МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Рязанский государственный радиотехнический университет имени В.Ф.Уткина»

Кафедра «Космические технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ**

на тему

«Разработка программы, позволяющей провести тестирование студента   
по основам компьютерных наук»

Студент: Ермилова Мария Сергеевна

Руководитель проекта: Доцент кафедры КТ

Наумов Д.А.

Рязань, 2019г.

ФГБОУ «Рязанский государственный радиотехнический университет»

Кафедра «Космические технологии»

Задание на курсовой проект

по курсу «Основы компьютерных наук»

**Студенту** Ермиловой Марии Сергеевны **Группа**: 748

**Срок предоставления проекта**

**1.** **Тема**: «Разработка программы, позволяющей провести тестирование студента по основам компьютерных наук»

**2. Исходные данные**:

– разработать тестовые вопросы для дисциплины «Основы компьютерных наук» по теме  архитектура ЭВМ;

– разработать модуль, осуществляющий редактирование тестовых вопросов.

**3. Содержание пояснительной записки (основные разделы)**:

 Введение

1 Постановка и анализ задачи

2 Проектирование структуры данных

3 Проектирование модульной структуры программы

4 Разработка алгоритмов

5 Тестирование программы

6 Разработка документации

 Заключение

 Библиографический список

**4. Рекомендации по составлению разделов и оформлению пояснительной записки**

Записка должна быть написана грамотным языком и оформлена с учетом всех норм оформления подобных документов. Содержание основных разделов рекомендуется разбивать на параграфы. Во «Введении» должны быть рассмотрены цели и задачи курсового проекта, приведен обзор содержания основных глав пояснительной записки, перечислены средства разработки с указанием причин их выбора. В раздел «Заключение» нужно подвести основные итоги по проделанной работе, а также дать оценку этой работе.

**5. Перечень материалов проекта**

К пояснительной записке необходимо проложить компакт-диск, содержащий:

– текст пояснительной записки;

– исходные файлы проекта;

– результаты компиляции и сборки системы.

Руководитель проекта

(дата, подпись)

Задание принял к исполнению «     »            2019 (подпись)

**Содержание**

Введение 4

1. Постановка и анализ задачи 7

2. Проектирование структуры данных 9

3. Проектирование модульной структуры программы 15

4. Проектирование алгоритмов 22

4.1 Проектирование алгоритмов 22

4.2 Проектирование алгоритмов 26

5. Тестирование программы 32

6. Разработка документации 37

Заключение

Библиографический список

Приложение А

Приложение Б

**Введение**

Одной из важнейших областей применения компьютеров является переработка и хранение больших объемов информации в различных сферах деятельности человека: в экономике, банковском деле, торговле, транспорте, медицине, науке и т.д.

Существующие современные информационные системы характеризуются огромными объемами хранимых и обрабатываемых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей.

Информационная система — это система, которая реализует автоматизированный сбор, обработку и манипулирование данными и включает технические средства обработки данных, программное обеспечение и обслуживающий персонал.

Цель любой информационной системы — обработка данных об объектах реального мира. Основой информационной системы является база данных. В широком смысле слова база данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления его объектами и, в конечном счете, автоматизации, например, предприятие, вуз и т. д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. При этом очень важно выбрать правильную модель данных. Модель данных — это формализованное представление основных категорий восприятия реального мира, представленных его объектами, связями, свойствами, а также их взаимодействиями.

*База данных* — это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

Информация в базах данных хранится в упорядоченном виде. Так, в записной книжке все записи упорядочены по алфавиту, а в библиотечном каталоге либо по алфавиту (алфавитный каталог), либо в соответствии с областью знания (предметный каталог).

Система программ, позволяющая создавать БД, обновлять хранимую в ней информацию, обеспечивающая удобный доступ к ней с целью просмотра и поиска, называется *системой управления базами данных (СУБД)* [1] .

Разработку программы можно разбить на следующие этапы:

1. Составление алгоритма решения задачи. Алгоритм — это описание последовательности действий, которые необходимо выполнить для решения поставленной задачи.

2. Написание текста программы. Текст программы пишут на каком-либо языке программирования и вводят его в компьютер с помощью текстового редактора.

3. Отладка программы. Отладка программы — это процесс устранения ошибок из текста программы. Все ошибки делятся на синтаксические и логические. При наличии синтаксических ошибок программа не запускается. Подобные ошибки исправляются проще всего. Логические ошибки — это ошибки, при которых программа работает, но неправильно, выдавая не те результаты, которые ожидает разработчик или пользователь. Логические ошибки исправить сложнее, чем синтаксические, иногда для этого приходится переписывать отдельные участки программы, а иногда перерабатывать весь алгоритм.

4. Тестирование программы. Тестирование программы — процесс выявления ошибок в ее работе. Процессы отладки и тестирования сопровождаются неоднократным запуском программы на выполнение.

При разработке программ на языке PascalABC.NET можно использовать синтаксис традиционных реализаций Паскаля (таких как Delphi Pascal или Free Pascal). Однако язык PascalABC.NET включает ряд новых возможностей, отсутствующих в традиционном Паскале. Эти возможности позволяют записывать программы более компактным и наглядным способом. В данной главе описываются наиболее существенные нововведения PascalABC.NET, без использования которых не обходится практически ни одна программа и которые будут активно применяться на протяжении всей книги.

Описание переменных PascalABC.NET позволяет описывать переменные не только в специальном разделе описаний, но и в любом месте раздела операторов. Область действия такой переменной продолжается до окончания того блока, в котором она описана. Еще одной важной особенностью описания переменных в PascalABC.NET является вывод типов (type inference): если в момент описания переменная инициализируется некоторым значением, то тип переменной можно не указывать: он будет автоматически выведен компилятором из типа инициализирующего выражения [2].

Стандартная библиотека платформы .NET включает ряд обобщенных классов, которые в ряде случаев оказываются более удобным средством обработки данных, чем массивы и последовательности. Все эти классы можно использовать в PascalABC.NET без каких-либо специальных действий по подключению дополнительных библиотек. В главах 2–5 дается обзор возможностей основных обобщенных классов для хранения наборов данных [3].

* + - 1. **1. Постановка и анализ задачи**

Целью курсового проекта является автоматизация процесса проверки знаний студентов по дисциплине «Основы компьютерных наук».

Задачей курсового проектирования является разработка программы модульной структуры (далее – программа), позволяющей провести тестирование студентов, рассчитать полученный студентом балл и сохранить результаты прохождения тестов по одной из тем дисциплины «Основы компьютерных наук».

Программа должна реализовывать следующие функции:

* ввод тестовых вопросов;
* ввод вариантов ответа на тестовый вопрос;
* указание (ввода), какой вариант правильный;
* указание балла за вопрос;
* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.

В рамках курсового проекта будут реализованы 2 модуля:

* модуль «Создание»;
* модуль «Редактирование».

Модуль «Создание» должен осуществлять в диалоговом режиме взаимодействие с пользователем и решать следующие задачи:

* ввод тестовых вопросов;
* ввод вариантов ответа на тестовый вопрос;
* указание (ввода), какой вариант правильный;
* указание балла за вопрос.

Модуль «Редактирование» должен осуществлять в диалоговом режиме взаимодействие с пользователем и решать следующие задачи:

* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.

На основе анализа задач, решаемых модулем «Тестирование», можно сделать вывод о том, что реализация обоих модулей будет зависеть от реализации следующих функций:

* ввод тестовых вопросов;
* ввод вариантов ответа на тестовый вопрос;
* указание (ввода), какой вариант правильный;
* указание балла за вопрос;
* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.
  + - 1. **2. Проектирование структуры данных**

Рассмотрим каждую из функций системы с точки зрения (Таблица 1). В столбце «Данные» перечислены категории данных, которые использует соответствующая функция.

Таблица 1 – Функции системы

|  |  |  |
| --- | --- | --- |
| № | Функция | Данные |
| Ф1 | Ввод тестовых вопросов | Тест |
| Ф2 | Ввод вариантов ответа на тест.вопрос | Тестовые вопросы |
| Ф3 | Указание(ввода),какой вариант правильный | Тест,  тестовые вопросы,  варианты ответа |
| Ф4 | Сохранение всего содержимого в файл | Тест,  тестовые вопросы,  варианты ответа;  готовый файл |
| Ф5 | Изменение отдельных текстовых вопросов | Тест,  тестовые вопросы,  готовый файл. |
| Ф6 | Изменение отдельных вариантов ответа | Тест,  тестовые вопросы,  варианты ответа,  готовый файл. |
| Ф7 | Изменение признака правильности ответа | Тест,  тестовые вопросы,  критерии оценивания теста,  готовый файл. |

Окончание Таблица 1 – Функции системы

|  |  |  |
| --- | --- | --- |
| № | Функция | Данные |
| Ф8 | Сохранение изменений в файл | Тест,  тестовые вопросы,  критерии оценивания теста,  варианты ответа,  готовый файл,  преобразованный файл. |
| Ф9 | Указание балла за вопрос | Тест,  тестовые вопросы. |
| Ф10 | Добавление нового вопроса | Тест,  тестовые вопросы,  готовый файл. |
| Ф11 | Добавление новых вариантов ответа | Тест,  тестовые вопросы,  варианты ответа,  готовый файл. |
| Ф12 | Удаление вопроса | Тест,  тестовые вопросы,  готовый файл. |
| Ф13 | Удаление ответа | Тест,  тестовые вопросы,  варианты ответа,  готовый файл. |

Таким образом, выделены следующие категории данных, с которыми будет работать модуль «Создание»:

* тесты;
* тестовые вопросы;
* варианты ответа;
* критерии оценивания теста;
* готовый файл.

Из данного списка категорий для модуля «Создание»:

* входными будут: тест, тестовые вопросы, варианты ответа, критерии оценивания теста.
* выходными данными будут: готовый файл.

Таким образом, выделены следующие категории данных, с которыми будет работать модуль «Редактирование»:

* тесты;
* тестовые вопросы;
* варианты ответа;
* критерии оценивания теста;
* готовый файл;
* преобразованный файл.

Из данного списка категорий для модуля «Редактирование»:

* входными будут: тест, тестовые вопросы, варианты ответа, критерии оценивания теста, готовый файл.
* выходными данными будут: преобразованный файл.

Для прохождения тестирования пользователь должен:

* выбрать создание или редактирование теста.

При выборе создание теста:

* дать название тесту;
* ввести количество вопросов;
* количество ответов на каждый вопрос;
* ввести сами вопросы и ответы;
* указать балл за каждый вопрос;
* сохранить тест в файл.

При выборе редактирование теста:

* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.

Каждый тест содержит:

* краткое название;
* описание теста;
* список тестовых вопросов.

В системе должны быть предусмотрены тестовые вопросы следующего вида:

* закрытая форма с одним возможным вариантом ответа;
* закрытая форма с несколькими возможными вариантами ответа.

Для обоих случаев, для каждого тестового вопроса необходимо хранить:

* текст вопроса;
* вид вопроса (один или несколько правильных ответов);
* балл сложности вопроса;
* список вариантов ответа.

Каждый вариант ответа содержит текст варианта ответа и признак того, что данный вариант ответа является правильным.

Для хранения ответов пользователя необходимо для каждого тестового вопроса хранить список вариантов ответа, и для каждого варианта ответа хранить признак того, что данный вариант ответа был выбран пользователем в процессе прохождения тестирования.

Возможно несколько вариантов хранения ответов пользователей:

* будут сохраняться тестовые вопросы, варианты ответа на тестовый вопрос, выбранные пользователем варианты и признак правильности ответа на вопрос;
* будут сохраняться варианты ответа на тестовый вопрос, выбранные пользователем варианты и признак правильности ответа на вопрос. Сами тестовые вопросы не хранятся;
* будут сохраняться выбранные пользователем варианты и признак правильности ответа на вопрос. Сами тестовые вопросы и варианты ответа на тестовый вопрос не хранятся;
* будут сохраняться только признаки правильности ответа на вопрос;
* будут сохраняться только баллы сложности вопросов (без текста вопросов и вариантов ответа) и признак правильности ответа.

Первые вариант наиболее сложен с точки зрения реализации, но позволяет восстановить процесс тестирования даже после того, как содержание исходного теста будет отредактировано.

Второй и третий варианты не позволят восстановить процесс тестирования после изменения теста.

Четвертый вариант наиболее простой в реализации, но не позволит вычислять оценку по заданному критерию оценивания, поэтому выберем для реализации пятый вариант.

Балл сложности вопроса – целое число, задаваемое при редактировании теста.

На сам тестовый вопрос можно ответить полностью правильно, частично правильно (выбраны не все правильные варианты ответа, не выбран ни один неправильный), полностью неправильно (выбран неправильный вариант ответа или не выбран ни один вариант ответа). Закодируем возможные ситуации (см. таблица 2):

Таблица 2 – Коды для вариантов ответа пользователей

|  |  |  |
| --- | --- | --- |
| № | Ситуация | Код |
| 1 | Выбраны все правильные варианты ответа, не выбран ни один неправильный | 0 |
| 2 | Выбраны некоторые правильные варианты ответа, не выбран ни один неправильный | 1 |
| 3 | Не выбран ни один правильный варианты ответа или выбран хотя бы один неправильный | 2 |

* + - 1. **3. Проектирование модульной структуры программы**

На основе анализа решаемых программой задач, был получен следующий список модулей (Таблица 3).

Таблица 3 – Модули системы

|  |  |
| --- | --- |
| Функция системы | Модуль |
| Создавать тест путем ввода информации | Создание |
| Редактировать ранее сохраненные тесты | Редактирование |

Модуль «Создание»:

* ввод тестовых вопросов;
* ввод вариантов ответа на тестовый вопрос;
* указание (ввода), какой вариант правильный;
* указание балла за вопрос.

Модуль «Редактирование»:

* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.

Требования

(см. расшифровку ниже)

Прав. ответ

Балл

Файл теста

Создание и редактирование тестов

А 0

Ответы

Вопросы

Модуль «Редактирование»

Модуль «Создание»

Рис. 1. Контекстная диаграмма программы «создание и редактирование тестов» (методология IDEF0)

Под Требованиями подразумевается:

* закрытая форма тестовых вопросов;
* ограниченное количество вариантов ответа (не более 9);
* каждый вопрос оценивается баллом;
* возможно 1 или несколько правильных ответов.

