

**ПОДСИСТЕМА КОНТРОЛЯ ЗНАНИЙ ПО ТЕМЕ  
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ DELPHI. РАБОТА  
С ЗАПИСЯМИ»**

---

Описываются результаты разработки программного и информационного обеспечения подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Приводятся результаты тестирования подсистемы контроля знаний, которые показывают, что разработанный программный продукт работает в соответствии с техническим заданием. Показываются примеры работы программы, демонстрирующие корректность ее функционирования. Эта подсистема может быть использована в учебном процессе для оперативного контроля знаний студентов как автономно, так и в составе более полной системы. Проводится экспериментальная проверка эффективности использования этой разработки. Проверка подтвердила правильность гипотезы исследования.

**1. Введение.** В последние годы методы автоматизированного контроля знаний студентов получают все большее распространение. Такой метод контроля знаний не может претендовать на объективную и всеобъемлющую оценку знаний студентов. Он никогда не заменит такие формы контроля, при которых выявляется творческое восприятие знаний, способность студентов самостоятельно мыслить и формулировать ответы на поставленные вопросы, умение применять полученные знания на практике (семинары, обсуждения, практические занятия, лабораторные работы, зачеты, экзамены).

Тестирование может применяться в учебном процессе для оперативного контроля студентов, «грубой» оценки усвоения полученных знаний и даже как косвенный способ закрепления пройденного материала. Таким образом, метод контроля знаний с помощью тестирования является весьма актуальным. Нет сомнений, что и в дальнейшем он будет получать все большее распространение (совместно с другими формами контроля).

Создание систем тестирования естественным образом распадается на две основные задачи: 1) разработка соответствующего программного обеспечения; 2) разработка набора вопросов и вариантов ответов к ним (информационного обеспечения). От преподавателя зависит, будет ли тест содержать достаточное количество вопросов, насколько «правдоподобно» сформулированы варианты ответов.

Работа посвящена решению двух задач – разработке программного и информационного обеспечения подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями».

**2. Основное содержание.** Как уже отмечалось, подсистема контроля знаний состоит из двух составляющих – программного и информационного обеспечения. К каждой из этих составляющих могут быть предъявлены определенные требования [1]. Прежде всего, программное обеспечение подсистемы контроля знаний должно исключить возможность, когда студент может просто «механически» запомнить, какие номера вариантов ответов являются правильными. Чтобы исключить такую возможность, подсистема контроля знаний при каждом новом запуске теста должна предлагать варианты ответов к вопросам в случайном порядке. Еще лучше будет, если и вопросы будут каждый раз появляться в случайном порядке. В этом случае учащемуся придется запоминать и содержание вопроса, и содержание правильного ответа к нему, что уже само по себе способствует содержательному усвоению материала.

Программное обеспечение должно давать студенту возможность возвращаться к уже пройденным вопросам.

Следующим требованием к программному обеспечению подсистемы контроля знаний является проверка того, что студент полностью ответил на все вопросы теста. Иначе возможно получение положительной оценки, в то время как отсутствие ответов на какие-либо вопросы свидетельствует о наличии у учащегося пробелов в знаниях по заданной теме.

Весьма важным требованием к программному обеспечению является возможность преподавателю по окончании тестирования просмотреть, какие именно варианты ответов выбирал студент. Во-первых, это позволяет выявить типичные ошибки, которые связаны со сложностью материала или с недостатками в преподавании. Во-вторых, результаты тестирования полезно обсудить со студентами, чтобы помочь им разобраться в своих ошибках и лучше усвоить материал.

Основным требованием к информационному обеспечению является наличие достаточного количества вопросов по заданной теме. Это необходимо для того, чтобы снизить вероятность случайного получения положительной оценки и повысить точность и объективность результатов тестирования. Можно утверждать, что минимальное количество вопросов в тесте должно быть в два раза больше, чем максимальный балл, по которому оцениваются знания (например, при пятибалльной системе в тесте должно быть как минимум десять вопросов, при 12-балльной системе – 24 вопроса).

