполнения и защиты очередной работы преподаватель должен проставить в такой таблице оценку её выполнения.

Пример выполнения лабораторной работы с использованием комплекса

Условия проведения

- Среда программирования - MS VS 2010 C++, создаваемое пользователем приложение - консольное.
- d:\temp\ - место расположения проекта приложения, создаваемого обучающимся, КПДПО, программы start.exe, инициирующей смену шаблона в проекте обучающегося, заголовочного файла RabotaLR.h с функцией Rus, преобразующей строковые константы С для правильного вывода русских текстов в окно программы.
- Группа ИУ1-11, КПД002, ЛР08, ВАР01 (задание на лабораторную работу упрощенное).

Шаги выполнения

1. Обучаемый, например, Иванов создаёт проект с именем "ИУ1-11К002ИвановЛ08", добавляет в файл "ИУ1-11К002ИвановЛ08.cpp" строки
`#include "windows.h"`

и

```
system("D:\\temp\\start.exe ИУ1-11К002ИвановЛ08");
```

как показано в следующем листинге

```
// ИУ1-11К002ИвановЛ08.cpp : Defines the entry point for the
console application.
//
#include "stdafx.h"
#include "windows.h"
int _tmain(int argc, _TCHAR* argv[])
{
    system("D:\\temp\\start.exe ИУ1-11К002ИвановЛ08");
    return 0;
}
```

и выполняет программу.

2. Программа start.exe извлекает из параметра ИУ1-11К002ИвановЛ08 путь

D:\temp\КПДПО\УчГод13-14\ОСЕНЬ\ФАК\ИУ\ИУ\ИУ1-11\КПД002\, проверяет по списку в файле "ТаблицаЛР" группы ИУ1-11, есть ли обучаемый с такой фамилией, и если нет, то выводит соответствующее сообщение. В этом случае обучаемый должен заново создать проект или обратиться за помощью к преподавателю.

Если проверка фамилии прошла успешно, то программа start.exe извлекает из таблицы "ТаблицаЛР" номер варианта задания для строки с фамилией Иванов и столбца ЛР08 (пусть это будет вариант с номером 01).

Далее программа start.exe находит путь к папке

D:\temp\КПДПО\УчГод13-14\ОСЕНЬ\КПД\КПД012\ЛР\ЛР08\ВАР01\, копирует из неё в папку проекта обучаемого все файлы, перечисленные в файле "СписокДФ" (Шаблон.txt, Тест.txt, Справка.doc, Задание.doc и добавленные в папку для этого варианта файлы Тест.exe и График.xls, назначение которых будет рассмотрено на следующих далее шаге 3. и в конце текста статьи соответственно), и потом заменяет текст в файле "ИУ1-11К012ИвановЛ08.cpp" текстом из файла Шаблон.txt.

Для завершения этих подготовительных работ обучаемый должен закрыть окно программы start.exe и в появившемся окне диалога щелкнуть кнопку "Yes to All", после чего на вкладке ИУ1-11К012ИвановЛ08.cpp редактора текста VS C++ появится новый шаблон, представленный в следующем листинге:

```

// новый ШАБЛОН_ИНФОРМАТИКА_СЕМ1_ЛР08_ВАР01
// ИУ1-11К002ИвановЛ08ВАР01
/*
    ЗАДАНИЕ
    Составить программу вычисления максимального Ymax и
    минимального Ymin значений функции  $Y=X^3-18X^2-10X+7$  и Xmax и Xmin
    – соответствующих значений аргумента, изменяемого на интервале
    от -4 до 16 с шагом 0,01.
    ! Использовать приём программирования "Поиск максимума
    (минимума) функции".
    ! Автоматически тестировать программу только после отладки
    кода вычисления Ymax, Ymin, Xmax, Xmin.
*/

#include "stdafx.h"
#include <stdlib.h>
#include "windows.h"
#include "d:\\temp\\РАБОТА_ЛР.h"

int _tmain(int argc, _TCHAR* argv[])
{
    //- #define test
    int j,i,k;
    double Ymax, Ymin, Xmax, Xmin, X, Y ;
    //ДАЛЕЕ РАЗМЕСТИТЕ КОД ВЫЧИСЛЕНИЯ Ymax, Ymin, Xmax, Xmin

    #include "ТЕСТ.txt"
    //ДАЛЕЕ РАЗМЕСТИТЕ ВЫВОД РЕЗУЛЬТАТОВ

    return 0;
}

```

3. Обучаемый добавляет в соответствующие места нового шаблона код вычисления *Ymax*, *Ymin*, *Xmax*, *Xmin* и оператор вывода результатов, отлаживает и переходит к автоматическому тестированию. Для этого достаточно поставить курсор в строке `// - #define test` программы перед символом `#` и нажать клавишу Enter, а затем, как обычно, запустить программу. При подготовке программы к компиляции в её текст будут добавлены директивой `#include Тест.txt` строки файла тестирования `ТЕСТ.txt`, расположенные между директивами `#ifdef test` и `#endif`, представленный ниже:

```

#ifdef test
    char stest[80]; sprintf(s, "    Тест.exe    %lf    %lf    %lf%lf",
Ymax,Ymin,Xmax,Xmin);

