Содержание

Введение

РАЗДЕЛ 1. Задача программы

1.1 Постановка задачи

РАЗДЕЛ 2. Обзор существующих тестирующих программ и обоснование выбора темы

2.1 Тестирующая программа по английскому языку

РАЗДЕЛ 3. Разработка программы

3.1 Используемые методы

3.2 Алгоритм реализации роботы

Заключение

Список литературы

Приложение 1

1.1 Код программы для создания файла с вопросами и ответами

1.2 Код программы для чтения файла с вопросами и введения ответов

Приложение 2

2.1 Блок – схема программы

# ВВЕДЕНИЕ

Оглядываясь назад, в эпоху становления средств вычислительной техники, можно условно выделить ступени развития автоматизированных обучающих систем.

Первые, самые ранние, представляли собой лишь электронные версии печатных изданий и содержали только текстовую информацию. Достоинством таких систем в сравнении с бумажными аналогами, по существу, являлась лишь интеграция информации.

Постепенно, с развитием вычислительной техники, их стали сменять системы, которыми можно было пользоваться и как справочниками, то есть поисковые системы. Далее арсенал автоматизированных обучающих систем пополняется статической графикой, компонентами мультимедиа (звук и видео), компьютерной анимацией.

Таким образом, системы развивались в области информационных баз знаний, поддерживающих уже не только текстовое, но и аудио - видеовоздействие на обучаемого.

Колоссальный скачок был сделан и в сфере развития систем контроля, ведь без них, по существу, обучающая система мало чем отличается от справочного пособия. Итак, средства контроля обеспечивают ввод ответа обучаемого, обработку полученной информации и выдачу управляющего воздействия (например, вы вводите ответ, а система одобряет или отвергает его).

Следуя дальше по ступеням интеллектуализации обучающих систем, можно заметить, как плавно механизмы контроля перетекают в механизмы анализа высказываний обучающегося. Отметим, однако, что не следует путать их с простым контролем как простой оценки единичного события с интегральным оцениванием, являющимся результатом анализа и систематизации множества разнородных ответов данных в разное время. Не вдаваясь в теорию обучения, давайте рассмотрим простой пример. Ведь чем по сути отличается средство для создания обучающего курса от средства для создания мультимедийных энциклопедий? Примерно тем же, чем учебник отличается от лектора, ведущего лекции в учебном заведении.

Ведь учебник дает ответы на конкретные вопросы, помогает решать конкретные задачи и позволяет контролировать себя, заглядывая на последнюю страничку и сверяя правильный ответ с полученным. А как быть тем, кто не знает, какой вопрос задать или не может правильно его сформулировать? В классических учебных заведениях для этого существует лектор, который заботливо проводит студента по коридору знаний, понимает, с чего начать, и по ответам студента, на основе собственного опыта, делает выводы об уровне знаний студента в целом (то есть дает ему интегральную оценку).

Иными словами, анализ высказываний обучающегося позволяет не только контролировать учебный процесс, но и определяет стратегию обучения в зависимости от предыстории ответов и исходя из определенного набора правил.

В свою очередь, из анализа высказываний обучаемого проистекает управляющее воздействие на него, которое в частном случае может быть и удаленным. Последний метод, который называется дистанционной формой обучения, в настоящее время находится в стадии становления. Вследствие данного обстоятельства в последнее время технологии автоматизированного обучения все больше тяготеют к Internet -технологиям.

РАЗДЕЛ 1. Задача программы

1.1 Постановка Задачи

Необходимо создать тестирующую программу, которая будет проверять знания тестируемого человека. При этом должно быть не меньше двадцати вопросов. При завершение тестирования , программа должна выводить оценку тестируемого.

РАЗДЕЛ 2. Обзор существующих тестирующих программ

2.1 Тестирующая программа по английскому языку

Существует множество разнообразных программ, которые производят проверку знаний учащихся, по различны предметам обучения. Все обучающие программы могу быть написаны на разны языках программирования: Delphi, C++, С Sharp, Turbo Pascal и др.

Например: Тестирующая программа по английскому языку. Эта программа прописана на языке объектного программирования Delphi. В среде Delphi существуют уже созданные формы процедур, функций и др. Этот язык облегчает работу описания программ тем, что не нужно пописывать формы. Но, как и во всех других языках програмирования, Delphi содержит свои недостатки. Например: если программист укажет форму процедуры вручную, то компилятор её не увидит. Для этого нужно в настройках отмечать, что указанную форму нужно прочитать.