По этой же причине может быть выдвинуто требование к количеству вариантов ответов. Оптимальное количество вариантов ответов на каждый вопрос составляет от 4 до 6. Меньшее количество вариантов повышает вероятность случайного угадывания правильного ответа, большее – затрудняет студенту правильное осмысление содержания вариантов ответов. Обязательным требованием к ошибочным вариантам ответов является их «правдоподобность». Ответы должны быть сформулированы таким образом, чтобы студент не мог чисто логическими умозаключениями отбросить явно неподходящие варианты. Они должны быть «похожими» на правильный ответ.

Вполне понятным требованием является полнота теста, т.е. вопросы должны по возможности охватывать весь материал по заданной теме.

Таким образом, в результате проведенного анализа можно сформулировать следующие общие требования к подсистеме контроля знаний, которые необходимо учитывать при ее разработке [2] :

– требования к программному обеспечению:

1) программа должна обеспечивать появление вариантов ответа на каждый вопрос в случайном порядке. Вопросы и варианты ответов к ним должны появляться в случайном порядке;

2) давать возможность студенту возвращаться к уже пройденным вопросам и при необходимости изменять выбор варианта ответа;

3) обеспечивать требование ответа на все вопросы теста;

4) программа должна давать возможность преподавателю после окончания тестирования просмотреть, какие варианты ответов выбирал студент;

– требования к информационному обеспечению:

1) тест должен содержать достаточное количество вопросов (как минимум в два раза больше, чем максимальный балл);

2) вопросы теста должны наиболее полно охватывать всю тему;

3) для каждого вопроса должно быть не менее четырех вариантов ответа;

4) варианты ответов должны выглядеть «правдоподобными».

Проведенный анализ показал, что возможны, как минимум, два подхода к организации данных. При первом подходе данные теста (вопросы, варианты ответов, информация о правильных ответах) хранятся в исходном коде программы. Основное преимущество здесь состоит в том, что эти данные полностью защищены от внешнего «изучения», в результате которого можно было бы получить информацию о правильных ответах. Студент работает только с одним исполняемым файлом (.exe) без каких-либо дополнительных данных. Главный недостаток такого подхода заключается в том, что для изменения имеющегося теста или разработки нового необходимо иметь исходный текст программы и уметь вносить в него необходимые изменения.

Второй подход предусматривает хранение данных теста во внешних файлах. Основное преимущество такого подхода состоит в универсальности программного обеспечения. Один раз откомпилированная программа может использоваться как основа для любого теста. Разрабатывается и изменяется только информационное обеспечение, которое может готовиться с помощью обычных текстовых редакторов. Однако в этой универсально-

сти заключается и основной недостаток. Если внешние данные оставить в исходном виде, то они могут быть легко прочитаны и легко может быть получена информация о правильных ответах. Чтобы этого избежать, данные необходимо «шифровать» (кодировать), а это вызывает необходимость дополнительного программного обеспечения.

В данной работе выбраны следующие параметры программного и информационного обеспечения:

- данные теста хранятся в исходном коде программы;
- тест рассчитан на 10 вопросов;
- к каждому вопросу предлагается 4 варианта ответа, только один из которых является правильным;
- программа рассчитана на оценку по 5-балльной системе;
- программа всегда предьявляет варианты ответов в случайном порядке (вопросы следуют в одном и том же порядке);
- программа предусматривает переходы от одного к вопросу к другому в любом направлении (от первого к последнему или наоборот) с возможностью изменений выбора варианта ответа.

Разработка программного обеспечения осуществлена в среде Delphi.

В качестве диалогового окна подсистемы контроля знаний была разработана следующая форма (рис. 1).

