

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра системного анализа и
автоматического управления

**РАЗРАБОТКА МОДУЛЯ ГРАФИЧЕСКОГО ОТОБРАЖЕНИЯ
РЕЗУЛЬТАТОВ РАБОТЫ ПРИЛОЖЕНИЯ PSYTESTSGU**
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Чуракова Игоря Денисовича

Научный руководитель
к. ф.-м. н., доцент

И. Е. Тананко

Заведующий кафедрой
к. ф.-м. н., доцент

И. Е. Тананко

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы. Использование тестов - неотъемлемая черта современной психодиагностики. Можно выделить несколько областей практического использования результатов психодиагностики: сфера обучения и воспитания, сфера профессионального отбора и профессиональной ориентации, консультативная и психотерапевтическая практика и, наконец, область экспертизы - медицинской, судебной и т. п.

В то же время, повсеместная автоматизация подаёт идеи для упрощения проведения психологического тестирования, что даст возможность проведения тестирования в больших масштабах, также уменьшая влияние человеческого фактора на получение результата.

Целью бакалаврской работы является командное создание самостоятельного приложения *PsyTestSGU* для проведения психологического тестирования, которое позволяет создавать тесты, проводить тестирование на основе любых созданных внутри приложения тестов, собирать статистику пройденных тестов а также экспортировать статистику для анализа в любых приложениях для анализа статистики.

В ходе написания бакалаврской работы должны быть изучены термины психологического тестирования, язык программирования *Pascal* и специальная среда программной разработки *Lazarus*, а также получены навыки работы с системами контроля версий, в частности *Git* и *GitHub* и навыки работы в команде.

В ходе проделанной работы должны быть разработаны такие модули приложения, как:

- модуль графического отображения результатов работы приложения;
- модуль экспорта данных приложения (результатов всех пройденных тестов).

Разработанное приложение предназначено для проведения психологического тестирования и профилирования пользователей.

Цель бакалаврской работы — получить опыт и навыки работы в команде, а также навыки работы с системами котроля версий, посредством командной разработки самостоятельного приложения для проведения психологического тестирования *PsyTestSGU*.

Поставленная цель определила **следующие задачи**:

1. Изучить основные понятия и классификации психодиагностических тестов;
2. Изучить язык *Pascal* и среду разработки *Lazarus* для создания приложения;
3. Освоить работу с системами контроля версий *Git* и платформой *GitHub*, настроить рабочий процесс команды;
4. Спроектировать структуру хранения тестов, результатов и статистики, а также модуль экспорта данных;
5. Разработать графический модуль приложения;

Методологические основы разработки на языке *Pascal*, работы в среде разработки *Lazarus* и использования системы контроля версий *Git* представлены в работах В. Абрамова, К. Мансурова, Л. Фишермана.

Практическая значимость бакалаврской работы. В ходе выполнения выпускной квалификационной работы были получены навыки работы в команде и навыки работы с системой контроля версий *Git*, а также разработаны графический модуль и модуль экспорта результатов работы для приложения *PsyTestSGU*, предназначенного для проведения психологического тестирования и профилирования пользователей.

Структура и объем работы. Бакалаврская работа состоит из введения, 4 разделов, заключения, списка использованных источников и цифрового носителя в качестве приложения. Общий объем работы — 50 страниц, из них 47 страниц — основное содержание, включая 29 рисунков и 1 таблицу, список использованных источников информации — 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Описание программы» посвящен описанию основных понятий психологического тестирования, использованных инструментов и процесса выбора подходящего формата для экспорта данных.

В подразделе 1.1 приведено описание сферы психологического тестирования, её понятий и истории.

Подраздел 1.2 посвящен обзору языка программирования *Pascal*.

В подразделе 1.3 рассматривается среда разработки *Lazarus*.

Подраздел 1.4 рассматривает принципы работы систем контроля версий и систему контроля версий *Git*.