```



```

switch (system(stest.exe))
{
    case 1:{printf(Ruc("ОШИБКА ВЫЧИСЛЕНИЯ Ymax. см. файл СПРАВКА,
пункт 1. \n")); break;}
    case 2:{printf(Ruc("ОШИБКА ВЫЧИСЛЕНИЯ Ymin. см. файл СПРАВКА,
пункт 1. \n")); break;}
    case 3:{printf(Ruc("ОШИБКА ВЫЧИСЛЕНИЯ Xmax. см. файл СПРАВКА,
пункт 2. \n")); break;}
    case 4:{printf(Ruc("ОШИБКА ВЫЧИСЛЕНИЯ Xmin. см. файл СПРАВКА,
пункт 2. \n")); break;}
    default:printf(Ruc("НЕ ОБНАРУЖЕНО ОШИБОК\n"));
}
printf(Ruc("Для продолжения нажмите Enter!\n"));
getchar();
return 0;
#endif

```

Проверяет результаты вычислений программа Тест.exe, которая вместе с файлом Тест.txt была скопирована в папку проекта обучаемого. Программа Тест.exe скрывает от обучаемого данные и механизм проверки, а возвращает целое число - номер ошибки или 0, если ошибок не обнаружено.

4. В случае неверных результатов обучаемый получит сообщение с указанием переменной, неправильно вычисленной, и должен продолжить отладку программы. Если же при тестировании ошибок обнаружено не будет и комплекс не содержит других средств для проверки соответствия проекта заданию, то обучаемый должен защитить работу - представить её преподавателю и ответить на задаваемые им вопросы по изучаемой теме. Преподаватель может потребовать переделать код программы, если не был использован указанный в задании математический метод, алгоритм или приём программирования. В случае успешной защиты преподаватель должен дать оценку выполнения работы обучаемым и отметить в файле "ТаблицаЛР", в простейшем случае 1 - "зачтено" или 0 - "не зачтено".

Основная часть задания содержится в новом шаблоне, но файл Задание.doc может содержать какую-либо дополнительную информацию, поясняющую его суть.

В этом примере файл Справка.doc содержит описание приёма программирования, который по заданию должен использоваться при выполнении лабораторной работы:

Приём программирования "Поиск максимума Y_{max} (минимума Y_{min}) функции $Y=F(X)$ и соответствующего значения аргумента X_{max} (X_{min}) на значениях аргумента, изменяемых на отрезке $[A,B]$ с шагом H "

Алгоритм

- a. Переменной Y_{max} (Y_{min}) присвоить значение $F(A)$ и X_{max} (X_{min}) присвоить значение A .
- b. Организовать цикл с параметром X , изменяющимся с шагом H от $A+H$ до $B+H/2$, внутри которого следует выполнять следующие действия:
 - вычислить для текущего значения X значение $Y=F(X)$,
 - сравнить Y с Y_{max} (с Y_{min}), и если окажется, что $Y > Y_{max}$ ($Y < Y_{min}$), то Y_{max} (Y_{min}) присвоить значение Y и X_{max} (X_{min}) - значение X .

Файл График.xls даёт наглядное представление (рис. 3) о значениях $F(X)$ на заданном интервале, что позволит обучаемому самостоятельно оценить работу своей программы при её отладке.

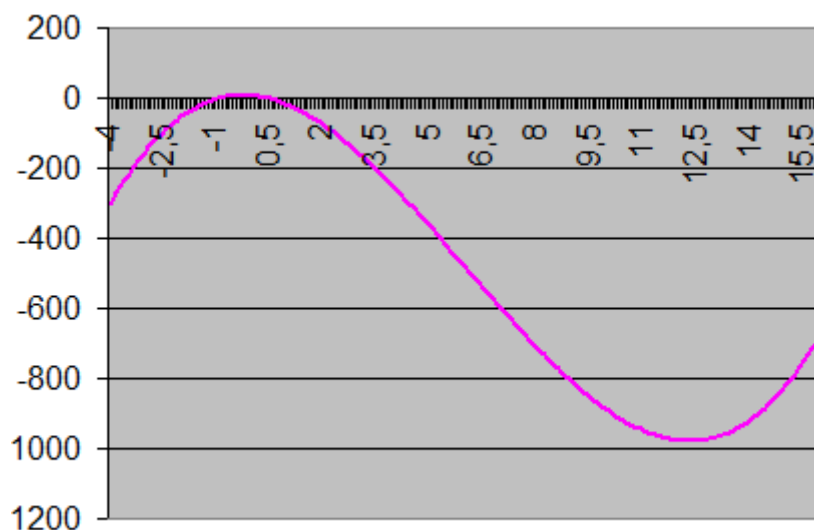


Рис. 3. График функции на заданном интервале

Список литературы

1. Алексеев Ю.Е., Куров А.В. Автоматизация тестирования студенческих программ. // Инженерный журнал: наука и инновации. Электронное научно-техническое издание. 2013. вып. 6 (18). 6 с. Режим доступа: <http://engjournal.ru/catalog/it/hidden/768.html> (дата обращения: 10.07.2015)
2. Алексеев Ю.Е., Ваулин А.С., Куров А.В. Практикум по программированию. Обработка числовых данных: учеб. пособие для вузов / Под ред. Б.Г. Трусова. М.: Изд-во МГТУ им. Н.Э. Баумана. 2008. 288 с.