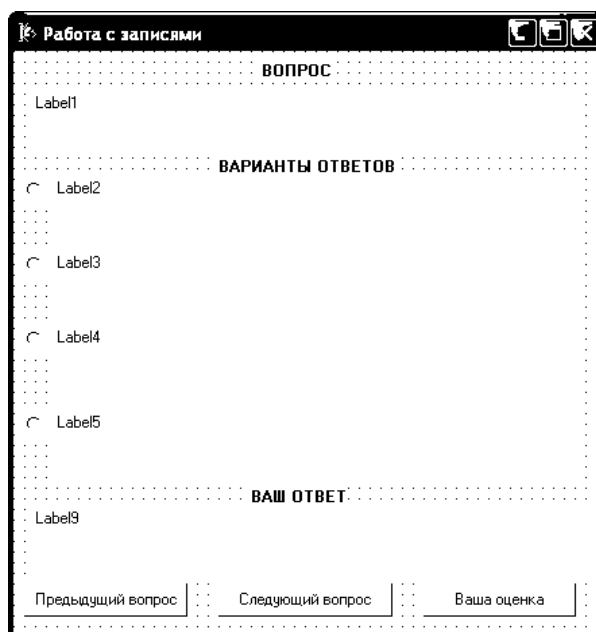


Рис. 1. Диалоговое окно подсистемы контроля знаний

Назначение элементов управления следующее. Метка Label1 используется для отображения текущего вопроса. Метки Label2 – Label5 содержат четыре варианта ответа на текущий вопрос. Метки Label6 – Label8 служат для отображения заголовков частей окна (надписи «Вопрос», «Варианты ответов» и «Ваш ответ»). В метке Label9 будет отображаться выбранный учащимся вариант ответа. Элементы управления RadioButton1 – RadioButton4 позволяют студенту указать выбранный вариант ответа. При разработке программы были выбрана следующая структура данных. Прежде всего, были определены следующие глобальные переменные:

1) строковый массив Vopros, содержащий вопросы теста и варианты ответов на каждый вопрос (ниже приведен фрагмент описания массива):

```
var  
  Vopros:array[1..50] of string=  
  (' Что такое «запись» в языке Delphi? ');
```

‘Это структура данных, которая состоит из отдельных именованных’ +  
’ компонент различного типа’,

‘Это синоним типа символьной строки’,

‘Это структура данных, содержащая набор (массив) символьных строк’,

‘Это специальная структура, в которую необходимо помещать данные’ +  
‘ перед их записью в файл’,

‘Что такое поля записи?’

‘Это именованные компоненты различного типа, из которых состоит запись’,

‘Это особые дополнительные данные, которые описывают структуру записи’,

Этот массив построен следующим образом. Первый элемент массива содержит первый вопрос. В следующих четырех элементах (со 2-го по 5-й) содержатся варианты ответа на этот вопрос. Далее идет второй вопрос, а в четырех следующих за ним элементах содержатся варианты ответа к нему и т. д. Таким образом, весь массив разбивается на группы по пять элементов, в первом из которых содержится вопрос, в четырех следующих за ним элементах – варианты ответа к нему;

2) строковый массив Prav, содержащий правильные ответы:

Prav: array [1..10] of string =

‘Это структура данных, которая состоит из отдельных именованных’ +  
’ компонент различного типа’,

‘Это именованные компоненты различного типа, из которых состоит запись’,

‘В разделе type’,

‘record, end’,

3) строковый массив Otvet, в который в процессе работы будут заноситься выбранные учащимся варианты ответов:

Otvet: array[1..10] of string=(‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’, ‘’);

4) целочисленная переменная NV, которая в процессе работы будет содержать номер текущего вопроса:

NV: integer;

5) также был объявлен новый тип, описывающий целочисленный массив из четырех чисел:

type

AR4 = array [1..4] of integer;

Исходя из основных требований к программному обеспечению, были разработаны процедуры: Rand4 (var RA:AR4), ShowVopros.

В работе выполнено программирование событий FormCreate, Button1Click, Button2Click, Button3Click.

Выполнена также разработка информационного обеспечения. Приведен полный список вопросов теста и варианты ответов к ним. В результате проведенного исследования разработано программное и информационное обеспечение в соответствии с выдвинутыми ранее требованиями.

Разработанная Delphi-программа имеет простой и понятный интерфейс. Вся работа с программой осуществляется только при помощи мыши.

После запуска программы на экране появляется диалоговое окно программы с первым вопросом теста. Варианты ответов приведены в случайном порядке (т.е. при других запусках программы порядок ответов будет другим) (рис.2).

В заголовке окна отображается название теста. Ниже – номер текущего вопроса. Правильный (по мнению учащегося) вариант ответа указывается путем нажатия левой клавишей мыши в белом кружочке, расположенном возле выбранного ответа. Выбранный вариант отображается в нижней части окна (поле «Ваш ответ») (рис. 3).