В подразделе 1.5 рассматриваются форматы экспорта данных приложения, условия для их отбора и доступные варианты, а также производится окончательный выбор формата.

Подраздел 1.6 описывает выбранный формат экспорта данных приложения — *CSV(Comma-Separated Values)*

В подразделе 1.7 описывается приложение *PsyTestSGU*

Второй раздел «Прделанная работа по разработке программы *PsyTestSGU*» посвящен описанию процесса командной разработки программы *PsyTestSGU*, её функционала и модулей.

В подразделе 2.1 описывается процесс разработки программы.

Подраздел 2.2 посвящён описанию функционала программы.

В подразделе 2.3 описываются все модули программы.

Третий раздел «Прделанная работа по разработке программы *PsyTestSGU*» посвящен описанию проделанной работы по разработке программы *PsyTestSGU*. Были описаны этапы разработки модуля экспорта данных и графического модуля приложения.

В подразделе 3.1 описывается разработка модуля экспорта данных приложения.

Модуль экспорта данных находится в административной части, так как затрагивает информацию о всех пользователях, которая не должна быть в открытом доступе.

По причинам, описанным в подразделе 1.5, в качестве формата для экспорта данных был выбран формат *CSV*.

Экспорт данных производится по нажатию на специальную кнопку ”Экспорт данных...”, и предлагает пользователю указать директорию для сохранения экспортированного файла, после чего производит экспорт в файл.

С помощью графического интерфейса среды разработки добавим кнопку на панель просмотра всех результатов, зададим ей позицию, размер, имя и триггер-функцию. В результате в интерфейсе появляется кнопка ”Экспорт данных...”.

Для языка *Pascal* логика алгоритма состоит из следующих шагов:

- инициализация контейнера для записи данных;
- начало цикла *1i* — проход по строкам таблицы;
- инициализация временного контейнера для записи текущей строки;

- начало цикла $2j$ — проход по столбцам таблицы;
- замена в клетке по адресу $[j,i]$ двойных кавычек на сдвоенные двойные кавычки;
- конкатенация временного контейнера с текущей клеткой и символом-разделителем “,”;
- конец цикла $2j$;
- добавление содержимого временного контейнера (текущей прочитанной строки) в контейнер для записи данных;
- конец цикла $1i$;
- установка имени файла по умолчанию;
- вывод пользователю диалога о сохранении файла и предложение выбора директории для сохранения;
- сохранение данных из контейнера в файл;

В созданной триггер-функции *OnClick* пропишем алгоритм экспорта данных:

```

1  procedure TFormResult.ExportButtonClick(Sender: TObject);
2  var
3      exportData: TStringList;
4      tempString: string;
5      i,j:         integer;
6  begin
7      exportData := TStringList.Create;
8      for i:= 0 to StringGrid1.RowCount-1 do begin
9          tempString:='';
10         for j:= 0 to StringGrid1.ColCount-1 do begin
11             tempString:= tempString + '"'
12                 + StringGrid1.Cells[j,i].Replace('','"') + ',';
13         end;
14         exportData.Add(tempString);
15     end;
16
17     SaveDialog1.FileName:='PsyTestExport.csv';
18     if SaveDialog1.Execute then
19     begin
20         exportData.WriteBOM:=true;
21         exportData.SaveToFile(SaveDialog1.FileName, TEncoding.UTF8);
22         AdminWindow.ListBox1.Clear;
23         AdminWindow.FormCreate(nil);

```

```
24     end;
25 end;
```

Подраздел 3.2 посвящен разработке модуля графического отображения результатов работы приложения.

Модуль графического отображения результатов работы приложения находится в административной части и выполняет функцию просмотра результатов пройденных тестов по выбранному пользователю.

Модуль представляет собой окно с выводом всех результатов пройденных тестов по всем пользователям, а также кнопки для удаления результата, кнопку экспорта всех результатов, кнопку перехода к просмотру индивидуальной статистики и график статистики по всем результатам тестов. На графике столбцы обозначают тип теста, а их высота обозначает средний балл результатов тестов, указанный в процентах.

