



ЛІТЕРАТУРА

1. Есипова Н.Д. Развитие логического мышления младших школьников на внеклассных занятиях по математике с использованием ЭВМ. Научная библиотека диссертаций и авторефератов disserCat <http://www.dissercat.com/content/razvitie-logicheskogo-myshleniya-mladshikh-shkolnikov-na-vneklassnykh-zanyatiyakh-po-matemat#ixzz4bgCHqioz>
2. Сенкевич Г. Компьютер для людей с ограниченными возможностями. - СПб.: БВХ-Петербург, 2014. – 320 с.: ил
3. Натан А. WPF 4. Подробное руководство. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с.
4. Чарльз Петцольд. Программирование для Microsoft Windows 8 (6 издание). Разработка приложений для Windows Store на C# и XAML.– СПб.: Питер, 2014. – 1008 с.
5. Управление воспроизведением. [Электронный ресурс]. Режим доступа: https://professorweb.ru/my/WPF/graphics_and_animation/level15/15_9.php — Загл. с экрана.

УДК 004.91

ИССЛЕДОВАНИЕ ОШИБОК ВО ВХОДЯЩИХ ДАННЫХ ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ В ОБРАЗОВАНИИ

Д.С. Ковальчук¹, В.В. Герасимов²

¹ студент 4 курса, кафедра электронных вычислительных машин, Днепровский национальный университет им. Олеся Гончара, г. Днепр, Украина, e-mail: dimon_koval2612@ukr.net

² ассистент кафедры электронных вычислительных машин, Днепровский национальный университет им. Олеся Гончара, г. Днепр, Украина, e-mail: gerasimov@dsu.dp.ua

Аннотация. В работе приведены анализ и решение особенностей и сложностей в организации процесса формирования унифицированной отчетной документации об успеваемости студентов посредством специализированной информационной системы.

Ключевые слова: программа, информационная система, ошибка, данные, отчетный документ, способ исправления ошибок.

RESEARCH OF INCOMING DATA MISTAKES OF INFORMATION SYSTEMS IN EDUCATION

D.S. Kovalchuk¹, V.V. Gerasimov²

¹ student, Department of Computer, Oles Honchar Dnepr National University, Dnepr, Ukraine, e-mail: dimon_koval2612@ukr.net

² assistant, Department of Computer, Oles Honchar Dnepr National University, Dnepr, Ukraine, e-mail: gerasimov@dsu.dp.ua



Abstract. The paper analyzes and solves the specifics and complexities in the organization of the process of forming a unified reporting documentation on student performance through a specialized information system.

Keywords: program, information system, mistake, data, reporting document, method of error correction.

Введение. Создание единой государственной электронной базы по вопросам образования повлекло за собой необходимость в автоматизации обработки больших объемов отчетной информации и ее унификации, и как следствие, повсеместное внедрение программного обеспечения, упрощающее работу в высших учебных заведениях. Внедрение информационных систем для обработки данных об учебном процессе в высших учебных заведениях, а также для создания отчетной документации на основе обработанной информации в данное время сопровождается рядом особенностей и сложностей, которые усложняют структуру собственно программного обеспечения, требуя контроля входящей информации на наличие ошибок человеческого фактора, полученных в ходе сбора данных, а также первоначального форматирования неунифицированных данных для дальнейшей обработки и их сохранения для работы в будущем. Данные особенности заключаются в отсутствии жесткого контроля при создании сводного файла, который содержит начальные данные для дальнейшей обработки, а также в ошибках, допускаемых операторами, работающими вне указанных информационных систем, как например, ошибки в именах, наименованиях дисциплин, и прочих названиях, которые могут претерпевать вариации в зависимости от учебного процесса в разных высших учебных заведениях и факультетах.

Исследование данных особенностей и сложностей проводилось на информационной системе, предназначенной для форматирования отчетной документации по результатам обучения студентов кафедры ЭВМ, получающих сертификаты о полученной квалификации в высшем учебном заведении, для единой государственной электронной базы по вопросам образования.

Цель работы. Целью данной работы есть анализ возможных сложностей в создании и работе информационной системы, предназначенной для форматирования отчетной документации по результатам обучения студентов кафедры ЭВМ, вызванных ошибками в данных, поступающими на обработку данной системой, и структуре сводного электронного файла, в котором хранятся эти данные, а также предложение способов и методов обработки и разрешения этих сложностей.