В нижней части окна находятся две кнопки с надписями «Предыдущий вопрос» и «Следующий вопрос». С помощью этих кнопок студент может перейти к следующему вопросу или вернуться к предыдущим (например, чтобы изменить выбор варианта ответа).

Когда все вопросы пройдены, появляется следующее сообщение (рис. 4).

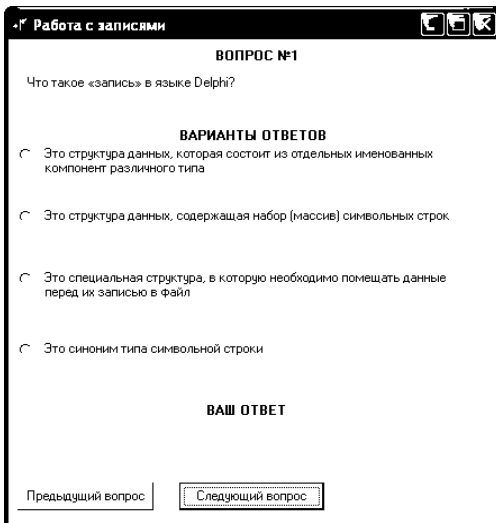


Рис. 2. Отображение первого вопроса теста после запуска программы

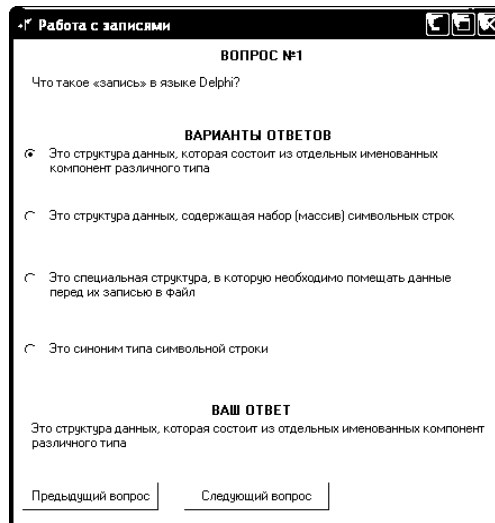


Рис. 3. Выбор варианта ответа

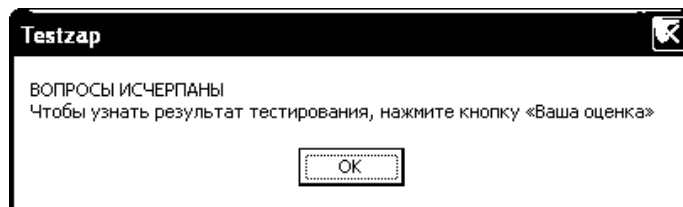


Рис. 4. Сообщение о том, что все вопросы исчерпаны

В нижней части окна появляется кнопка «Ваша оценка».

До того, как нажать на эту кнопку для получения результата тестирования (т.е. оценки), студент может еще раз просмотреть свой выбор вариантов ответов по всем вопросам и, при необходимости, изменить его. После нажатия на кнопку «Ваша оценка» уже ничего нельзя будет изменить.

Если на какой-либо вопрос не был выбран вариант ответа (вопрос был пропущен), об этом будет выдано соответствующее сообщение и вопрос будет вновь предъявлен для ответа (рис. 5).

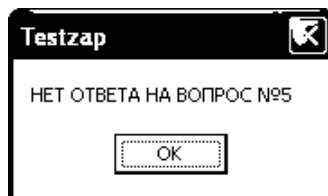


Рис. 5. Сообщение о том, что на определенный вопрос не был дан ответ

Когда на все вопросы будут даны ответы, будут выданы результаты тестирования, которые должны быть предъявлены преподавателю (до нажатия кнопки ОК) (рис. 6). После нажатия кнопки ОК происходит окончательный выход из программы (окно закрывается).

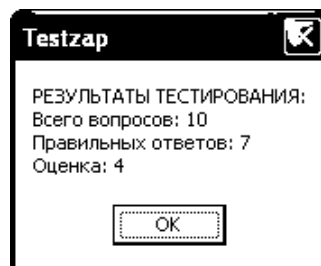


Рис. 6. Результаты тестирования

Подробнее результаты тестирования можно просмотреть в текстовом файле ResTest.txt, который создается в той же папке, из которой был произведен запуск программы. В нем с помощью любого текстового редактора (например, Блокнота) можно просмотреть все вопросы и выбранные варианты ответов, а также увидеть, какие из них были правильными, а какие – ошибочными (рис. 7).