С помощью графического интерфейса среды разработки создадим панель просмотра всех результатов, зададим всем составляющим панели позицию, размер, имя и триггер-функцию.

Процедура загрузки данных из файла *JSON* производится посредством использования метода *GetJSON* и дальнейшего перехода по дереву свойств внутри файла:

```
1 procedure ParseJson(pathToFile: string);
2 var
3     JRoot:TJSONData;
4 begin
5     JRoot:=GetJSON(ReadFileToString(pathToFile));
6     try
7         FormResult.StringGrid1.RowCount:=FormResult.StringGrid1.RowCount+1;
8         FormResult.StringGrid1.Cells[0,FormResult.StringGrid1.RowCount-1]:=
9             JRoot.FindPath('group').AsString;
10        FormResult.StringGrid1.Cells[1,FormResult.StringGrid1.RowCount-1]:=
11            JRoot.FindPath('testname').AsString;
12        FormResult.StringGrid1.Cells[2,FormResult.StringGrid1.RowCount-1]:=
13            JRoot.FindPath('surname').AsString + ' ' +
14            JRoot.FindPath('firstname').AsString + ' ' +
15            JRoot.FindPath('lastname').AsString;
16        FormResult.StringGrid1.Cells[3,FormResult.StringGrid1.RowCount-1]:=
17            JRoot.FindPath('result').AsString;
18    finally
```

```

19     FreeAndNil(JRoot);
20 end;
21 end;

```

Далее для заполнения данных в таблицу по всем имеющимся результатам, пройдемся по всем *JSON* файлам в директории:

```

1  procedure CheckResultsUsers;
2  var
3      Sr : TSearchRec;
4      Attr : Integer;
5  Begin
6      Attr := faAnyFile - faDirectory;    //Все файлы, исключая папки.
7      try
8          if FindFirst(gPath + '*', Attr, Sr) = 0 then
9              repeat
10                 ParseJson(gPath + Sr.Name)
11                 until FindNext(Sr) <> 0;
12             finally
13                 FindClose(Sr);
14             end;
15 end;

```

В созданной триггер-функции *OnClick* пропишем алгоритм предзагрузки пользователей в выпадающее меню окна индивидуальной статистики:

```

1  procedure TFormResult.StatsTestButtonClick(Sender: TObject);
2  var
3      JResult:TJSONData;
4      studentList: TStringList;
5      resultFiles: TStringList;
6      i,j:integer;
7  begin
8      try
9          stats.StatsTestWindow.Show;
10         FormResult.Visible:=false;
11         studentList := TStringList.create;
12         studentList.Sorted:= true;
13         studentList.Duplicates:= dupIgnore;
14         resultFiles := FindAllFiles('PsyTest/results/', '*.json', true);
15         for i:=0 to (resultFiles.Count-1)do
16             begin

```

```

17         JResult:=GetJSON(ReadFileToString(resultFiles[i]));
18         studentList.add(JResult.FindPath('group').AsString+' ',
19                         +JResult.FindPath('surname').AsString+' ',
20                         +JResult.FindPath('firstname').AsString+' ',
21                         +JResult.FindPath('lastname').AsString);
22     end;
23     for j:=0 to (studentList.Count-1)do
24     begin
25         stats.StatsTestWindow.UserSelector.Items.Add(studentList[j]);
26     end;
27 finally
28     resultFiles.Free;
29     FreeAndNil(JResult);
30 end;
31 end;

```

Результирующее окно представлено на рисунке 1.