Требования

(см. расшифровку ниже)

Прав.ответ

Создание

А 1

Файл теста

Балл

Ответы

Вопросы

Файл теста

Корректирующая тест информация

Редактирование

А 2

Модуль «Редактирование»

Модуль «Создание»

Рис. 2. Диаграмма декомпозиции первого уровня(модуль «Создание» и модуль «Редактирование»)

Под Требованиями подразумевается:

* закрытая форма тестовых вопросов;
* ограниченное количество вариантов ответа (не более 9);
* каждый вопрос оценивается баллом;
* возможно 1 или несколько правильных ответов.

Требования

(см. расшифровку ниже)

Ввод вопроса

А 1.1

Вопросы

1

Ответы

Ввод ответов

А 1.2

2

Балл

Ввод балла

А 1.3

3

Прав.ответ

Ввод прав.ответ

А 1.4

Файл теста

Сохр.теста в файл

А 1.5

4

Модуль «Создание»

Рис. 3. Диаграмма декомпозиции второго уровня(модуль «Создание»)

1-передача данных о введенных вопросах.

2-передача данных о введенных вопросах и вариантов ответов.

3-передача введенных вопросов, вариантов ответов, балла.

4-передача введенных вопросов, вариантов ответа, балов, правильных ответов.

Под Требованиями подразумевается:

* закрытая форма тестовых вопросов;
* ограниченное количество вариантов ответа (не более 9);
* каждый вопрос оценивается баллом;
* возможно 1 или несколько правильных ответов.

Требования

(см. расшифровку ниже)

Доб.нов.вопрос

А 2.1

передача данных

Доб.нов.ответ

А 2.2

Удал.вопроса

А 2.3

Удал.ответа

А 2.4

Корректирующая тест информация

Изм.Вопроса

А 2.5

Изм.ответа

А 2.6

Файл теста

Изм. балла

А 2.7

Сохр.теста в файл

А 2.8

Модуль «Редактирование»

Рис. 4. Диаграмма декомпозиции второго уровня(модуль «Редактирование»)

Под Требованиями подразумевается:

* закрытая форма тестовых вопросов;
* ограниченное количество вариантов ответа (не более 9);
* каждый вопрос оценивается баллом;
* возможно 1 или несколько правильных ответов.
  + - 1. **4.Проектирование алгоритмов**
  1. Выполнение декомпозиции задач на набор процедур и функций

В процессе анализа задачи было принято решение, разбиение исходной задачи на 2 модуля:

* модуль «Редактирование»;
* модуль «Создание».

Модуль «Создание» будет содержать следующие функции:

* ввод тестовых вопросов;
* ввод вариантов ответа на тестовый вопрос;
* указание (ввода), какой вариант правильный;
* сохранение в файл;
* указание балла за вопрос.

Модуль «Редактирование» будет содержать следующие функции:

* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильного ответа;
* сохранение изменений в файл.

Блок схема. Основная программа.

начало

Передвижение по меню, выбор действий

2

1

Модуль «Создание»

Модуль «Редактирование»

Программа завершила работу

выход

Блок схема. Модуль «Создание» (при выборе ).

4

5

3

2

1

Указание балла за вопрос

Ввод прав варианта ответа

Ввод вар.ответов

Ввод тест.вопросов

variant

Сохранение в файл

Модуль завершил работу

выход

Блок схема. Модуль «Редактирование» (при выборе ).

Введите имя файла

Передвижение по меню, выбор действий

4

5

3

2

1

Удаление вар.ответа

Удаление вопроса

Добавление нов.вар.ответа

Добавление нов.вопроса

Изменение вопроса

6

Изменение ответа на вопрос

7

Изменение балла за вопрос

8

Сохранение в файл

Модуль завершил работу

выход

* 1. Определение сигнатур процедур и функций.

Модуль «Создание»:

**Procedure** createqu(**var** n10:integer; **var** ot1:**array of** test1);

Процедура создания вопросов.

Входные данные:

* количество вопросов (на входе пустое);
* массив записей (каждая запись – вопрос, на входе пустой).

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов.

**Procedure** createan(**var** n10:integer; **var** ot2:**array of** test1);

Процедура создания ответов.

Входные данные:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов и ответов.

**Procedure** createball(**var** n10:integer; **var** ot3:**array of** test1);

Процедура определения сложности вопроса (балл).

Входные данные:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов и ответов.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов и балла сложности.

**Procedure** creategoodan(**var** n10:integer; **var** ot4:**array of** test1);

Процедура определения правильного ответа.

Входные данные:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов и балла сложности.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, балла сложности, номеров правильных ответов.

Модуль «Редактирование»:

**Procedure** insertqu(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

Процедура добавления вопроса в тест.

Входные параметры:

* имя файла, куда будет записан тест с добавленным вопросом;
* файловая переменная, для связи процедуры с реальным файлом;
* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* файловая переменная, для связи процедуры с реальным файлом;
* обновленное количество записей в массиве записей (обновленное количество вопросов в тесте);
* очищенный массив с вопросами (без теста).

**Procedure** insertan(**var** n10:integer; **var** ot1:**array of** test2);

Процедура добавления ответа к конкретному вопросу теста.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* количество обновленных записей в массиве записей;
* обновленный массив с вопросами (добавлен ответ к конкретному вопросу).

**Procedure** deletequ(**var** n10:integer; **var** ot1:**array of** test2);

Процедура удаления вопроса из теста.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* обновленное количество записей в массиве записей (обновленное количество вопросов в тесте);
* обновленный массив с вопросами (с 1 удаленным вопросом).

**Procedure** deletean(**var** n10:integer; **var** ot1:**array of** test2);

Процедура удаления ответа у конкретного вопроса теста.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* количество обновленных записей в массиве записей;
* обновленный массив с вопросами (удален ответ у конкретного вопроса).

**Procedure** editqu(**var** n10:integer; **var** ot1:**array of** test2);

Процедура изменения вопроса.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

**Procedure** editan(**var** n10:integer; **var** ot1:**array of** test2);

Процедура изменения ответа.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

**Procedure** editball(**var** n10:integer; **var** ot1:**array of** test2);

Процедура изменения балла за вопрос.

Входные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

**Procedure** rf(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

Процедура чтения теста из файла в массив записей.

Входные параметры:

* имя файла, откуда будет считан тест;
* файловая переменная, для связи процедуры с реальным файлом;
* количество записей в массиве записей (количество вопросов в тесте, на входе не определено);
* пустой массив записей.

Выходные параметры:

* файловая переменная, для связи процедуры с реальным файлом;
* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

**Procedure** wf1(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

Процедура записи теста в файл.

Входные параметры:

* имя файла, куда будет записан тест;
* файловая переменная, для связи процедуры с реальным файлом;
* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

Выходные параметры:

* файловая переменная, для связи процедуры с реальным файлом;
* количество записей в массиве записей (количество вопросов в тесте);
* массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.
  + - 1. **5. Тестирование программы**
         1. В связи с невозможностью и большой трудоемкостью разработки тестов для автоматизации тестирование, проведем вручную тестирование созданных алгоритмов. Для ручного тестирование, мы запускаем мою программу и видим меню управление, которое состоит из 2 частей (создание теста и редактирование теста). Для выбора нужного пункта меню используются клавиши "w" и "s" для перемещения вверх и вниз, "пробел" для выбора, "z" для выхода.

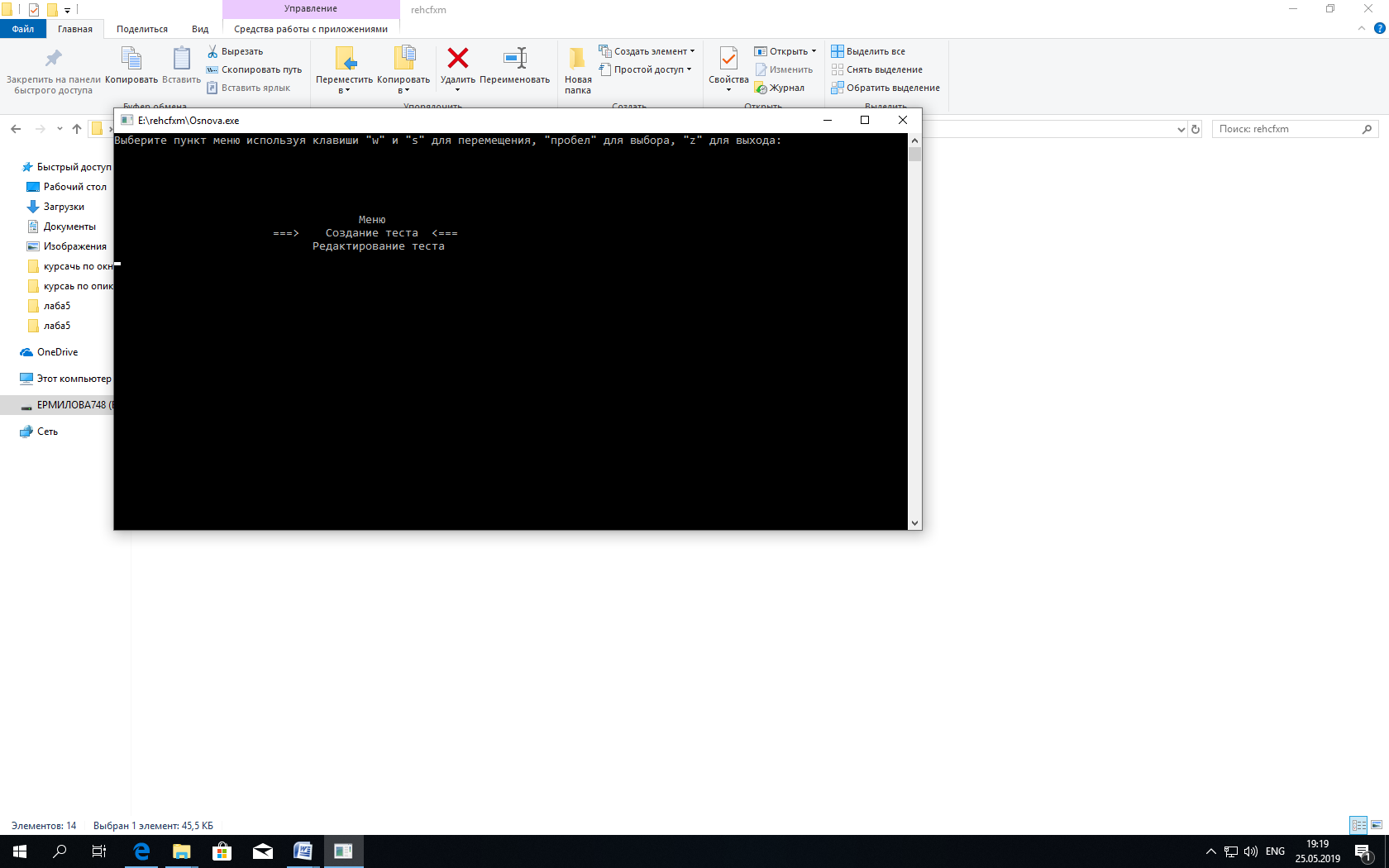
****

Рис. 5. Главное меню программы

После выбора пункта 1 в меню, мы видим, что нам необходимо ввести количество вопросов в тесте.