Turbo Pascal – это основа всех языков программирования, поэтому программа тестирования по математике написана на этом языке.

РАЗДЕЛ 3. Разработка программы

3.1 Используемые методы

В программе используется следующий алгоритм работы с данными;

Использование записей. Записи - это структура данных, состоящая из фиксированного числа компонентов, называемых полями записи. В отличие от массива, компоненты (поля) записи могут быть различного типа. Чтобы можно было ссылаться на тот или иной тип компонент записи, поля именуются.

Использование файловой системы. Файл – это создаваемый документ, при дальнейшем его использовании в программе. Все данные тестирования хранятся в файле name.dat и загружаются в программу при помощи инициализации файлов. Для этого нужно указать направление передачи данных для этого файла. В турбо Паскале можно открыть файл для чтения. Записи информации, а также для чтения и записи одновременно.

Использование модуля CRT. Модуль CRT содержит мощные подпрограммы, которые предоставляют возможность полного управления PC. Подпрограммы модуля CRT обеспечивают контроль над текстовыми режимами экрана, расширенными кодами клавиатуры, цветами, окнами и звуком.

Использование процедур. Процедура – это независимая именованная часть программы, которую можно вызвать из любой части программы по имени для выполнения определённой в ней последовательности действий.

3.2 Алгоритм реализации работы

Для реализации алгоритма создадим две программы:

1. Программа "Vvod" – это программа, которая создаёт файл с вопросами и ответами;
2. Программа "Initialization" – при запуске программы выводится сообщение о начале тестирования и производится инициализация файла, т.е считывание вопросов и даётся возможность ввести свои ответы на вопросы.

1. "Vvod". Создаём запись. Раздел записи состоит из нескольких идентификаторов полей, за идентификатором ставится двоеточие, и описание типа полей.

Type rec=record

vopros:string;

otvet:string;

end;

Здесь показано, что rec есть запись с полями vopros, otvet стригового типа.

* Vopros – будет содержать вопрос;
* Otvet – будет содержать ответ;

Создадим переменные:

* f – файл стрингового типа;
* x - целого типа, используется в качестве счётчика, который используется в цикле ввода вопросов;
* q - содержит запись типа rec, для ввода записи в файл;

Тело программы:

В начале программы, счётчик х обнуляем: х:=0. С помощью операторов Assign и Rewrite создается файл данная программа создоёт запись в файл name.dat. Создаём цикл с пост условием(repeat).

Счётчику х присвоим шаг +1: х:=х+1: т.е после каждого прохода цикла, х будет на одну единицу больше. Для текстовых файлов, открытых процедурой RESET, нельзя использовать процедуру WRITE или WRITELN. Нужно использовать стандартную процедуру REWRITE.

REWRITE (<ф.п.>)

инициирует запись информации в файл , связан с sq ранее с файловой переменной <f>. Процедурой REWRITE нельзя инициировать запись информации в ранее существовавший дисковый файл: при выполнении этой процедуры старый файл уничтожается и никаких сообщений об этом в программу не передается. Новый файл подготавливается к приему информации и его указатель принимает значение 0.

При запуске программы на экране будет выводиться сообщение "ввести вопрос" . После считывания переменной q. vopros происходит запись в файл и на экране выводится новое сообщение "введите ответ" считывания переменной q. otvet происходит запись в файл. На экране выводится новое сообщение "ввести вопрос".

Эти два сообщения будут выводиться на экран и записываться в файл name.dat до тех пор, пока вы не введёте двадцать вопросов и ответов, т.е пока х не будет равен n (Условие выхода из цикла: until x=n).

После окончания количества сеансов ввода, файл закрывается. Функция close(f) – закрывает файл, однако связь файлов с именем файла, установленная ранее процедурой ASSIGN, сохраняется. Вызов процедуры Close необходим при завершении работы с файлом. Если по какой-то причине процедура Close не будет выполнена, файл все - же будет создан на внешнем устройстве, но содержимое последнего буфера в него не будет перенесено. Для входных файлов использование оператора закрытия файла необязательно.

program Vvod;

type rec=record

vopros:string;

otvet:string;

end;

var f:file of string;

x:integer;

q:rec;

begin

x:=0;

assign(f,'name.dat');

rewrite(f);

repeat

x:=x+1;

writeln('введите вопрос');

readln(q.vopros);

write(f,q.vopros);

writeln('введите ответ');

readln(q.otvet);

write(f,q.otvet);

until x=n;

close(f);

end.