Материал и результаты исследований. Для исследования особенностей и сложностей в работе информационных систем в образовании, основанных на ошибках в данных, с которыми собственно и производится работа, была взята программа, основной задачей которой является создание отчетного документа с оценками выпускников кафедры ЭВМ для единой государственной электронной базы по вопросам образования (далее — ЕГБО). В ходе создания данной программы было обнаружено ряд особенностей и сложностей, которые усложняли алгоритм работы программы и процесс обработки поступающих данных, а именно:

- 1) написание имен студентов в формате «Фамилия и инициалы», в то время, как унифицированный документ для ЕГБО требует написания полного имени в формате «Фамилия Имя Отчество»;
- 2) ошибки в названиях дисциплин (грамматические ошибки, сокращения, аббревиатуры, некоторые дополнения в скобках);
- 3) ошибочный формат в поле для оценки, из-за которого возникает пропуск чтения оценки из файла;
- 4) многократное использование отступов для переноса части текста на новую строку.

Все вышеперечисленные проблемы приводили к ошибкам и трудностям в ходе работы программы:

- 1) невозможно было поставить в соответствие имена студентов из сводного файла с поступающими данными и файлом ЕГБО, где имена уже указаны и требуется перенести оценки в определенном формате;
- 2) невозможно было поставить в соответствие название дисциплин из сводного файла с поступающими данными и файлом ЕГБО, где названия дисциплин также уже указаны;
- 3) оценки студентов из-за неверного формата в сводном файле с входящими данными невозможно было считать, что приводило к потере данных и ошибкам в работе программы;
- 4) ошибки и пренебрежение стандартным форматированием сводного файла усложняли понимание обрабатываемых данных и их дальнейшее использование.

Для предотвращения и исправления вышеперечисленных ошибок было выбрано два способа их обработки:

- 1) предотвращение попадания не отформатированных данных в программу и их предварительное форматирование на этапе загрузки в память;
- 2) возможность редактирования уже загруженных данных в программу при помощи элементов управления во время работы с программой.

Первый способ заключался в обработке названий дисциплин на наличие двух и более отступов подряд, а также дополнительной информации в

скобках после собственно названия, которая не входит в официальное название дисциплины. Для этого был разработан метод `string formatStr(string str)`, который удалял лишние отступы внутри и в конце строки и данные, которые были указаны в скобках после названия дисциплины. Предполагаемые данные для удаления были взяты по примеру, указанному в предоставленном сводном файле за 2016 год, а именно указание некоторой цифровой информации или указание вида оценивания. Ниже приведен листинг метода `string formatStr(string str)`:

```
public static string formatStr(string str)
{
    string newStr = "";
    for (int i = 0, j = 0; i < str.Length; i++)
    {
        if(str[i] == '(')
        {
            if (((str[i + 1] == '1') || (str[i + 1] == '2') || (str[i + 1] == '3')
            || (str[i + 1] == '4') || (str[i + 1] == '5') || (str[i + 1] == '6')
            || (str[i + 1] == '7') || (str[i + 1] == '8') || (str[i + 1] == '9'))
            || ((str[i + 1] == 'д') && (str[i + 2] == 'и') && (str[i + 3] == 'ф'))))
            {
                break;
            }
        }
        if (i != str.Length - 1)
        {
            if ((str[i] == ' ') && (str[i + 1] == ' '))
            {
                continue;
            }
            else
            {
                newStr += str[i];
                j++;
            }
        }
        else
        {
            newStr += str[i];
        }
    }
}
```

```
return newStr.Trim();  
}
```

Данный метод вызывается при первом прочтении строки из поля в сводном файле, обрабатывает входящую строку `str` формата `string` [1] и либо после форматирования в случае наличия вышеперечисленных ошибок, либо после отрицательного результата проверки всей строки на наличие данных ошибок возвращает строку [2], которой программа оперирует в дальнейшем.

Данный метод не предусматривает редактирование грамматических ошибок и исправление иных ошибок и неточностей в собственно официальных названиях, как например наличие цифр в названии в середине или начале названия или иных символов, не входящих в официальное название.