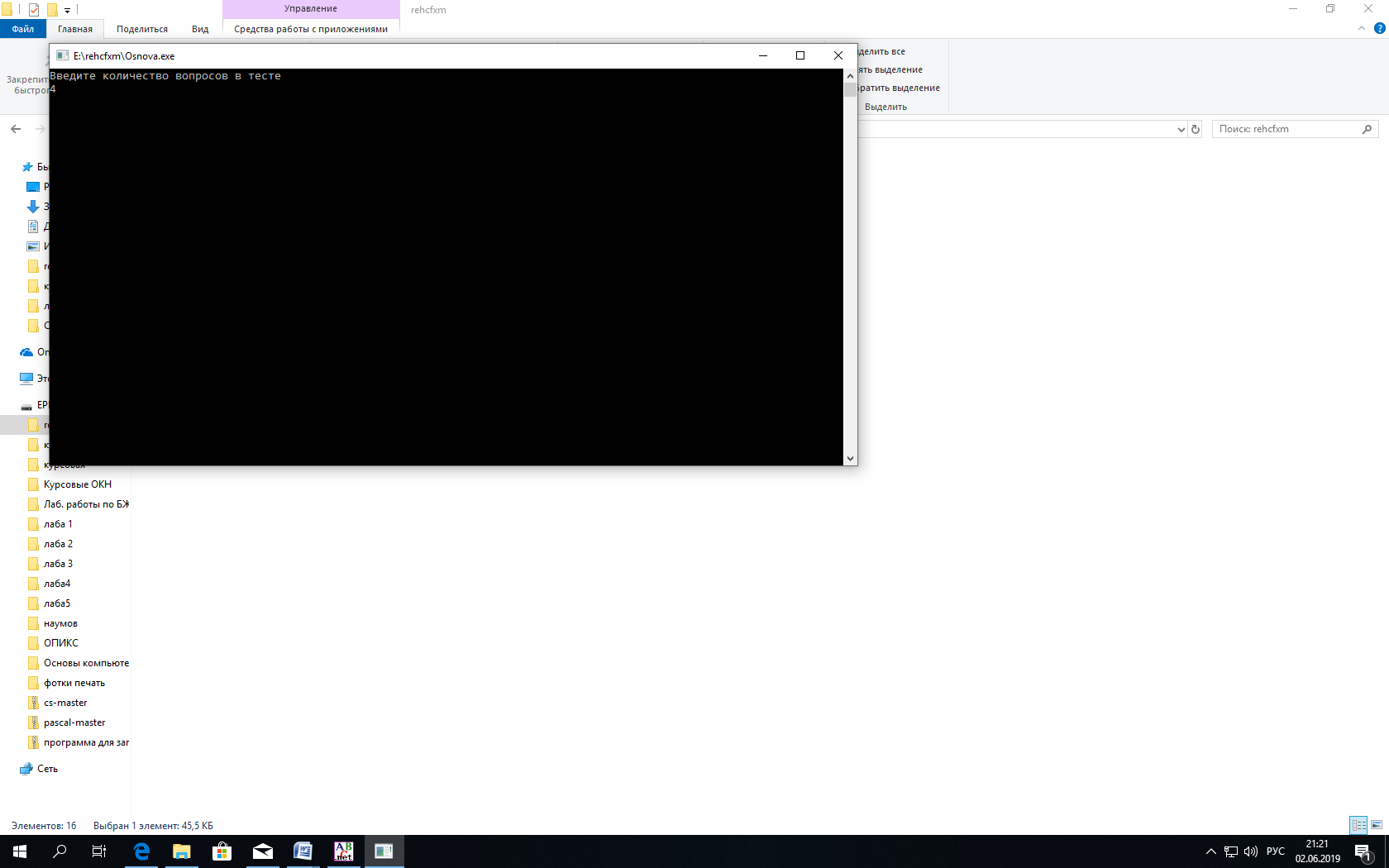
****

Рис. 6. Ввод количества вопросов

Затем мы вводим каждый вопрос

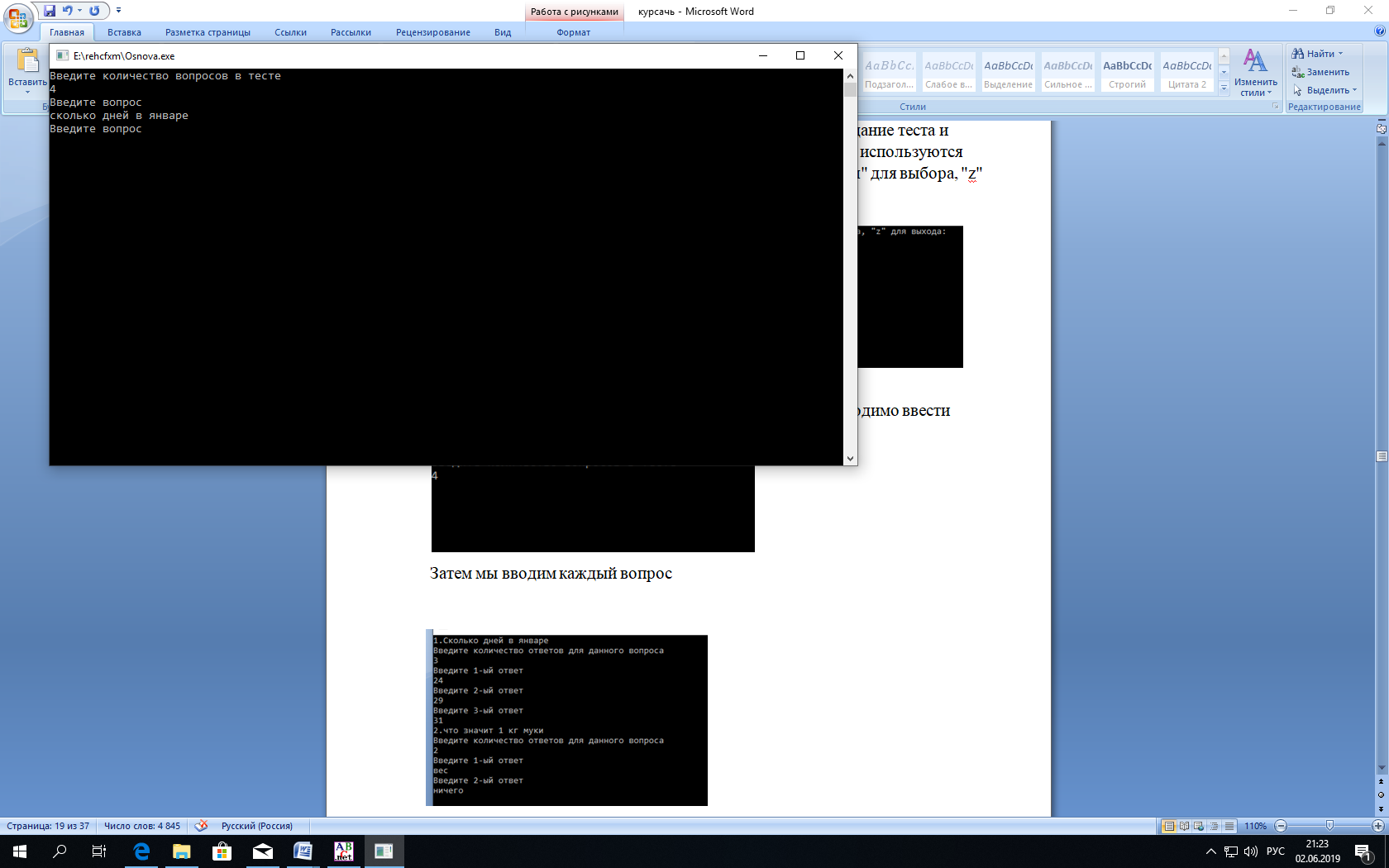


Рис. 7. Ввод каждого вопроса

Далее мы вводим на каждый вопрос, количество правильных ответов и вводим ответы.