2 "Initialization"

В начале этой программы на экран выводится сообщение процедуры "Main Menu". Для начала тестирования нажмите "y". Для редактирования вопросов или выхода из программы нажмите "R". Чтобы создать эту процедуру, нужно в теле основной программы подключить модуль "crt", описать константы, в описании переменных указать ключ символьного типа:

uses crt;

const simbol:set of char=['y','Y','n','N',’r’,’R’,’t’,’T’];

….

var … key:char;

Procedure MainMenu(posmaster:word);

В описании констант мы указываем ['y','Y','n','N',’r’,’R’,’t’,’T’] – это символы вводимые с клавиатуры. Далее начинаем цикл с пост условием

Repeat

Ключу (key) присвоим значение считать ключ (readkey) , при этом условие выхода ключ указанный в списке констант (until key in symbol):

key:=readkey until key in simbol;

По нажатию клавиши 'y','Y' – происходит дальнейшее считывание. Срабатывает процедура "Test".

case key of

'y','Y':writeln;

…;

end;

По нажатию клавиши 'r','R' -

case key of

'r','r':begin

на экран выводится сообщение "Для завершения программы нажмите – n" "Для редактирования вопросов –T". При нажатии "n" происходит завершение программы и, всех процедур и функций.

case key of

'n','N':halt;

При нажатии "T" – происходит редактирование вопросов и ответов.

case key of

't','T':begin

…;

End;

Вы должны указать количество вопросов, после чего будет совершена перезапись файла. Для этого в теле основной программы нужно указать счетчик "x" – целого типа, он указывает количество вопросов; и счетчик "j" – для использования в качестве условия выхода из цикла

until j=x;

После пререзаписи файла старые данные файла не сохраняются. При повторном запуске программы используется уже вновь записанные данные.

Процедура "umnogenie"

Эта процедура разработана для вывода на экран сообщения – предупреждения, о том что знак умножения (shift+8) обязателен, иначе ответ не будет считатьс правильным.

Procedure umnogenie;

Begin

Writeln(‘учтите, что при вводе ответа знак умножения "\*" (shift+8) – обязателен’);

Writeln(‘ иначе ответ не будет считаться правильным. ‘);

Writeln (‘Для продолжения нажмите ENTER’);

End.

Процедура " Test"

Создадим переменные:

* Readstring – переменная, для считывания переменной стрингового типа из файла;
* Otv – переменная стрингового типа , для ввода ответа;
* Otvfile – переменная ответа из файла, для сравнения правильности ответов на вопросы;
* z, e, u – переменные целого типа (счетчики). "z" – Количество вопросов; "е" – Счётчик правильных ответов; "u" – Оценка;
* z1, z2, z3 – переменные действительного типа, для определения количества правильных ответов на оценку.

Переменные z1, z2, z3 описываем в теле основной программы

Var … z1, z2, z3: integer;

Эти переменные зависят от того сколько вопросов "x" вы ввели. Количество вопросов делим на заданные числовые переменные. Встроенная функция "round" округляет полученные ответы до ближайшего целого

z1:=round(x/4);

z2:=round(x/2);

z3:=round(x/1.3);

Переменные z, e, u также описываются в теле основной программы.

Var … z, e, u :integer;

Начало процедуры: обнуляем счетчики "z", "е" (для корректной работы программы).

e:=0; z:=0;

Далее с помощью процедуры "ASSIGN" связываем ранее объявленные файловые переменные с именем уже существующего файла.

assign(f,' name.dat');

C помощью стандартной процедуры RESET, инициализируем файл. Инициировать файл означает указать для этого файла направление передачи данных.

reset(f);

При выполнении этой процедуры дисковый файл подготавливается к чтению информации. В результате специальная переменная-указатель, связанная с этим файлом, будет указывать на начало файла, т.е. на компонент с порядковым номером 0.

Начинаем цикл с предусловием: пока не конец файла, начинаем подпрограмму.

While not eof(f) do begin

Логическая функция EOF(var f): Boolean возвращает значение TRUE, когда при чтении достигнут конец файла. Это означает, что уже прочитан последний элемент в файле или файл после открытия оказался пуст.