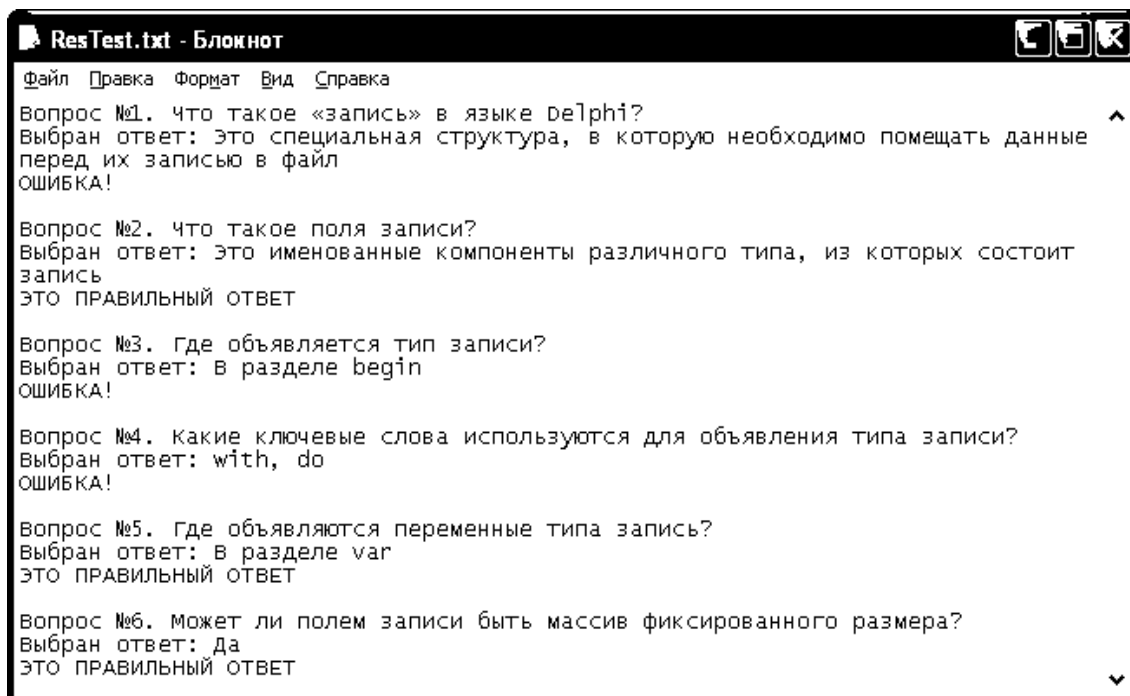


Рис. 7. Просмотр результатов тестирования в текстовом файле (фрагмент)

*Научная новизна.* Разработана и протестирована подсистема контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Подсистема удовлетворяет основным требованиям, предъявляемым к программному и информационному обеспечению для таких систем.

В результате проделанной работы был проведен анализ общих требований, выдвигаемых к подсистемам контроля знаний, разработано соответствующее программное и информационное обеспечение для подсистемы контроля знаний по теме «Программирование на языке Delphi. Работа с записями». Показаны примеры работы программы, демонстрирующие корректность ее функционирования.

Разработанная подсистема может быть использована в учебном процессе для оперативного контроля знаний студентов как автономно, так и в составе более полной системы.

**Список литературы:** 1. Бондаренко М.А. Інформаційні технології та програмування у середовищі Delphi. Х.: ФЛП Александрова К.М., 2014. 536 с. 2. Бондаренко М.А. Програмування у середовищі Delphi. Х.: «Бізнес Інформ», 2010. 704 с.

Поступила в редколлегию 12.03.2014

**Бондаренко Николай Андреевич**, канд. техн. наук, профессор, Украинская инженерно-педагогическая Академия. Научные интересы: информационные технологии и проектирование технических систем. Адрес: Украина, 61000, Харьков, ул. Клочковская, 195 г, кв. 44, тел. 7 19 50 01.