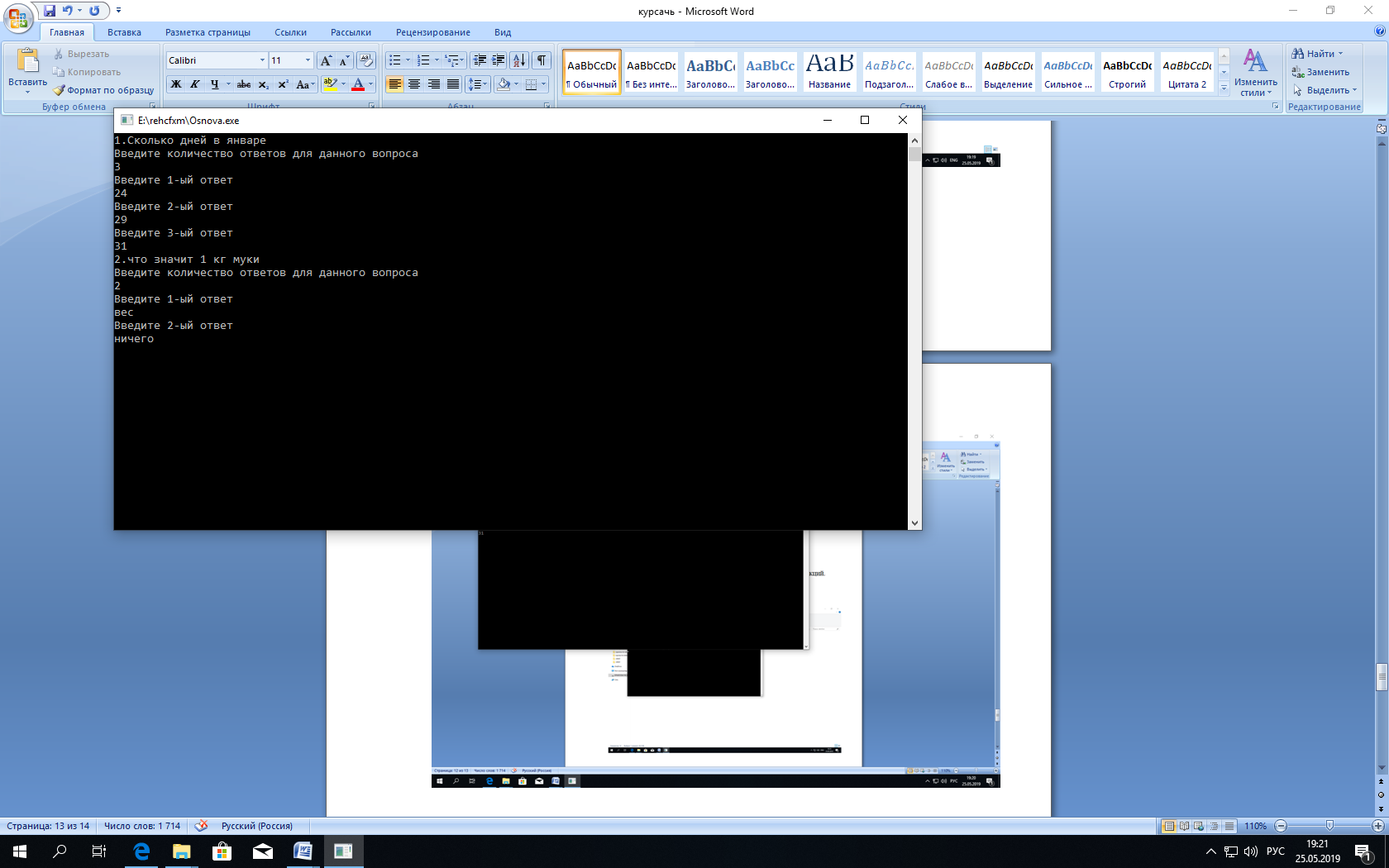


Рис. 8. Ввод количества правильных ответов на вопрос

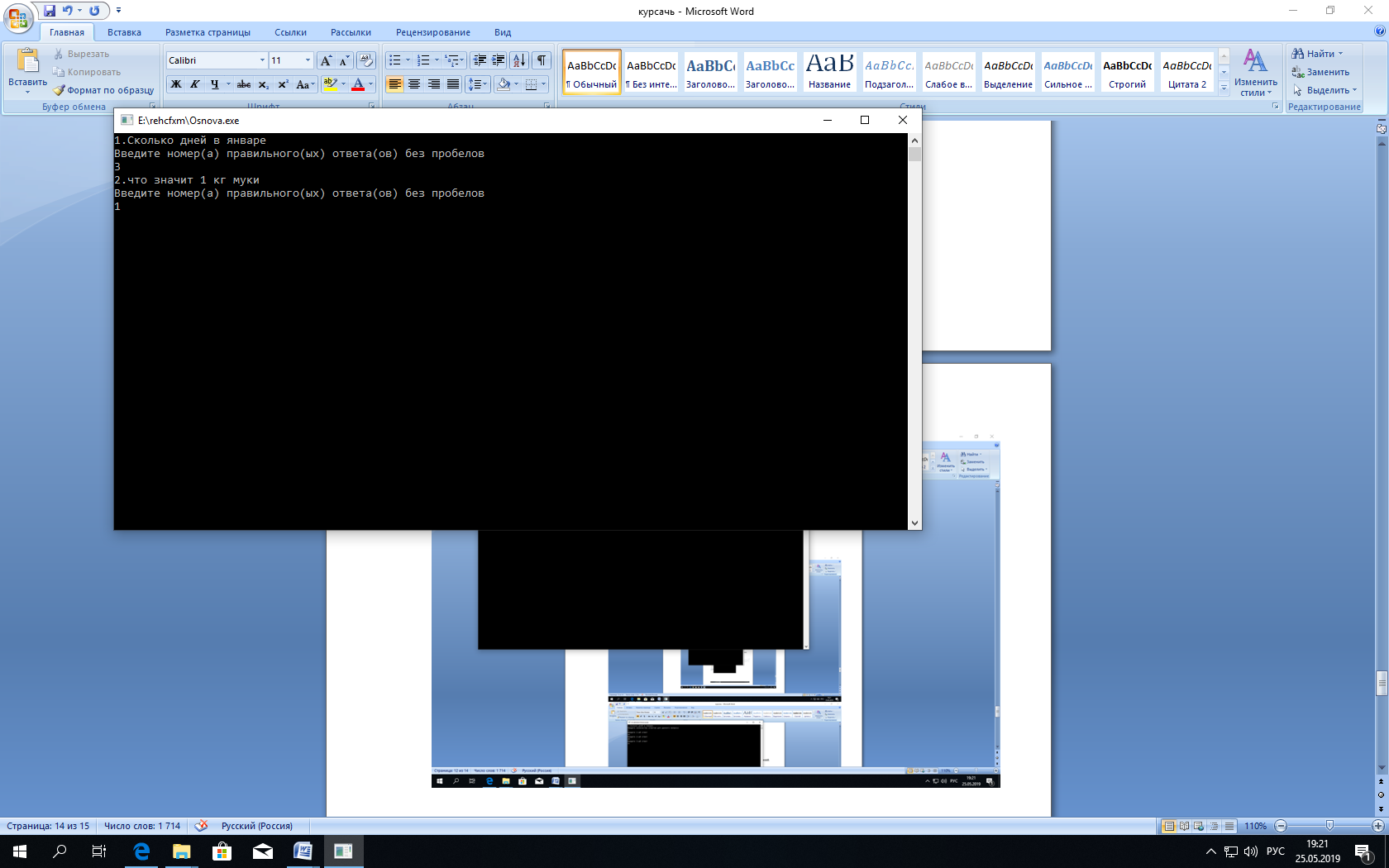
Затем вводим номера правильных ответов

Рис. 9. Ввод номеров правильных ответов

Потом вводим балл за каждый вопрос

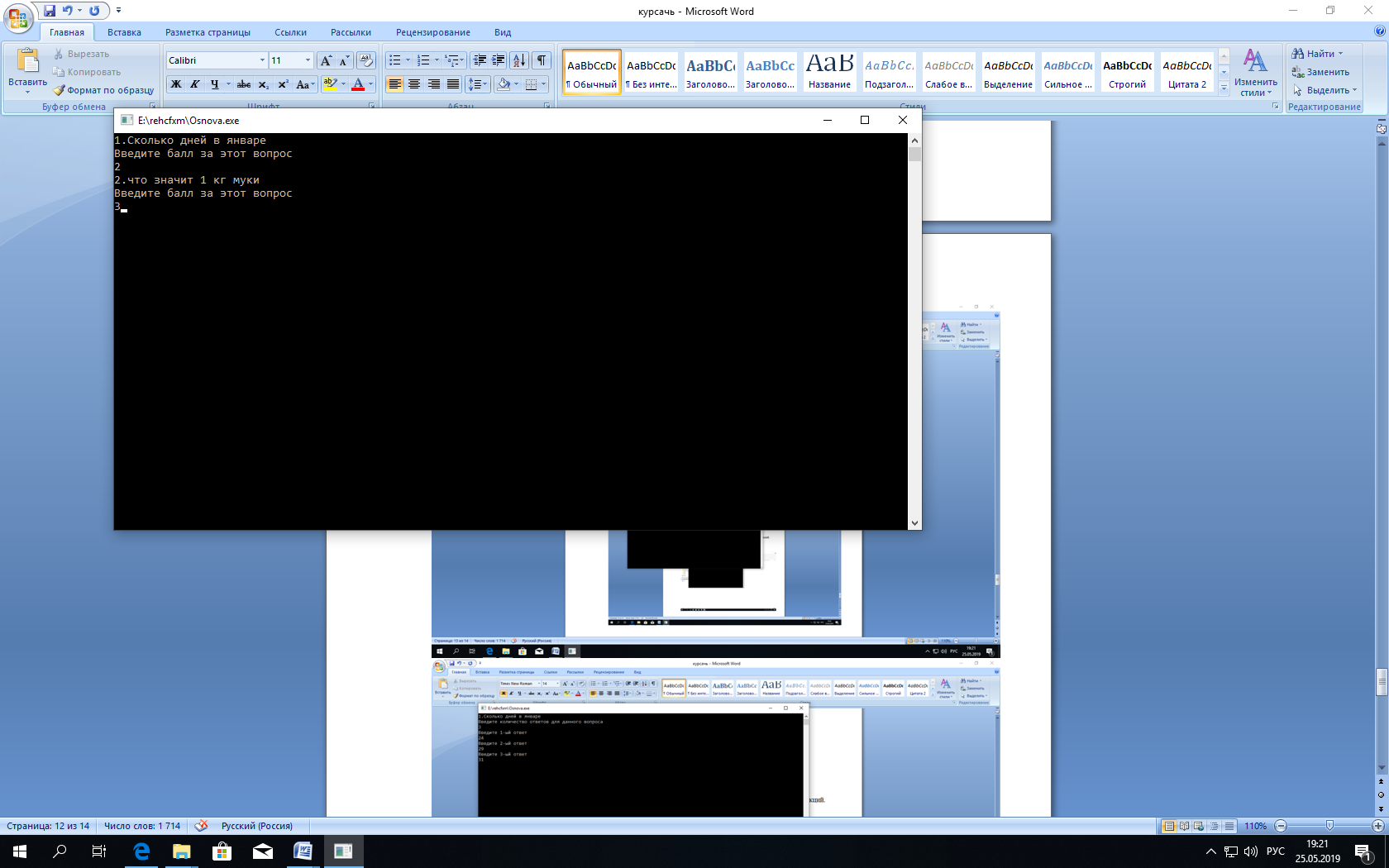


Рис. 10. Ввод балла за каждый вопрос

Затем вводим имя файла и проверяем занесение данных в файле

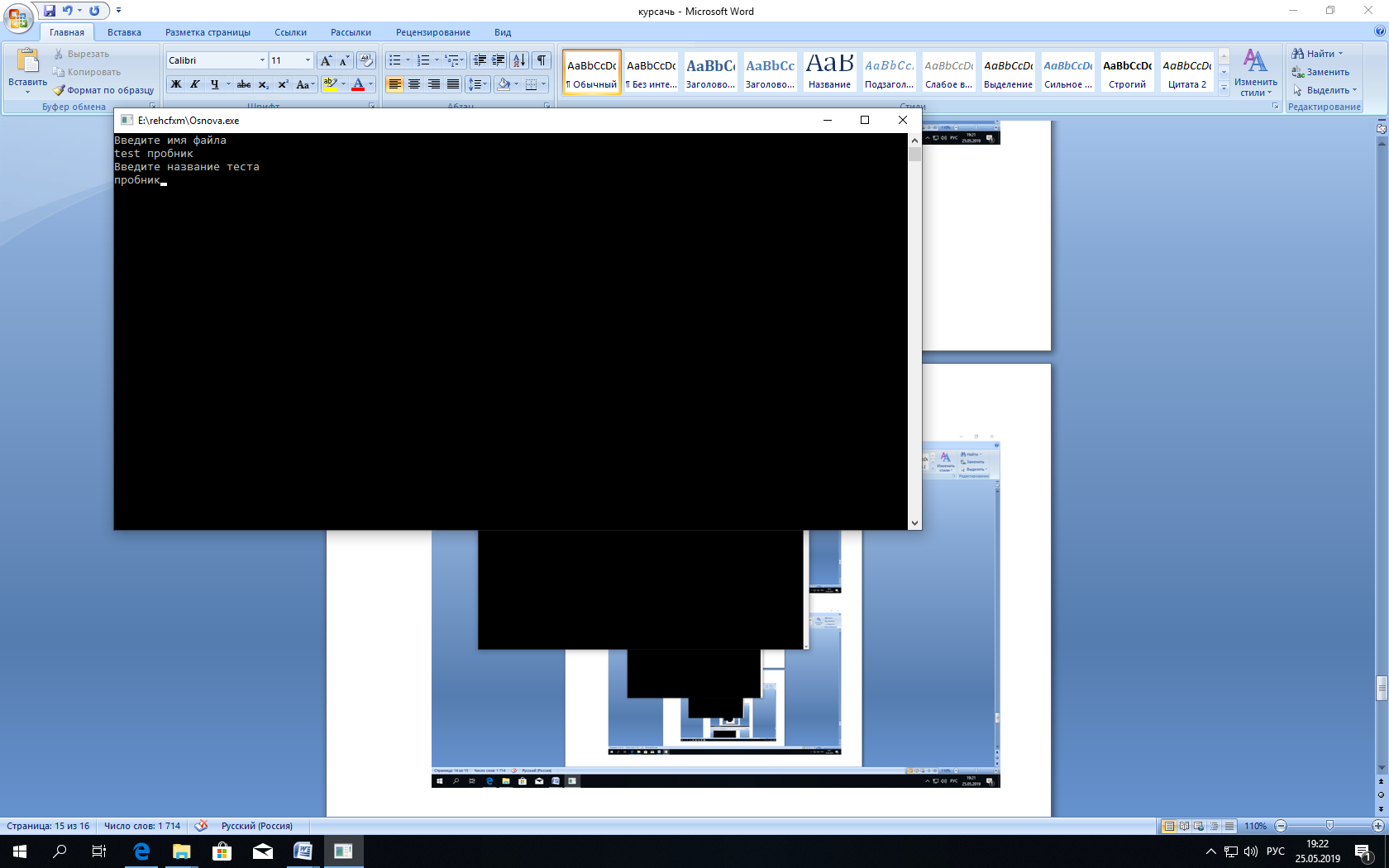


Рис. 11. Ввод имени файла и название теста

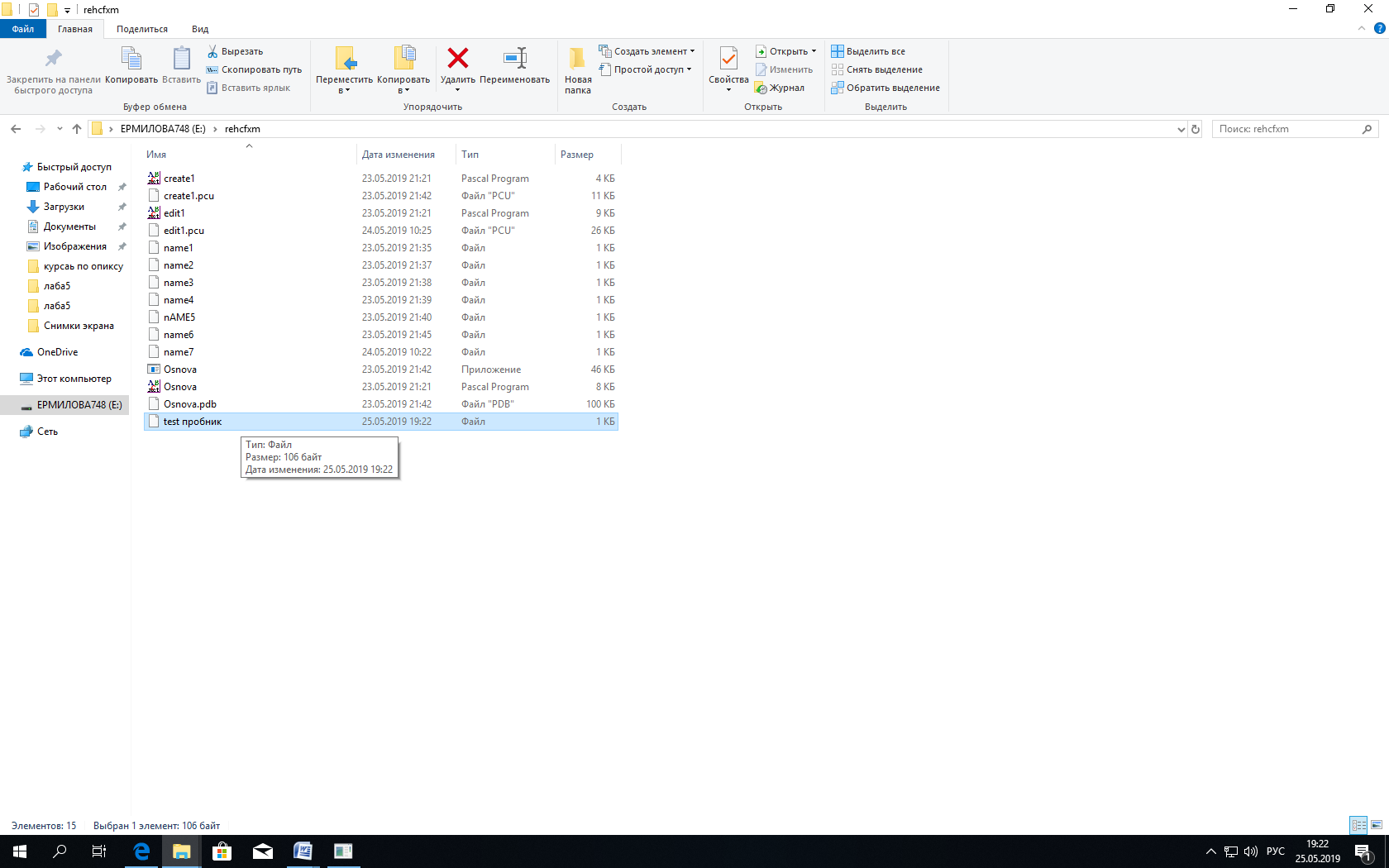


Рис.12. Проверка сохранения файла

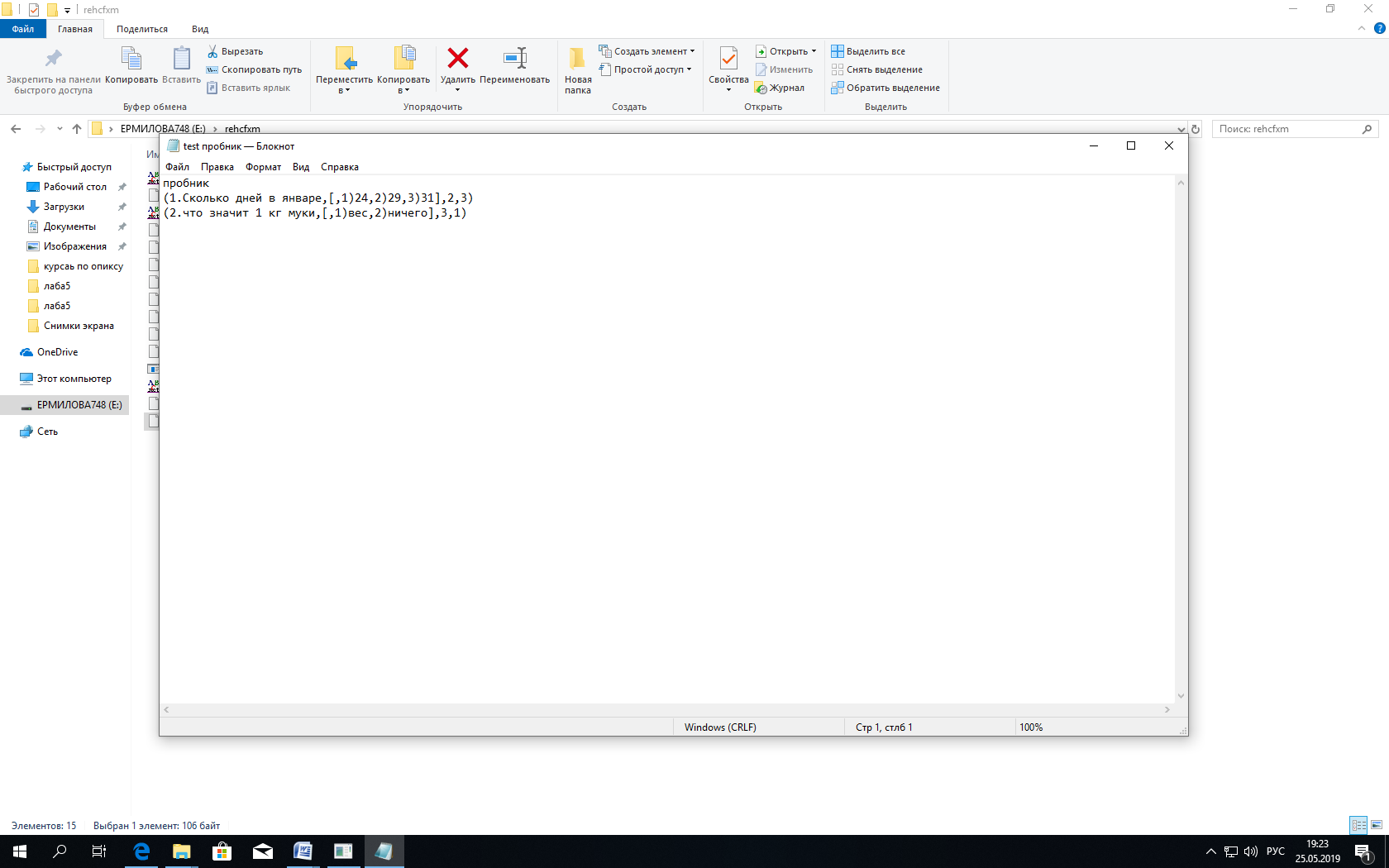


Рис. 13. Содержание сохраненного файла

После выбора пункта 2 в меню, мы видим, что нам необходимо ввести имя файла для изменения.