Счетчик "z" – количество вопросов, присвоим шаг +1 т.е после каждого прохода цикла, "z" будет на одну единицу больше.

z:=z+1;

C помощью оператора "clrscr" - очищаем экрана, на протяжении всего цила. Это для того, чтобы новый вопрос выводился на месте предыдущего.

Выводим сообщение о номере вопроса:

write('вопрос',z,':',' ');

С помощью порцедуры "read" – считываем связанную переменную из файла:

read(f,readstring);

read(f,namefile);

Далее на экран выводится сообщение считанной переменной и сообщение про возможность ввода ответа:

Writeln(readstring);

Write('Ответ ');Readln(otv); writeln;

Оператора "Readln(otv)" считывает вводимый ответ с клавиатуры.

Для контролирования количества правильных ответов и вывода на

экран оценки используем алгоритм с ветвлением.

"If" – оператор ветвления.

If x>=0 then y:=x else y=-x

Если введённый ответ будет соответствовать считываемому ответу из файла, то количество правильны ответов будет увеличиваться, тем самым увеличивается оценка проверки знаний тестируемого.

if otv = otvfile then begin e:=e+1; u:=u+1;

Для того чтобы оценка соответствовала не количеству вопросов, а оценкам пяти бальной системы, нужно указать условие: если количество ответов меньше либо равна z1, ваша оценка – 2 иначе, если ответов больше z1, но меньше либо равно z2, ваша оценка – 3 иначе , если количество ответов больше z2, но меньше либо равна z3, ваша оценка – 4 иначе, ваша оценка – 5;

if u<=z1 then p:=2 else

If (u>z1) and (u<=z2) then p:=3 else

If (u>z2) and (u<=z3) then p:=4 else p:=5;

Стандартной процедурой close(f) – закрываем файл.

Для вывода на экран показания счетчиков "e" "u", очистим экран "clrscr". Конец процедуры.

Clrscr;

writeln('Правильны ответов ', e);

writeln('Ваша оценка ', p);

end;

Начинаем тело основной программы. После каждого выполнения цикла, очищаем экран. Всё тело программы состоит из процедур. Для выполнения процедур необходим их вызов. Вызов осуществляется по имени данной процедуры в теле программы. Имя процедуры воспринимается как оператор. При его выполнении, выполняется вся подпрограмма. Посе процедуры "umnogenie" прописываем опрерато "readln", для вывода на экран предупреждающего сообщения и задержки работы процедуры "Test". В конце программы ставится "end".

Заключение

Работа сделана на языке программирования Turbo Pascal. Turbo Pascal является реализацией Pascal'я. Самая первая версия Pascal была разработана на кафедре информатики Стэндфордского университета швейцарским ученым Николаусом Виртом в 1968 году.

С момента появления Pascal на рынке продуктов прошло много времени, прежде чем он получил всеобщее признание. В середине 80-х годов американской фирмой Borland International Inc была создана реализация языка Pascal, известная и по сей день под именем Turbo Pascal. Эта фирма объединила очень быстрый компилятор с редактором текста и добавила к стандартному Паскалю мощное расширение, что способствовало успеху первой версии этого языка.

Язык программирования Turbo Pascal, является мощной платформой (родителем) для других языков программирования. Из Turbo Pascal’ ля развилось множество других языков, поэтому он есть составная часть всех языков программирования.

В ходе разработки тестирующей программы проводилось обработка данных записи в файл, создана программа для считывания созданного файла, разработана подпрограмма, обрабатываемая ответы и оценку тестируемого.

При выполнении данной курсовой работы были приобретены навыки работы с файлами и записями, изучён синтаксис данного языка программирования.

В ходе выполнения проекта данной курсовой работы:

* были изучены методы применения записей в файл;
* разработан программный код тестирующей программы.

Список литературы

1. Немнюгин С.А. Turbo Pascal. Программирование на языке высокого

уровня: Учебник для ВУЗов

2. С.М.Окулов - 2 главы из школьного учебника по информатике;

3. C. А. Немнюгин - учебник TURBO PASCAL;

4. Delannoy C. Exercises en Turbo-Pascal, Eyrolles, Paris, 1990;

5. Епанешников А. Программирование в среде Turbo-Pascal 7.0 /

А. Епанешников, В. Епанешников. – М.: Диалог-МИФИ, 1993;

6. Фаронов В. В. Turbo Pascal 7.0. – М.: Надежда, 2000;

7.Лукин С.Н. Turbo Pascal 7.0. Самовчитель для початківців. – М.: "Діалог-МІФІ", 2002

8. . Пратт Т. Языки программирования: разработка и реализация;

9. Брукшир Дж. Гленн. Введение в компьютерные науки. – М.:

Вильямс, 2001;

10. Miller L. N. Advanced programming: Design and Structure using

Pascal, Addison-Wesley publishing Company, 1986гина, М. Г. Громада. – СПб.: Наука и техника, 2001;

11 Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989.