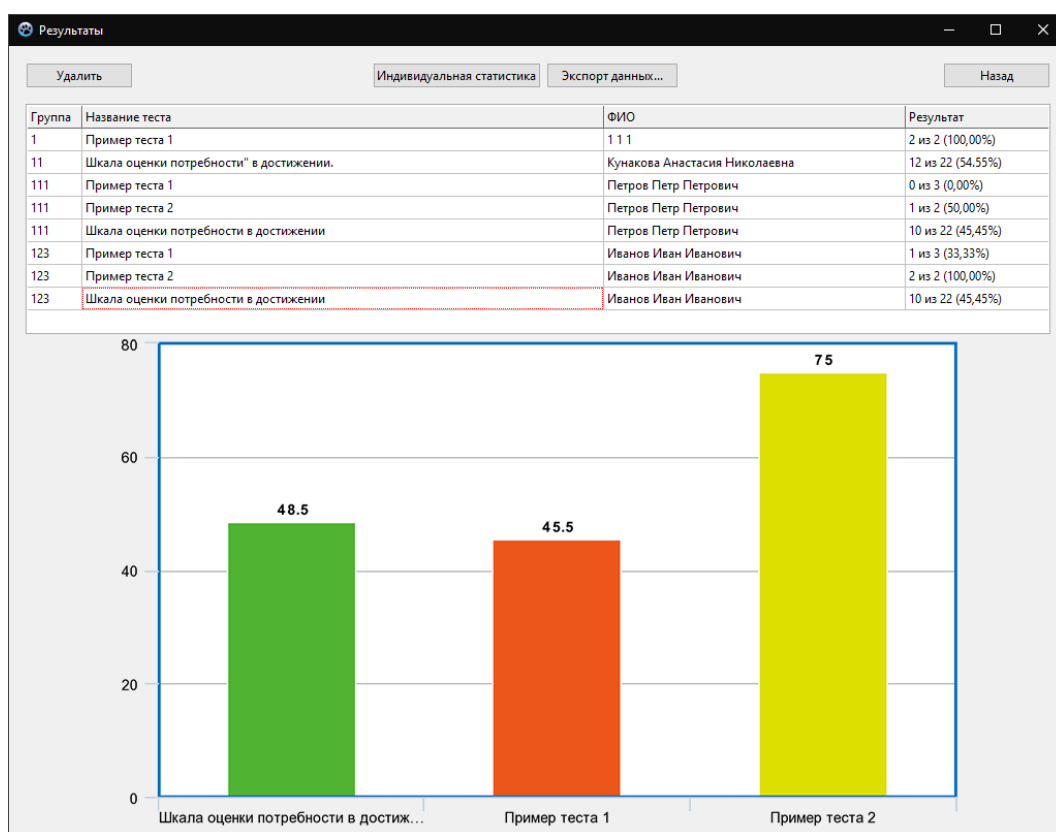


Рисунок 1 – Окно просмотра всех результатов тестирования

С помощью графического интерфейса среды разработки создадим новую форму для окна, добавим выпадающее меню выбора пользователя, список пройденных тестов и область вывода результатов по пройденному тесту, а также сконфигурируем вывод круговой диаграммы с информацией об

ответах пользователя. В результате в графическом интерфейсе программы получаем окно модуля графического отображения результатов работы приложения, как представлено на рисунках 2 и 3.