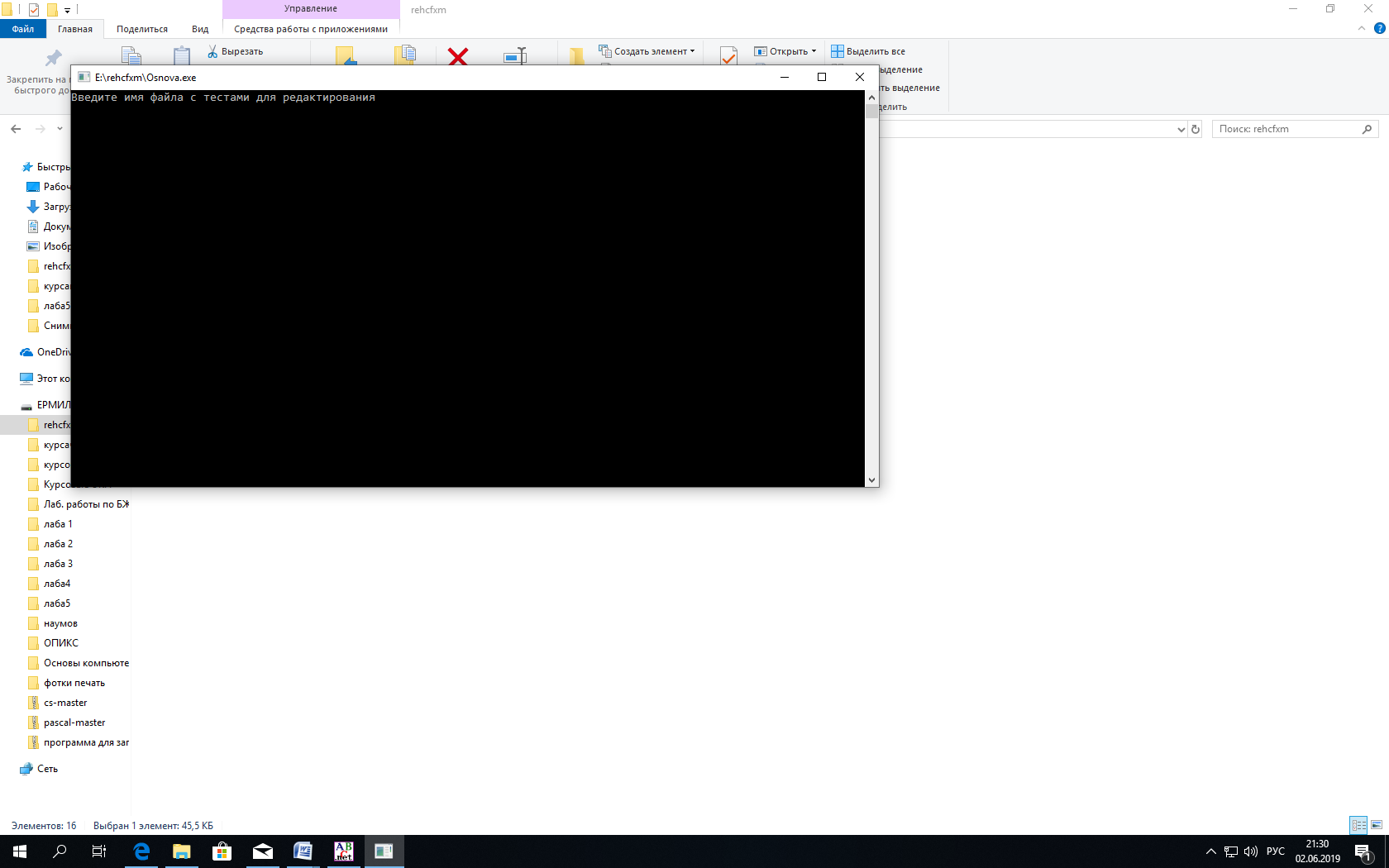


Рис. 14. Ввод имени файла для изменения

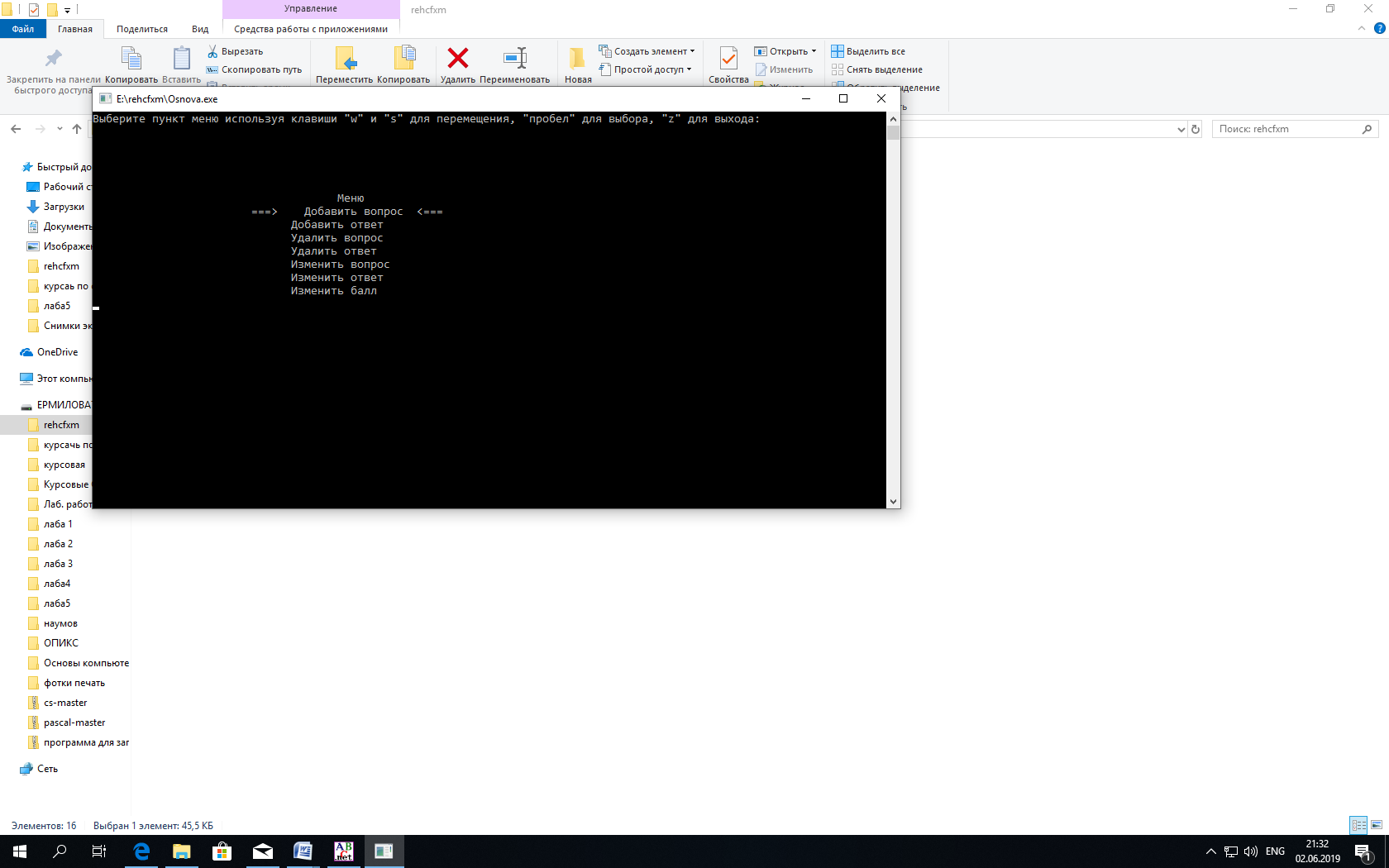
Затем появляется список команд.

Рис. 15. Главное меню подпрограммы «Редактирование тестов»

Затем при выборе 1,2 ,мы проводим все те же действия , что мы проводили в пункте «создание теста», для создания вопроса и для ввода вариантов ответов.

Затем п.3 мы указываем номер вопроса, который необходимо удалить.

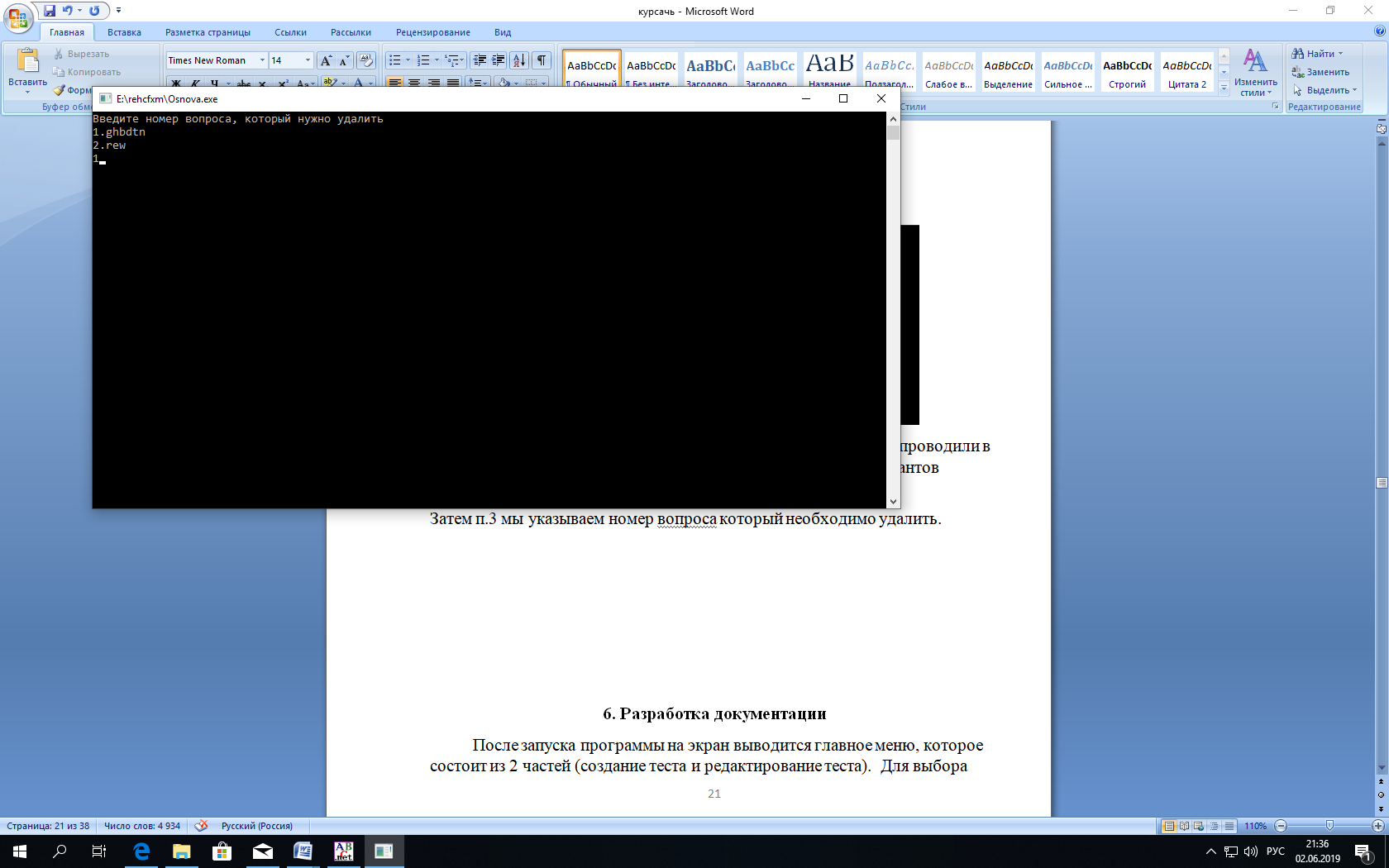


Рис. 16. Ввод номера вопроса, который необходимо удалить

Затем п.3 мы указываем номер вопроса, в котором нужно удалить ответ и указываем номер ответа, который хотим удалить.