Т. Пратт, М. Зелковиц. – СПб.: Питер, 2002;

12. Архангельский А. Я. Программирование в Delphi 7. – М.: Би-

ном, 2005;

13. Самуйлов С. В. Разработка программ в среде Турбо-Паскаля 7.0.

Лабораторный практикум. – Пенза, 1996;

14. Есипов А. С. Информатика. Задачник;

15 Марченко А. И. Программирование в среде Turbo Pascal 7.0 / Т. 1. – М.: Мир, 1982;

16.А.Ф. Верлань, Н.В. Апатова "Информатика 10-11кл.", Киев 2001

17. С. А. Немнюгин "Учебное пособие", Санкт –Петербург 2004

18Фаронов В. В., серия: "В подлиннике", 2004 г., Изд.: БХВ-Петербург

19. Коффман Э., серия: "Руководство разработчика.", 2002 г., Изд.: Диалектика: Киев;

20. Кассера Винфрид и Фолькер "Turbo Pascal 7.0" 2003 г., Изд.: ДиаСофт;

Приложение 1

1.1 Код файла для создания файла с вопросами и ответами

program Vvod;

type rec=record

vopros:string;

otvet:string;

end;

var f:file of string;

c:string;x:integer;q:rec;

begin

x:=0;

assign(f,'control.dat');

rewrite(f);

repeat

x:=x+1;

writeln('Введите вопрос');

readln(q.vopros);

write(f,q.vopros);

writeln('введите ответ');

readln(q.otvet); write(f,q.otvet); until x=20; close(f); end.

1.2 Код файла для чтения файла с вопросами и введения ответов

program initialization;

uses crt;

const simbol:set of char=['y','Y','n','N','r','R','t'];

type rec=record

vopros:string;

otvet:string;

end;

var f:file of string;z,e,u:integer;key:char;

z1,z2,z3:real;

x:integer;q:rec; j:integer;

Procedure MainMenu;

begin

writeln('Для начала тестирования нажмите - y');

writeln ('Для редактирования вопросов или выхода из программы нажмите - R');

repeat

key:=readkey until key in simbol;

case key of

'y','Y':writeln;

'r','R':begin clrscr;

writeln('Для завершения программы нажмите - n');

writeln('Для редактирования вопросов -T');

repeat

key:=readkey until key in simbol;

case key of

'n','N':halt;

't','T':begin

j:=0;

assign(f,'control.dat');

rewrite(f); clrscr;

writeln('Введите количество вопросов');

readln(x);

repeat

j:=j+1;

writeln('Введите вопрос');

readln(q.vopros);

write(f,q.vopros);

writeln('Введите ответ');

readln(q.otvet);

write(f,q.otvet);

until j=x;

close(f); begin halt; end;

end;

end;end; end; end;

procedure umnogenie;

begin

writeln('Учтите что при вводе ответов знак умножения " \* "(shift+8) обязателен ');

writeln('иначе ответ не будет считаться правильным');

writeln('Для продолжения нажмите - ENTER') ;

end;

Procedure Test;

var

readstring,otv,otvfile:string;

begin

e:=0; z:=0;

assign(f,'control.dat');

reset(f);

While not eof(f) do begin

z:=z+1; clrscr;

write('Вопрос',z,' : ',' ');

writeln;writeln;inc(x);

read(f,readstring);

read(f,otvfile);

writeln (readstring);

Write('Ответ');Readln(otv);

if otv = otvfile then begin e:=e+1;end;end;

z1:=round(x/4);

z2:=round(x/2);

z3:=round(x/1.3);

if e<=z1 then u:=2 else

if (e>z1) and(e<=z2) then u:=3 else

if (e>z2) and (e<=z3) then u:=4 else u:=5;

close(f);

clrscr;

writeln('Правильных ответов’,e);

writeln('Ваша оценка',u);

end;

begin clrscr;

MainMenu;

umnogenie; readln;

Test;

end.

Приложение 2

2.1 Блок - схема