Для исправления уже загруженных в программу данных, которые не поддаются программному форматированию и исправлению, было предусмотрено возможность их динамического исправления и редактирования данных в элементах управления [1] во время работы в программе, в частности в таком элементе, как `dataGridView` [1]. Этот элемент позволяет редактировать данные и в случае редактирования того или иного поля генерирует событие, которое оповещает программу о редактировании путем установления специального флага (булевой переменной [2]) в значение `true` (истина), что позволяет сохранять измененные данные.

Пример данного элемента управления представлен на рис. 1, где содержится информация о дисциплинах.

Предмет	Година	ECTS	Код
Програмні та інструментальні засоби web-дизайну	180	0	
Теорія ймовірності та мат. статистика	36	0	
Електромагнітні вимірювання	162	0	
2 Архітектура комп'ютерів	360	0	
Операційні системи	252	0	
Цифрова обробка сигналів	126	0	

Рисунок 1 – Пример элемента управления `dataGridView` с ошибками

Как видно из примера, после прочтения всего сводного файла программа получает набор названий без дополнительной информации и лишних отступов как в середине строки, так и в конце. Но в то же время сохраняются сокращения, которые противоречат официальным названиям дисциплин и определенные данные, для отслеживания которых метод `string formatStr(string str)` не был предназначен. Все эти ошибки и неточности исправляются оператором вручную. Так в данном примере было исправлено

название «Теорія ймовірності та мат. статистика» на «Теорія ймовірності та математична статистика», а «2 Архітектура комп'ютерів» на «Архітектура комп'ютерів», что показано на рис. 2.

Данный способ обработки неточностей и ошибок в данных производится в данной программе для разного набора данных, а именно названий дисциплин, учебных и производственных практик и т.д.

Дисципліни, з котрих студенти отримували залік/складали іспит

Предмет	Години	ECTS	Код
Програмні та інструментальні засоби web-дизайну	180	0	
Теорія ймовірності та математична статистика	36	0	
Електрорадіовимірювання	162	0	
Архітектура комп'ютерів	360	0	
Операційні системи	252	0	
Цифрова обробка сигналів	126	0	

Рисунок 2 – Пример элемента управления dataGridView без шибок

Для исправления ошибок в фамилиях студентов, а также написания их имен в формате «Фамилия Имя Отчество» был использован элемент управления textBox [1], в котором отображается имя студента при работе с его данными в программе. При изменении его имени и переключении на другой набор данных (при смене студента, с которым производится работа), отредактированное имя сохраняется. Пример исправления имени изображен на рис. 3.

Студент (прізвище та ініціали)	Повне ім'я
Волкова А.О.	Волкова А.О.
Студент (прізвище та ініціали)	Повне ім'я
Волкова А.О.	Волкова Анна Олегівна

Рисунок 3 – Пример исправления формата имени

Данный метод обработки ошибок не гарантирует полного исправления ошибок в виду человеческого фактора, а также не исключает допущения новых ошибок. В перспективе дальнейшей разработки и поддержания функционирования программы рассматривается добавление функционала для сравнения названий дисциплин, полученных в ходе работы программы, и тех, что содержатся в файле для ЕГБО. Также данный способ может быть применен для сравнения имен студентов с целью предотвращения потери данных, так как занесение данных, а именно оценок, в файл ЕГБО производится путем поиска столбца с названием необходимой дисциплины и

строки с именем определенного студента. При обнаружении ошибки будет произведено изменение названия или имени и сохранение новых данных. На рис. 4 изображена форма, которая предусматривает проверку ошибок и их исправление перед занесением данных в файл ЕГБО. Здесь элемент управления `textBox` [1] отвечает за возможность редактирования имени названия и названия дисциплины.

Также будет проведено ручную сортировку списков имен студентов и названий дисциплин для удобной их записи данных в файл для ЕГБО. Это будет произведено путем перемещения строк в нужном порядке в элементе управления `listBox` [1]. На рассматриваемой форме есть по два списка для имен студентов и названий дисциплин, где левый список – это данные, полученные в ходе работы программы, а правый – это перечень, считанный из файла ЕГБО. В результате перемещения строк по левым спискам при помощи кнопок (элемент управления `button` [1]) будет выставлен нужный порядок записи данных в файл для ЕГБО.