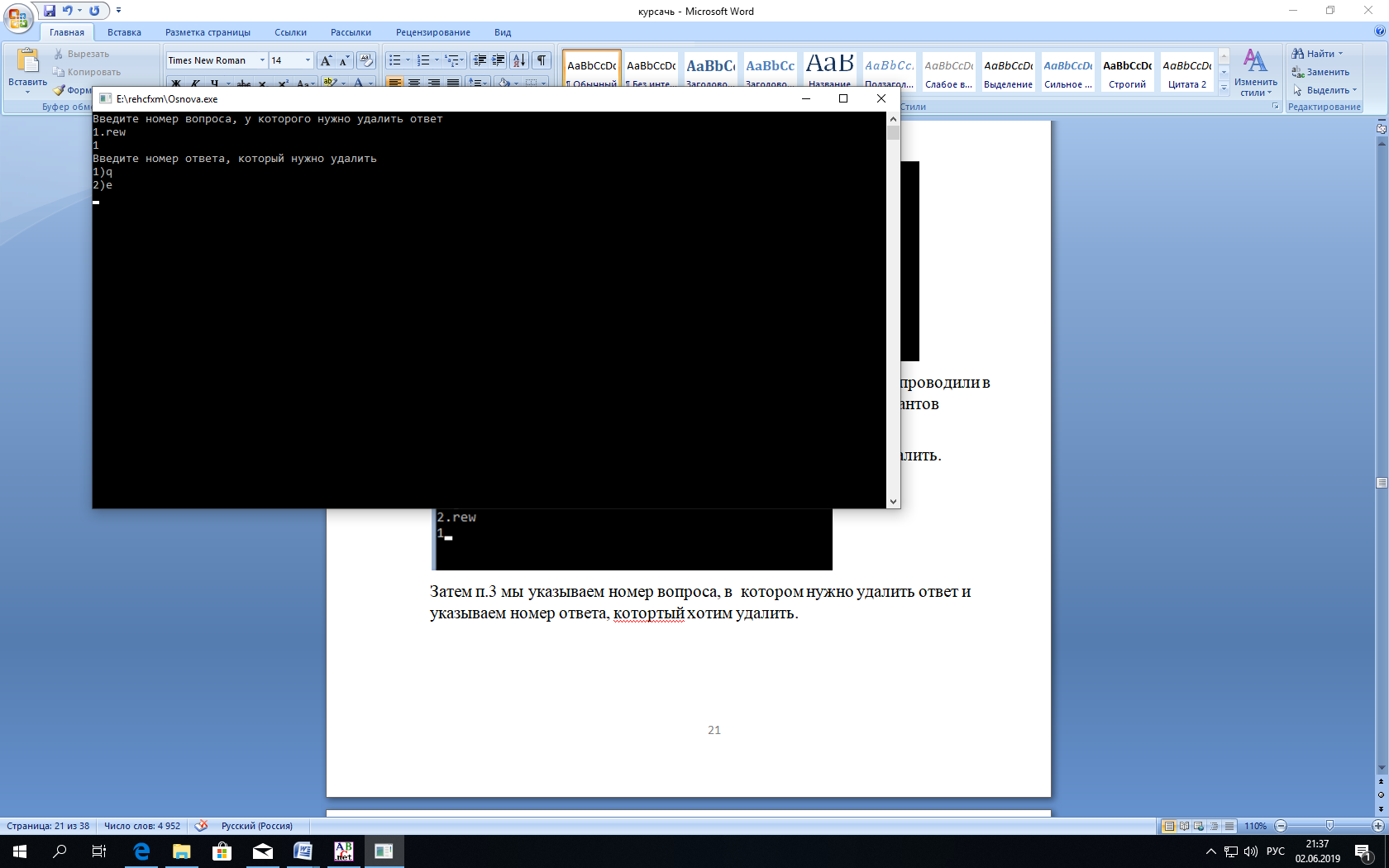


Рис. 17. Ввод номера вопроса, у которого нужно удалить указанный номер ответа

В пункте 4,5,6 вносим изменения в вопросы, ответы, и балл

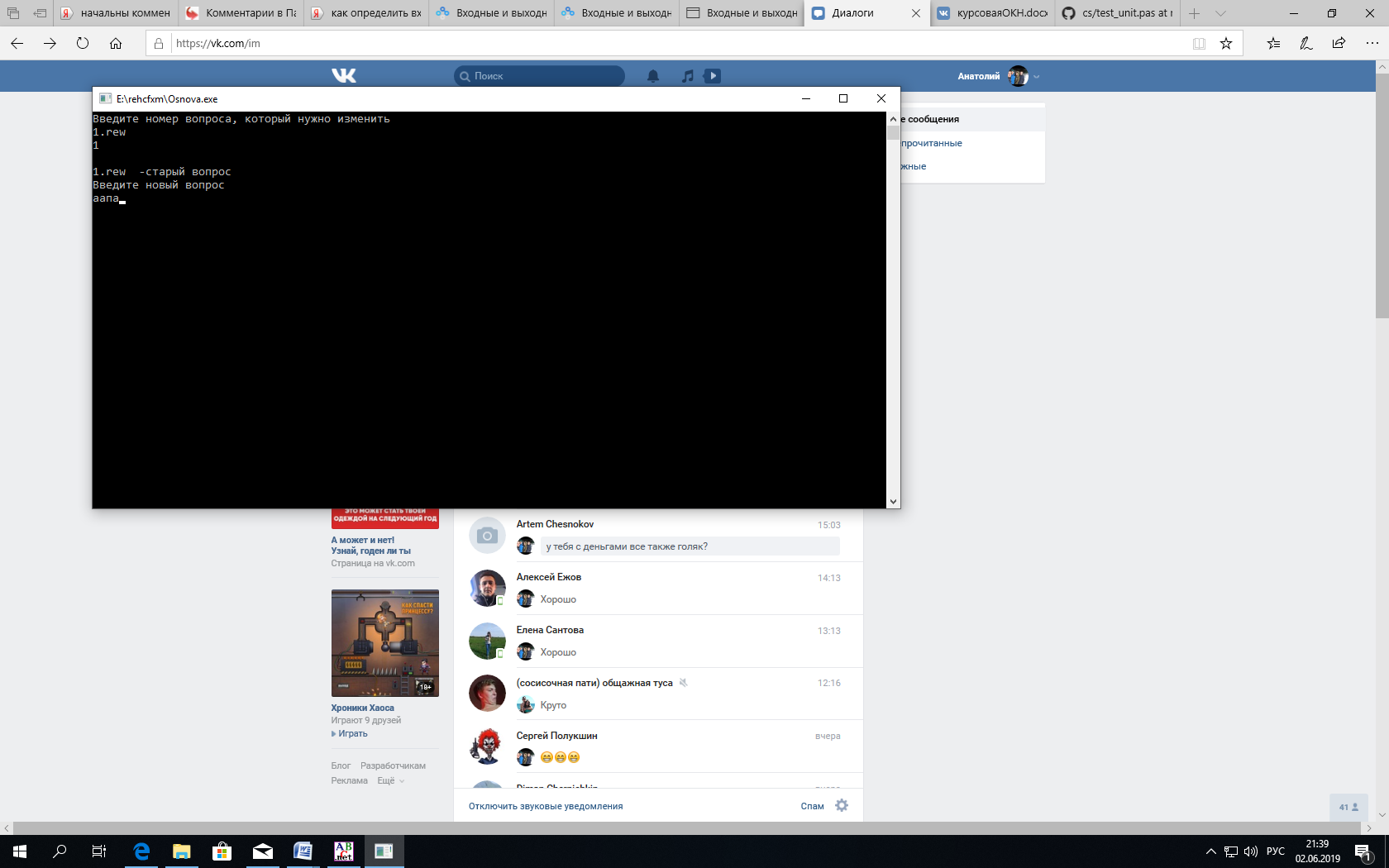


Рис. 18. Изменяем формулировку выбранного вопроса

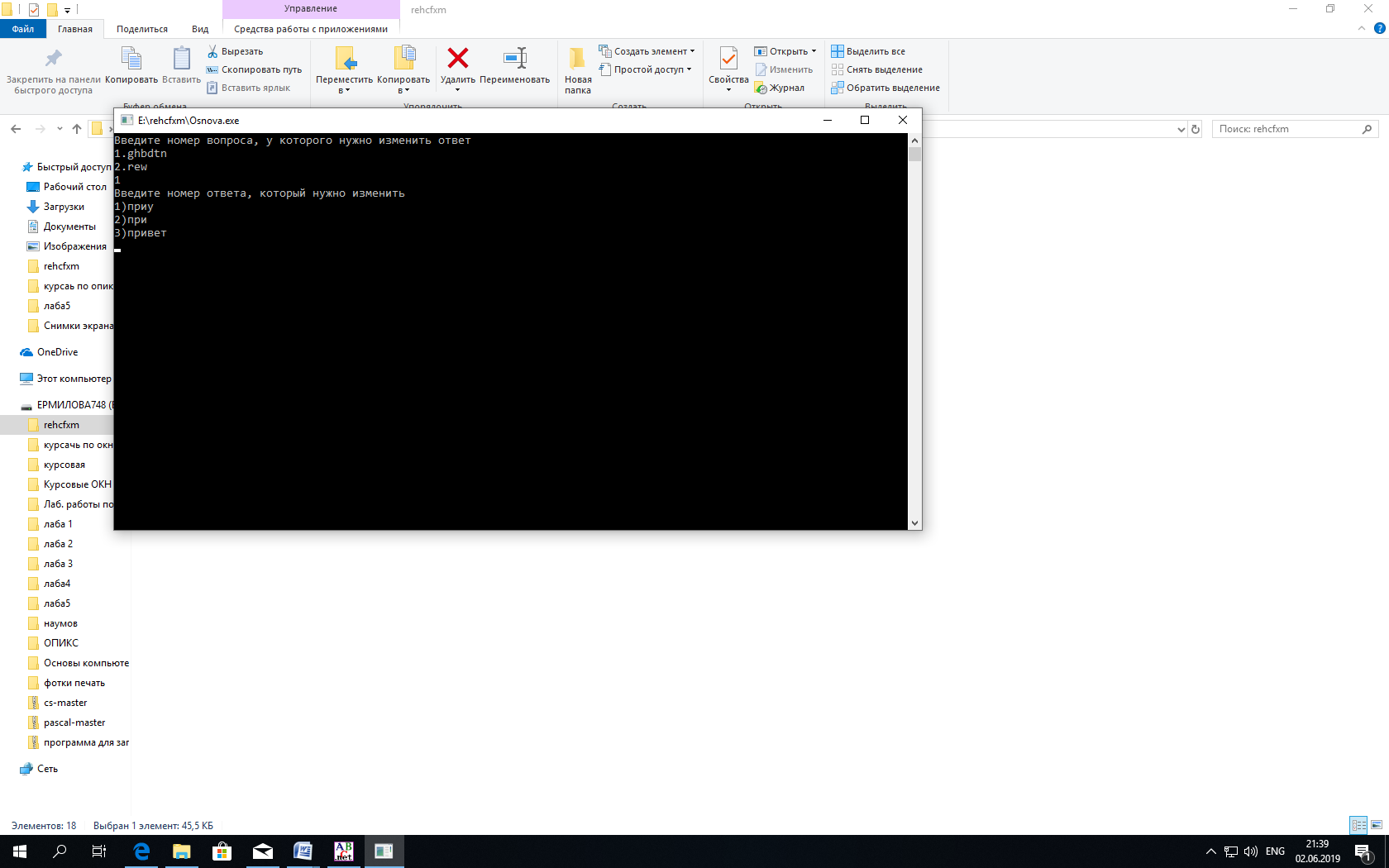


Рис. 19. Изменяем формулировку выбранного ответа у конкретного вопроса

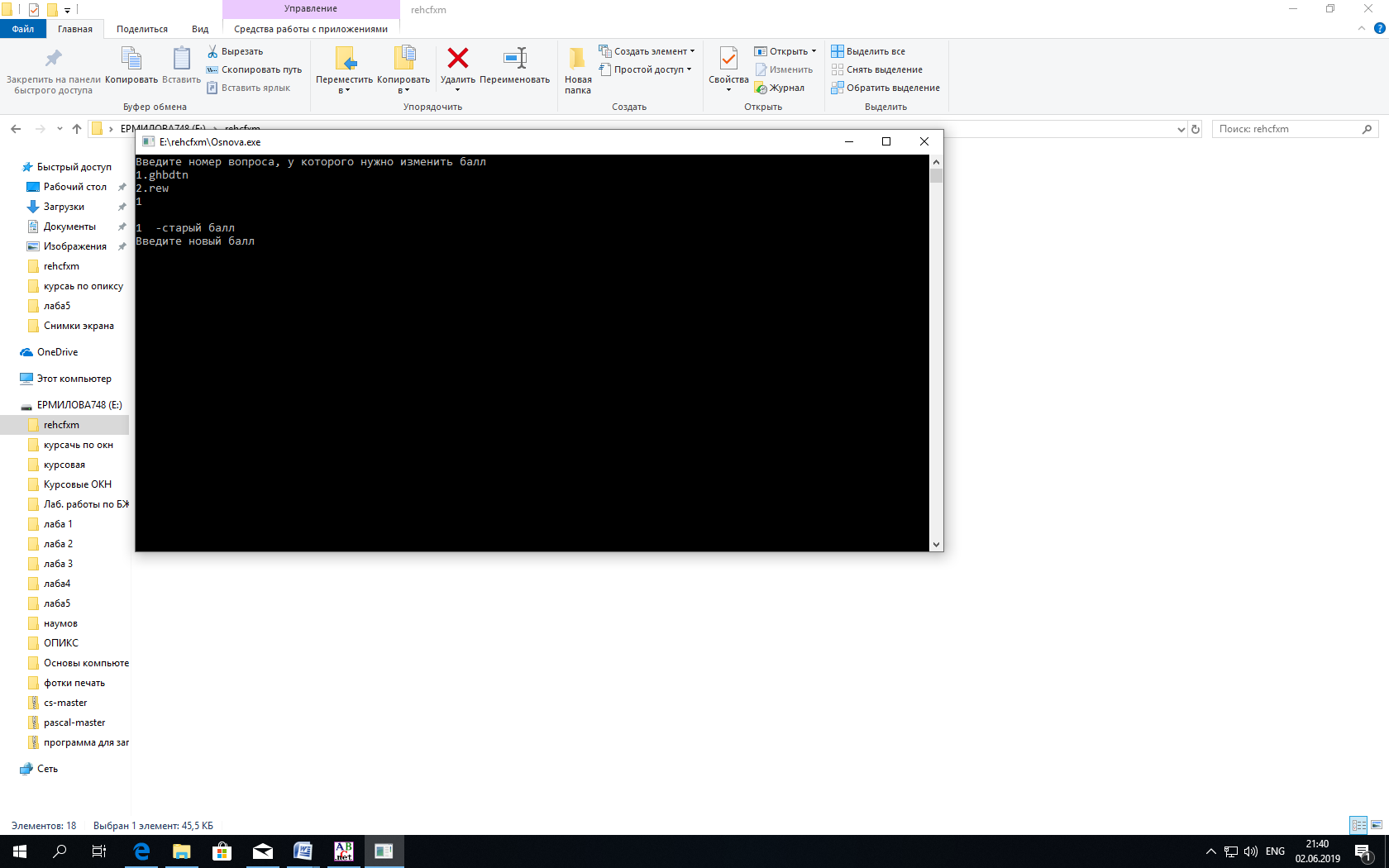


Рис. 20. Изменяем балл за выбранный вопрос

**6. Разработка документации**

После запуска программы на экран выводится главное меню, которое состоит из 2 частей (создание теста и редактирование теста). Для выбора нужного пункта меню используются клавиши "w" и "s" для перемещения вверх и вниз, "пробел" для выбора, "z" для выхода.

При выборе п. 1:

После выбора данного пункта меню будет переход на выполнение функции или процедуры.

В п.1.1 меню на экране появится сообщение "Введите количество вопросов в тесте", затем появится сообщение "Введите вопрос". Следует ввести вопросы по очереди.

В п.1.2 меню на экране появится сообщение "Введите количество ответов к вопросу», затем после ввода появиться сообщение «Введите ответ». Следует ввести ответы по очереди.