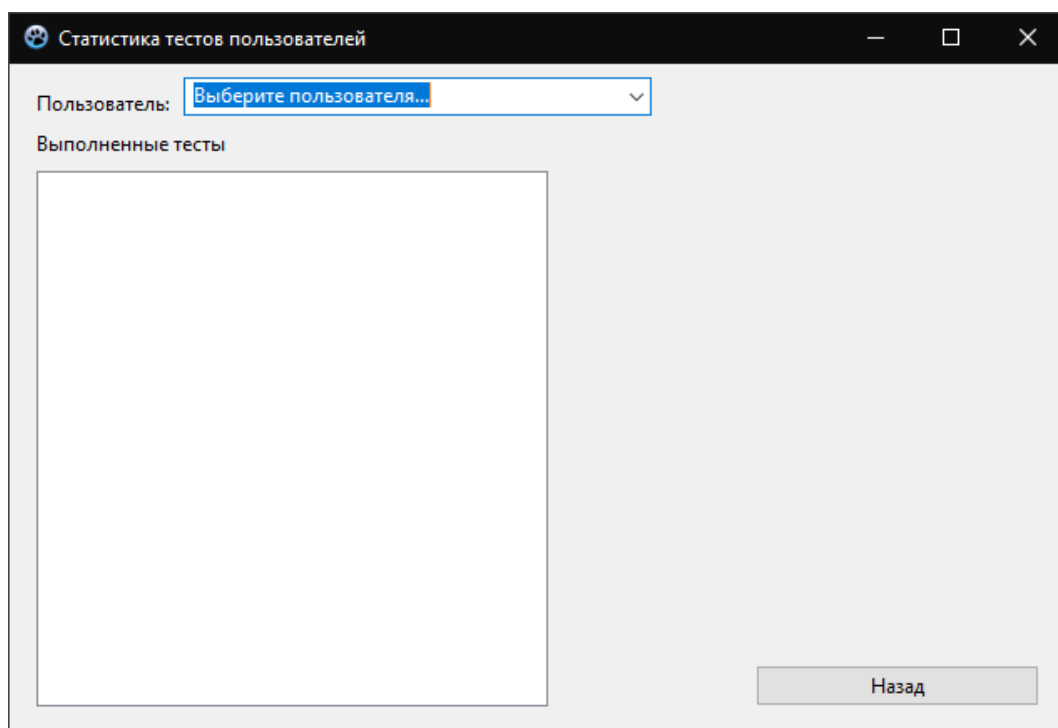


Рисунок 2 – Форма просмотра результатов тестов пользователя, изначальное состояние

Четвертый раздел «Демонстрация работы приложения

PsyTestSGU» посвящен презентации результатов разработки программы и обзор её функционала.

Подраздел 4.1 описывает процесс работы с программой от лица Администратора.

В подразделе 4.2 демонстрируется окно просмотра всех результатов тестирования.

Подраздел 4.3 описывает процесс экспорта данных приложения.

В подразделе 4.4 демонстрируется процесс просмотра результатов по конкретному пользователю программы.

Подраздел 4.5 описывает процесс добавления нового теста.

В подразделе 4.6 демонстрируется процесс удаления существующего теста из приложения.

Подраздел 4.7 описывает вход в систему в качестве пользователя.

В подразделе 4.8 демонстрируется главное окно пользователя.