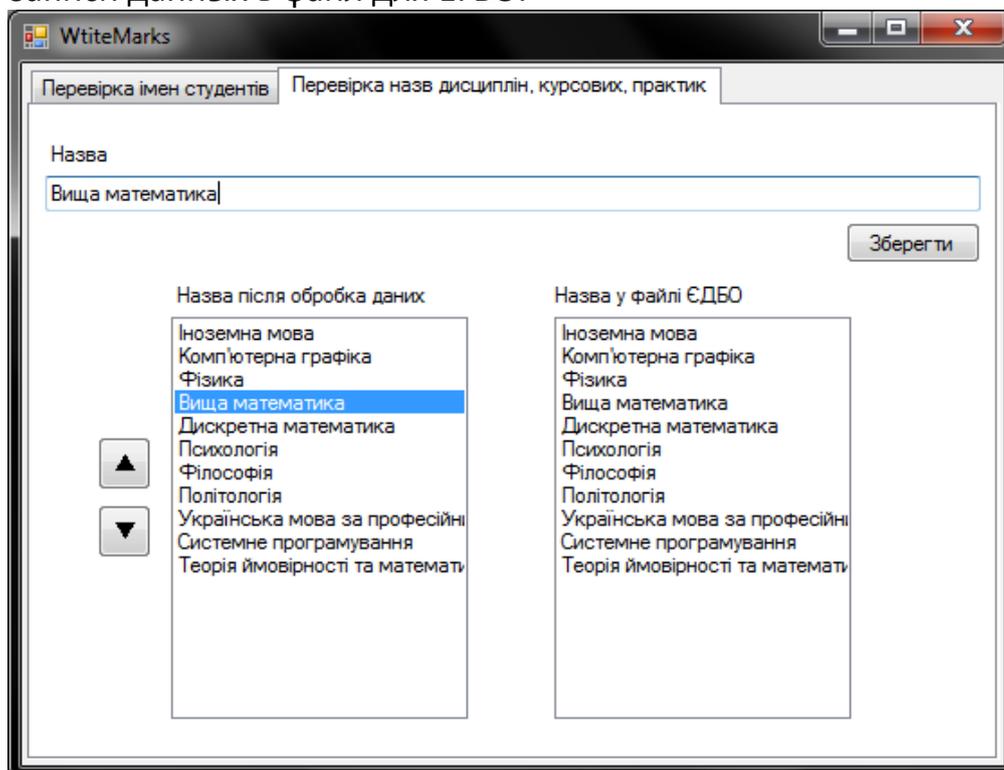


Рисунок 4 – Форма для проверки имен студентов и названий дисциплин перед записью оценок в выходящий файл ЕГБО

Данный способ будет давать возможность оператору, работающему с программой, несмотря на ошибки с легкостью передать обработанные данные в выходящий файл, при этом найденные ошибки могут быть исправлены путем редактирования имени или названия в текстовом поле элемента управления `textBox` [1].

Все способы исправления ошибок так или иначе не являются полными методами исправления ошибок в данных, поступающих в работу программы, но дают возможность оператору их выявить и внести изменения для корректной работы программы и удачного завершения формирования отчетного документа для ЕГБО.

Вывод. В результате исследования особенностей и сложностей в создании и работе программы направленной на формирование отчетной документации, в данном рассмотренном случае формирования сводного документа для ЕГБО было выявлено, что программно сложно отслеживать все виды ошибок и неточностей, а попытки их предусмотреть могут значительно усложнить алгоритм обработки поступающих данных. Большое количество данных все так же предстоит обрабатывать оператору, работающему с программой, с целью проверки данных на ошибки, неточности и несоответствия установленному формату определенных данных. В перспективе для рассмотренной информационной системы будет добавлен функционал, позволяющий на последнем шаге обработки данных и подготовки их к занесению в файл ЕГБО внести последние коррективы или же организовать работу системы несмотря на ошибки, путем установления нужного порядка занесения данных при помощи сортировки объектов, содержащих данные для записи.

Как видно из проведенного исследования, ошибки во входящих данных, а также отсутствие жесткого контроля при создании сводного файла с входящими данными очень сильно усложняет и замедляет процесс обработки данных и подготовку отчетной документации, а также усложняет алгоритм работы программы, требуя проверки входящих данных на предмет особенностей, которые не отвечают стандартам и правилам.

ЛИТЕРАТУРА

1. Байдачный С. С. .NET Framework 2.0: Секреты создания Windows-приложений. М.: «Солон-Пресс», 2006. 519 с.
2. Шилдт Г. C# 4.0. Полное руководство. / Пер. с англ. М. : ООО "И. Д. Вильямс". 2015. 1056 с.