В п.1.3 меню на экране появится сообщение "Введите номер правильного ответа". Следует ввести номер или номера правильных ответов.

В п.1.4 меню на экране появится сообщение "Введите балл за вопрос". Следует ввести балл за вопрос.

В п.1.5 в меню на экране появится сообщение "Введите имя файла", затем необходимо ввести имя и сохранить файл.

При выборе п. 2 в меню, на экране появится сообщение "Введите имя файла с тестами для редактирования», затем на экране появится следующий список команд:

* добавление нового вопроса;
* добавление новых вариантов ответа;
* удаление вопроса;
* удаление ответа;
* изменение отдельных текстовых вопросов;
* изменение отдельных вариантов ответа;
* изменение признака правильности ответа.

После запуска программы на экран выводится главное меню, которое состоит из 2 частей (создание теста и редактирование теста). Для выбора нужного пункта меню используются клавиши "w" и "s" для перемещения вверх и вниз, "пробел" для выбора, "z" для выхода.

При выборе п. 2.1 в меню на экране появится сообщение "введите новый вопрос", затем необходимо произвести похожие действия, что и в п.1

При выборе п. 2.2 в меню на экране появится сообщение "введите номер вопроса, к которому нужно добавить ответ", затем необходимо ввести новый ответ на данный вопрос.

При выборе п. 2.3 в меню на экране появится сообщение "введите номер вопроса, который нужно удалить", затем необходимо ввести номер вопрос.

При выборе п. 2.4 в меню на экране появится сообщение "введите номер вопроса, у которого нужно удалить ответ", затем необходимо ввести номер ответа для удаления.

При выборе п. 2.5 в меню на экране появится сообщение "введите номер вопроса, который нужно изменить", затем необходимо ввести номер вопроса и сам вопрос.

При выборе п. 2.6 в меню на экране появится сообщение "введите номер вопроса, у которого нужно изменить ответ", затем необходимо ввести номер вопроса и, потом номер ответа, который нужно изменить и ввести сам ответ.

При выборе п. 2.7 в меню на экране появится сообщение "введите номер вопроса, у которого нужно изменить балл", затем необходимо ввести номер вопроса и изменить балл.

Затем после произведения изменений в указанном файле, необходимо сохранить эти изменения в новый файл.

**Заключение**

Подводя итоги, можно сказать, что работа была успешна проведена. Выполнены все задания, был разработаны тестовые вопросы для дисциплины «Основы компьютерных наук» по теме  архитектура ЭВМ и разработан модуль, осуществляющий  редактирование тестовых вопросов.

**Библиографический список**

1. Фуфаев Э. В. Ф94 Базы данных : учеб. пособие дли студ. учреждений сред, проф. образования— 7-е изд., М : Издательский центр «Академия», 2012. — 320 с.
2. Абрамян М. Э. А13 Структуры данных в PascalABC.NET. Выпуск 1. Массивы и последовательности. Запросы. — Ростов н/Д : Изд-во ЮФУ, 2016. — 119 с.: ил.
3. Абрамян М. Э. А 13 Структуры данных в PascalABC.NET. Выпуск 2. Минимумы и максимумы. Списки, множества, словари, стеки и очереди. Многомерные структуры. — Ростов н/Д : Изд-во ЮФУ, 2016. — 118 с

**Приложение А (тестовые вопросы)**

Вопрос №1. Что такое быстродействие ЭВМ?

Варианты ответа:

1. такого понятия не существует.

2. совокупность элементов и их связей.

3. дэто многоуровневая иерархия аппаратно-программных средств, из которых строится ЭВМ..

4. среднестатистическое число операций (команд), выполняемых ЭВМ в единицу времени.

Правильный ответ (4)

Вопрос №2. Чем быстродействие, характеризуется?

Варианты ответа:

1. Производительность.

2. Надежность.

3. Точность.

4. стандартные универсальные тесты для ЭВМ, предназначенных для крупномасштабных вычислений, специализированные тесты для конкретных областей применения компьютеров.

Правильный ответ (4)

Вопрос №3. Единица измерения быстродействия компьютера

Варианты ответа:

1. операции в секунду

2. операции в минуты.

3. операции в сутки.

4. операции в час.

Правильный ответ (1)

Вопрос №4. Что такое надежность компьютера?

Варианты ответа:

1.запас прочности;

2.нет такого понятия;

3.запас точности;

4.это способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного времени;

Правильный ответ (4)

Вопрос №5. Что такое надежность изделия?

Варианты ответа:

1. состояние, при котором изделие выполняет все заданные функции с параметрами установленными для него в технической документации.

2. событие, которое заключается в нарушении работоспособности.

3. состояние, при котором изделие выполняет все заданные функции, но с некоторыми пониженными показателями качества

4. свойство изделия выполнять заданные функции, сохраняя свои эксплуатационные показатели в течение требуемого промежутка времени или требуемой наработки.

Правильный ответ (4)

Вопрос №6. Что такое безотказность?

Варианты ответа:

1. свойство изделия сохранять работоспособность в течение заданной наработки без перерывов.

2. свойство изделия сохранять работоспособность до предельного состояния (до списания) с перерывами на техническое обслуживание и ремонт

3. свойство изделия обнаруживать, устранять и предупреждать неисправность и отказы путем проведения технического обслуживания и ремонта.

4. свойство изделия сохранять свои эксплуатационные показатели в течение и после срока транспортирования или хранения его на складе.

Правильный ответ (1)

Вопрос №7. Что такое ремонтопригодность?

Варианты ответа:

1. свойство изделия обнаруживать, устранять и предупреждать неисправность и отказы путем проведения технического обслуживания и ремонта.

2. стандартные универсальные тесты для ЭВМ, предназначенных для крупномасштабных вычислений, специализированные тесты для конкретных областей применения компьютеров с перерывами на техническое обслуживание и ремонт

3. свойство изделия обнаруживать, устранять и предупреждать неисправность и отказы путем проведения технического обслуживания и ремонта.

4. свойство изделия сохранять свои эксплуатационные показатели в течение и после срока транспортирования или хранения его на складе.

Правильный ответ (1)

Вопрос №8. Что такое показатель эффективности (ПЭ)?

Варианты ответа:

1. стандартные универсальные тесты для ЭВМ, предназначенных для крупномасштабных вычислений, специализированные тесты для конкретных областей применения компьютеров.

2. среднестатистическое число операций (команд), выполняемых ЭВМ в единицу времени. компьютеров с перерывами на техническое обслуживание и ремонт

3. состояние, при котором изделие выполняет все заданные функции, но с некоторыми пониженными показателями качества.

4. количественная характеристика ЭВМ, рассматриваемая применительно к определенным условиям ее функционирования.

Правильный ответ (4)

Вопрос №9. Что такое критерий эффективности?

Варианты ответа:

1. отношение результатов — величины прибыли к величине суммарных затрат на создание и эксплуатацию ЭВМ.

2. это правило, служащее для сравнительной оценки качества вариантов вычислительных машин

3. состояние, при котором изделие выполняет все заданные функции, но с некоторыми пониженными показателями качества.

4. количественная характеристика ЭВМ, рассматриваемая применительно к определенным условиям ее функционирования.

Правильный ответ (2)

Вопрос №10. Что такое эффективность?

Варианты ответа:

1. совокупность элементов и их связей.

2. это способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного времени

3. это свойство выполнять поставленную цель в заданных условиях использования и с определенным качеством.

4. совокупность элементов и их связей.

Правильный ответ (3)

**Приложение Б (листинг программных модулей)**

Основная программа

// ------Программа "Создание и редактирование тестов".

//

// Содержит два модуля:

// -модуль"Создание"

// -модуль "Редактирование"

//

//Назначение модуля "Создание":

// -Ввод тестовых вопросов;

// -Ввод вариантов ответа на тест.вопрос;

// -Указание (ввода), какой вариант правильный;

// -Указание балла за вопрос;

// -Сохранение в файл.

//

//

//Назначение модуля "Редактирование":

// -Открытие ранее сохраненного файла;

// -Добавление нового вопроса;

// -Добавление новых вариантов ответа;

// -Удаление вопроса;

// -Удаление ответа;

// -Изменение отдельных текстовых вопросов;

// -Изменение отдельных вариантов ответа;

// -Изменение признака правильности ответа;

// -Сохранение изменений в файл.

//

//Дата последнего изменения:

// 02.06.2019

// -------------------------------------------------------------

**uses** crt,create1,edit1;

**var**

c,c1:char;

variant,variant1,ii,nn:integer;

namefail:string;

zapis1:text;

ot:**array of** test2;

to1:**array of** test1;

**begin**

variant:=1;

variant1:=1;

**while** c<>'z' **do**

**begin**

**if** variant=0 **then** variant:=2;

**if** variant=3 **then** variant:=1;

writeln('Выберите пункт меню используя клавиши "w" и "s" для перемещения, "пробел" для выбора, "z" для выхода:');

writeln;writeln;writeln;writeln;writeln;writeln(' Меню');

**if** variant=1 **then** write (' ===> ') **else** write(' '); write(' Создание теста '); **if** variant=1 **then** writeln (' <===') **else** writeln;

**if** variant=2 **then** write (' ===> ') **else** write(' '); write(' Редактирование теста '); **if** variant=2 **then** writeln (' <===') **else** writeln;

C:=readkey;clrscr;

**if** c='w' **then** variant:=variant-1;

**if** c='ц' **then** variant:=variant-1;

**if** c='s' **then** variant:=variant+1;

**if** c='ы' **then** variant:=variant+1;

**if** c='я' **then** c:='z';

**if** c=' ' **then**

**case** variant **of**

1:

**begin**

clrscr;

createqu(nn,to1);

clrscr;

createan(nn,to1);

clrscr;

createball(nn,to1);

clrscr;

creategoodan(nn,to1);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf(namefail,zapis1,nn,to1);

to1:=nil;

**end**;

2:

**Begin**

clrscr;

writeln('Введите имя файла с тестами для редактирования');

readln(namefail);

rf(namefail,zapis1,nn,ot);

**while** c1<>'z' **do**

**begin**

**if** variant1=0 **then** variant1:=7;

**if** variant1=8 **then** variant1:=1;

writeln('Выберите пункт меню используя клавиши "w" и "s" для перемещения, "пробел" для выбора, "z" для выхода:');

writeln;writeln;writeln;writeln;writeln;writeln(' Меню');

**if** variant1=1 **then** write (' ===> ') **else** write(' '); write(' Добавить вопрос '); **if** variant1=1 **then** writeln (' <===') **else** writeln;

**if** variant1=2 **then** write (' ===> ') **else** write(' '); write(' Добавить ответ '); **if** variant1=2 **then** writeln (' <===') **else** writeln;

**if** variant1=3 **then** write (' ===> ') **else** write(' '); write(' Удалить вопрос '); **if** variant1=3 **then** writeln (' <===') **else** writeln;

**if** variant1=4 **then** write (' ===> ') **else** write(' '); write(' Удалить ответ '); **if** variant1=4 **then** writeln (' <===') **else** writeln;

**if** variant1=5 **then** write (' ===> ') **else** write(' '); write(' Изменить вопрос '); **if** variant1=5 **then** writeln (' <===') **else** writeln;

**if** variant1=6 **then** write (' ===> ') **else** write(' '); write(' Изменить ответ '); **if** variant1=6 **then** writeln (' <===') **else** writeln;

**if** variant1=7 **then** write (' ===> ') **else** write(' '); write(' Изменить балл '); **if** variant1=7 **then** writeln (' <===') **else** writeln;

C1:=readkey;clrscr;

**if** c1='w' **then** variant1:=variant1-1;

**if** c1='ц' **then** variant1:=variant1-1;

**if** c1='s' **then** variant1:=variant1+1;

**if** c1='ы' **then** variant1:=variant1+1;

**if** c1='я' **then** c1:='z';

**if** c1=' ' **then**

**case** variant1 **of**

1:

**Begin**

clrscr;

writeln('Введите имя файла для исправленного теста');

readln(namefail);

insertqu(namefail,zapis1,nn,ot);

clrscr;

rf(namefail,zapis1,nn,ot);

clrscr;

**End**;

2:

**Begin**

clrscr;

insertan(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

3:

**Begin**

clrscr;

deletequ(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

4:

**Begin**

clrscr;

deletean(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

5:

**Begin**

clrscr;

editqu(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

6:

**Begin**

clrscr;

editan(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

7:

**Begin**

clrscr;

editball(nn,ot);

clrscr;

writeln('Введите имя файла');

readln(namefail);

wf1(namefail,zapis1,nn,ot);

**End**;

**end**;

**end**;

**end**;

**end**;

**end**;

**end**.

Модуль «Создание»

**Unit** create1;

// -------------------------------------------------------------

// Модуль "Создание" тестов.

//

//Назначение данного модуля:

// -Ввод тестовых вопросов;

// -Ввод вариантов ответа на тест.вопрос;

// -Указание (ввода), какой вариант правильный;

// -Указание балла за вопрос;

// -Сохранение в файл.

//

//Дата последнего изменения:

// 02.06.2019

// -------------------------------------------------------------

**Unit** create1;

**Interface**

//создание теста

**Type** test1=**record**

qu1:string; //тестовый вопрос

an1:**array of** string; //вариант ответа на тестовый вопрос

ball1:real; //балл за ответ на вопрос

goodan1:integer; //правильный ответ

**end**;

//Создание вопроса

**Procedure** createqu(**var** n10:integer; **var** ot1:**array of** test1);

//Создание ответа

**Procedure** createan(**var** n10:integer; **var** ot2:**array of** test1);

//Ввод балла за вопрос

**Procedure** createball(**var** n10:integer; **var** ot3:**array of** test1);

//ввод правильного ответа

**Procedure** creategoodan(**var** n10:integer; **var** ot4:**array of** test1);

//запись в файл

**Procedure** wf(name:string; **var** zapis:text; **var** n10:integer; **var** ot5:**array of** test1);

**Implementation**

// -------------------------------------------------------------

// Процедура создания вопросов.

//Входные данные:

//- количество вопросов (на входе пустое);

//- массив записей (каждая запись – вопрос, на входе пустой).