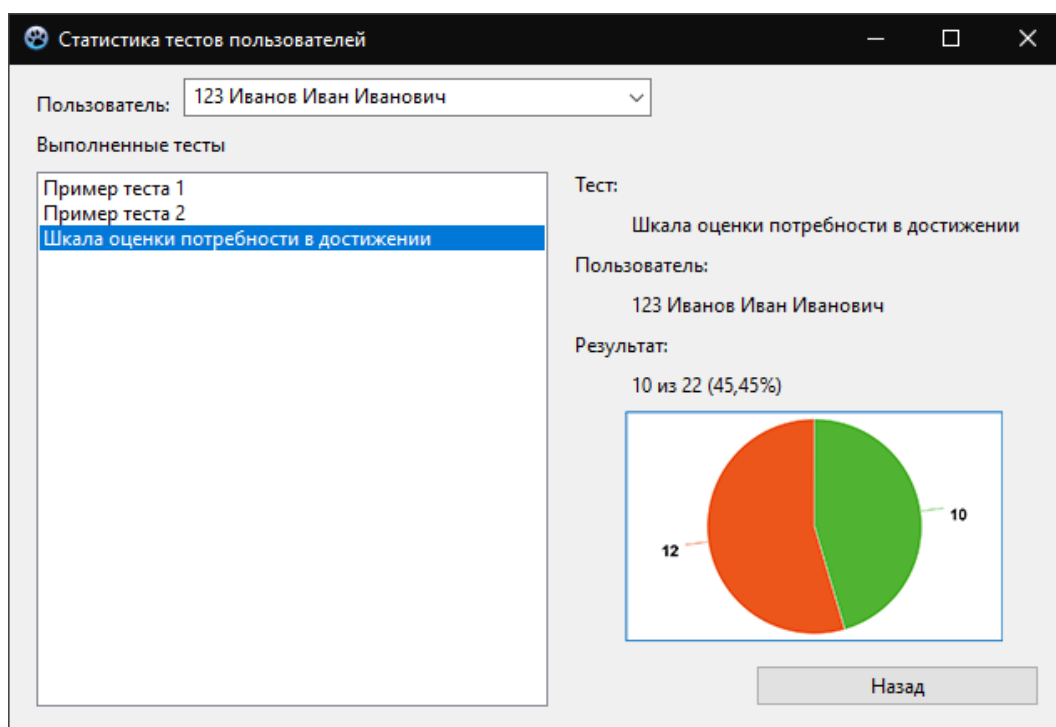


Рисунок 3 – Форма просмотра результатов тестов пользователя, выбран пользователь и тест

Подраздел 4.9 описывает процесс прохождения тестирования пользователем программы.

ЗАКЛЮЧЕНИЕ

Во время написания работы было создано самостоятельное приложение *PsyTestSGU* для проведения психологического тестирования, которое позволяет создавать тесты, проводить тестирование на основе любых созданных внутри приложения тестов, собирать статистику пройденных тестов а также экспортировать статистику для анализа в любых приложениях для анализа статистики. Приложение охватывает все возможные способы использования в рамках поставленной задачи.

В ходе написания бакалаврской работы были изучены термины психологического тестирования, язык программирования *Pascal* и специальная среда программной разработки *Lazarus*, а также были получены навыки работы с системами контроля версий, в частности *Git* и *GitHub*.

Результатом бакалаврской работы является приложение *PsyTestSGU*, позволяющее проводить психологическое тестирование пользователей, а также экспортировать данные для дальнейшего анализа в другом программном обеспечении или вручную.

В ходе проделанной работы были разработаны такие модули приложения, как:

- модуль графического отображения результатов работы приложения;
- модуль экспорта данных приложения (результатов всех пройденных тестов).

Разработанное приложение рекомендуется применять преподавателям университета для проведения психологического тестирования и профилирования своих студентов.

Основные источники информации:

1. Абрамов, В. Введение в язык Pascal. (СПО). Учебное пособие. / В. Абрамов. — М.: КноРус, 2022. — 562 с
2. Мансуров, К. Основы программирования в среде Lazarus / К. Мансуров. — СПб.: Нобель Пресс, 2013. — 772 с.
3. Фишерман, Л. Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения / Л. Фишерман. — СПб.: Питер, 2021. — 304 с.
4. Эгбал, Н. Open Source. Разработка программ с открытым исходным кодом Открытый исходный код на примере GitHub / Н. Эгбал. — СПб.: Питер, 2021. — 390 с
5. Анастаси, А. Психологическое тестирование / А. Анастаси. — СПб.: Педагогика, 1982. — 588 с.
6. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. — СПб.: Питер, 2022. — 352 с
7. Сассман, Д. Структура и Интерпретация Компьютерных Программ / Д. Сассман. — М.: Добросвет, 2022. — 608 с.
8. Лишнер, Р. Delphi. Справочник / Р. Лишнер. — М.: Символ-Плюс, 2011. — 127 с.
9. Джоэл, Г. Data Science. Наука о данных с нуля / Г. Джоэл. — СПб.: БХВ-Петербург, 2020. — 416 с.
10. Келлехер, Д. Наука о данных. Базовый курс / Д. Келлехер. — М.: Альпина Паблишер, 2021. — 224 с