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** createqu;

**Var**

i,n:integer;

s,s1:string;

**Begin**

writeln('Введите количество вопросов в тесте');

readln(n);

n10:=n+1;

setlength(ot1,n10);

**for** i:=1 **to** n **do**

**begin**

Writeln('Введите вопрос');

readln(s);

str(i,s1); //Номер итерации=номеру вопроса, для этого я его переношу в строковую переменную

s:=s1+'.'+s; //Перед вопросом подставляю его номер, полученный из предыдущей строки

ot1[i].qu1:=s;

**end**;

**End**;

// -------------------------------------------------------------

// Процедура создания ответов.

//Входные данные:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов и ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** createan;

**Var**

i,i1,n,n1:integer;

s,s1:string;

**Begin**

setlength(ot2,n10);

n:=n10-1;

**for** i:=1 **to** n **do**

**begin**

writeln(ot2[i].qu1);

writeln('Введите количество ответов для данного вопроса');

readln(n1);

n1:=n1+1;

setlength(ot2[i].an1,n1);

n1:=n1-1;

**for** i1:=1 **to** n1 **do**

**begin**

writeln('Введите ',i1,'-ый ответ');

readln(s);

str(i1,s1);

s:=s1+')'+s;

ot2[i].an1[i1]:=s;

**end**;

**end**;

**End**;

// -------------------------------------------------------------

//Процедура определения сложности вопроса (балл).

//Входные данные:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов и ответов.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов и балла сложности.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** createball;

**Var**

i,n:integer;

**Begin**

setlength(ot3,n10);

n:=n10-1;

**for** i:=1 **to** n **do**

**begin**

writeln(ot3[i].qu1);

writeln('Введите балл за этот вопрос');

readln(ot3[i].ball1);

**end**;

**end**;

// -------------------------------------------------------------

//Процедура определения правильного ответа.

//Входные данные:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов и балла сложности.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, балла сложности, номеров правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** creategoodan;

**Var**

i,n:integer;

**Begin**

setlength(ot4,n10);

n:=n10-1;

**for** i:=1 **to** n **do**

**begin**

writeln(ot4[i].qu1);

writeln('Введите номер(а) правильного(ых) ответа(ов) без пробелов');

readln(ot4[i].goodan1);

**end**;

**end**;

// -------------------------------------------------------------

// Процедура записи теста в файл.

//Входные параметры:

//- имя файла, куда будет записан тест;

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** wf;

**var**

i,n:integer;

s:string;

**begin**

{связываем файловую переменную с файлом

открываем файл для записи

закрываем файл}

Assign(zapis,name);

Rewrite(zapis);

writeln('Введите название теста');

readln(s);

writeln(zapis,s);

n:=n10-1;

**for** i:=1 **to** n **do**

**begin**

writeln(zapis,ot5[i]);

**end**;

Close(zapis);

**end**;

**end**.

Модуль «Редактирование».

// -------------------------------------------------------------

// Модуль "Редактирование" тестов.

//

//Назначение данного модуля:

// -Открытие ранее сохраненного файла;

// -Добавление нового вопроса;

// -Добавление новых вариантов ответа;

// -Удаление вопроса;

// -Удаление ответа;

// -Изменение отдельных текстовых вопросов;

// -Изменение отдельных вариантов ответа;

// -Изменение признака правильности ответа;

// -Сохранение изменений в файл

//

//Дата последнего изменения:

// 02.06.2019

// -------------------------------------------------------------

**Unit** edit1;

**Interface**

//редактирование теста

**Type** test2=**record**

qu2:string; //тестовый вопрос

an2:**array of** string; //вариант ответа на тестовый вопрос

ball2:real; //балл за ответ на вопрос

goodan2:integer; //правильный ответ

**end**;

//Добавление нового вопроса

**Procedure** insertqu(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

//Добавление нового ответа

**Procedure** insertan(**var** n10:integer; **var** ot1:**array of** test2);

//Удаление вопроса

**Procedure** deletequ(**var** n10:integer; **var** ot1:**array of** test2);

//Удаление ответа

**Procedure** deletean(**var** n10:integer; **var** ot1:**array of** test2);

//Изменение существующего вопроса

**Procedure** editqu(**var** n10:integer; **var** ot1:**array of** test2);

//изменение существующего ответа

**Procedure** editan(**var** n10:integer; **var** ot1:**array of** test2);

//Изменение балла за вопрос

**Procedure** editball(**var** n10:integer; **var** ot1:**array of** test2);

//Чтение из файла

**Procedure** rf(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

//Запись в файл

**Procedure** wf1(name:string; **var** zapis:text; **var** n10:integer; **var** ot1:**array of** test2);

**Implementation**

// -------------------------------------------------------------

// Процедура добавления вопроса в тест.

//Входные параметры:

//- имя файла, куда будет записан тест с добавленным вопросом;

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- файловая переменная, для связи процедуры с реальным файлом;

//- обновленное количество записей в массиве записей (обновленное количество вопросов в тесте);

//- очищенный массив с вопросами (без теста).

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** insertqu;

**var**

i,n,n1:integer;

s,s1,s2:string;

ot2:test2;

**Begin**

n10:=n10+1;

setlength(ot1,n10);

n:=n10-1;

writeln('Введите вопрос');

readln(ot2.qu2);

str(n,s2);

ot2.qu2:=s2+'.'+ot2.qu2;

writeln('Введите количество ответов');

readln(n1);

n1:=n1+1;

setlength(ot2.an2,n1);

n1:=n1-1;

**for** i:=1 **to** n1 **do**

**begin**

writeln('Введите ',i,'-й ответ');

readln(s);

str(i,s1);

s:=s1+')'+s;

ot2.an2[i]:=s;

**end**;

writeln('Введите балл за вопрос');

readln(ot2.ball2);

writeln('Введите номер(а) правильного(ых) ответа(ов) без пробелов');

readln(ot2.goodan2);

Assign(zapis,name);

Rewrite(zapis);

writeln('Введите название теста');

readln(s);

writeln(zapis,s);

n:=n-1;

**for** i:=1 **to** n **do**

**begin**

writeln(zapis,ot1[i]);

**end**;

writeln(zapis,ot2);

Close(zapis);

ot1:=nil;

**end**;

// -------------------------------------------------------------

// Процедура добавления ответа к конкретному вопросу теста.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- количество обновленных записей в массиве записей;

//- обновленный массив с вопросами (добавлен ответ к конкретному вопросу).

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** insertan;

**var**

i,i1,n,n1,n2:integer;

massiv:**array of** string;

s1:string;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, к которому нужно добавить ответ');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

n2:=high(ot1[n1].an2);

n2:=n2+1;

setlength(massiv,n2);

n2:=n2-1;

**for** i1:=1 **to** n2 **do**

**begin**

massiv[i1]:=ot1[n1].an2[i1];

**end**;

ot1[n1].an2:=nil;

n2:=n2+2;

setlength(ot1[n1].an2,n2);

n2:=n2-2;

**for** i1:=1 **to** n2 **do**

**begin**

ot1[n1].an2[i1]:=massiv[i1];

**end**;

n2:=n2+1;

writeln('Введите ',n2,'-й ответ');

readln(ot1[n1].an2[n2]);

str(n2,s1);

ot1[n1].an2[n2]:=s1+')'+ot1[n1].an2[n2];

**end**;

// -------------------------------------------------------------

// Процедура удаления вопроса из теста.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- обновленное количество записей в массиве записей (обновленное количество вопросов в тесте);

//- обновленный массив с вопросами (с 1 удаленным вопросом).

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** deletequ;

**var**

n4:real;

i,n,n1,n2,n3:integer;

s1:string;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, который нужно удалить');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

str(n1,s1);

delete(ot1[n].qu2,1,1);

ot1[n].qu2:=s1+ot1[n].qu2;

ot1[n1].qu2:=ot1[n].qu2;

ot1[n1].ball2:=ot1[n].ball2;

ot1[n1].goodan2:=ot1[n].goodan2;

n2:=high(ot1[n].an2);

ot1[n1].an2:=nil;

n2:=n2+1;

setlength(ot1[n1].an2,n2);

n2:=n2-1;

**for** i:=1 **to** n2 **do**

**begin**

ot1[n1].an2[i]:=ot1[n].an2[i];

**end**;

ot1[n].an2:=nil;

ot1[n].qu2:='';

ot1[n].ball2:=n4;

ot1[n].goodan2:=n3;

n10:=n10-1;

**end**;

// -------------------------------------------------------------

//Процедура удаления ответа у конкретного вопроса теста.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- количество обновленных записей в массиве записей;

//- обновленный массив с вопросами (удален ответ у конкретного вопроса).

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** deletean;

**var**

i,i1,n,n1,n2,n3,n4:integer;

massiv:**array of** string;

s1:string;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, у которого нужно удалить ответ');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

n2:=high(ot1[n1].an2);

writeln('Введите номер ответа, который нужно удалить');

**for** i:=1 **to** n2 **do**

**begin**

writeln(ot1[n1].an2[i]);

**end**;

readln(n3);

str(n3,s1);

n4:=high(ot1[n1].an2);

n4:=n4+1;

setlength(massiv,n4);

n4:=n4-1;

**for** i1:=1 **to** n4 **do**

**begin**

massiv[i1]:=ot1[n1].an2[i1];

**end**;

delete(massiv[n4],1,1);

massiv[n3]:=s1+massiv[n4];

ot1[n1].an2:=nil;

setlength(ot1[n1].an2,n4);

n4:=n4-1;

**for** i1:=1 **to** n4 **do**

**begin**

ot1[n1].an2[i1]:=massiv[i1];

**end**;

**end**;

// -------------------------------------------------------------

// Процедура изменения вопроса.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** editqu;

**var**

i,n,n1:integer;

s,s1:string;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, который нужно изменить');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

writeln;

writeln(ot1[n1].qu2,' -старый вопрос');

writeln('Введите новый вопрос');

readln(s);

str(n1,s1);

ot1[n1].qu2:=s1+'.'+s;

**end**;

// -------------------------------------------------------------

// Процедура изменения ответа.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** editan;

**var**

i,n,n1,n2,n3:integer;

s,s1:string;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, у которого нужно изменить ответ');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

n2:=high(ot1[n1].an2);

writeln('Введите номер ответа, который нужно изменить');

**for** i:=1 **to** n2 **do**

**begin**

writeln(ot1[n1].an2[i]);

**end**;

readln(n3);

writeln;

writeln(ot1[n1].an2[n3],' -старый ответ');

writeln('Введите новый ответ');

readln(s);

str(n3,s1);

ot1[n1].an2[n3]:=s1+')'+s;

**end**;

// -------------------------------------------------------------

//Процедура изменения балла за вопрос.

//Входные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** editball;

**var**

i,n,n1:integer;

**begin**

setlength(ot1,n10);

n:=n10-1;

writeln('Введите номер вопроса, у которого нужно изменить балл');

**for** i:=1 **to** n **do**

**begin**

writeln(ot1[i].qu2);

**end**;

readln(n1);

writeln;

writeln(ot1[n1].ball2,' -старый балл');

writeln('Введите новый балл');

readln(ot1[n1].ball2);

**end**;

// -------------------------------------------------------------

// Процедура чтения теста из файла в массив записей.

//Входные параметры:

//- имя файла, откуда будет считан тест;

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте, на входе не определено);

//- пустой массив записей.

//Выходные параметры:

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** rf;

**var**

s,s1,s2,s3,s4,s5,s6:string;

ss,ss1,ss2,ss3,ss4:string;

i,n,n1,k,k1,k2,k3,k4,i1,i2:integer;

**begin**

ss:=',[';

ss1:=']';

ss2:=',';

ss3:='],';

ss4:=')';

assign(zapis,name);

reset(zapis);

n1:=0;

n:=1;

**while not** EOF(zapis) **do**

**begin**

ReadLn(zapis,s);

Inc(n1);

**end**;

close(zapis);

assign(zapis,name);

reset(zapis);

readln(zapis,s);

writeln(s);

setlength(ot1,n1);

**while not** eof(zapis) **do**

**begin**

readln(zapis,s);

k:=pos(ss,s);

k:=k-2;

s1:=copy(s,2,k);

ot1[n].qu2:=s1;

k:=k+5;

k1:=pos(ss1,s);

k1:=k1-k;

s2:=copy(s,k,k1); //massiv

s2:=s2+',';

s6:=s2;

i2:=0;

**while** s6<>'' **do**

**begin**

k2:=pos(ss2,s6);

delete(s6,1,k2);

i2:=i2+1;

**end**;

i2:=i2+1;

i:=1;

setlength(ot1[n].an2,i2);

**while** s2<>'' **do**

**begin**

k2:=pos(ss2,s2);

k2:=k2-1;

s3:=copy(s2,1,k2);

ot1[n].an2[i]:=s3;

k2:=k2+1;

delete(s2,1,k2);

i:=i+1;

**end**;

k3:=pos(ss3,s);

k3:=k3+2;

s4:=copy(s,k3,1);

val(s4,ot1[n].ball2,i1);

k3:=k3+1;

delete(s,1,k3);

k4:=pos(ss4,s);

k4:=k4-1;

s5:=copy(s,1,k4);

val(s5,ot1[n].goodan2,i1);

n:=n+1;

**end**;

n10:=n;

**end**;

// -------------------------------------------------------------

//Процедура записи теста в файл.

//Входные параметры:

//- имя файла, куда будет записан тест;

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//Выходные параметры:

//- файловая переменная, для связи процедуры с реальным файлом;

//- количество записей в массиве записей (количество вопросов в тесте);

//- массив записей, состоящих из вопросов, ответов, баллов, правильных ответов.

//

// Дата: 01.06.19

// -------------------------------------------------------------

**Procedure** wf1;

**var**

i,n:integer;

s:string;

**begin**

{связываем файловую переменную с файлом

открываем файл для записи

закрываем файл}

Assign(zapis,name);

Rewrite(zapis);

writeln('Введите название теста');

readln(s);

writeln(zapis,s);

n:=n10-1;

**for** i:=1 **to** n **do**

**begin**

writeln(zapis,ot1[i]);

**end**;

Close(zapis);

**end**;

**end**.