**Содержание**

[1. Исследование предметной области 4](#_Toc59996880)

[1.1 Описание программы 4](#_Toc59996881)

[1.2 Основные функции и подфункции 4](#_Toc59996882)

[1.3 Распределение ролей в команде 6](#_Toc59996883)

[1.3.1 Синицын Иван Сергеевич 6](#_Toc59996884)

[1.3.2 Карпенко Александр Павлович 6](#_Toc59996885)

[1.3.3 Смирнов Михаил Александрович 6](#_Toc59996886)

[1.3.4 Джагаев Вячеслав Андреевич 6](#_Toc59996887)

[1.4 Макет программы 7](#_Toc59996888)

[2. Использование система контроля версий 10](#_Toc59996889)

[3. Тестирование информационной системы 14](#_Toc59996890)

[3.1 Тестируемые функции 14](#_Toc59996891)

[3.2 Наборы тестов для тестируемых функций 14](#_Toc59996892)

[3.3 Пример фрагмента кода с использованием Юнит-тестов 15](#_Toc59996893)

[3.4 Пример сценария тестирования 17](#_Toc59996894)

[4. Кодирование 18](#_Toc59996895)

[4.1 Правила форматирования кода 18](#_Toc59996896)

[4.2 Оценка цикломатической сложности 19](#_Toc59996897)

[4.3 метрика Холстеда 19](#_Toc59996898)

[5. Результат работы программы 20](#_Toc59996899)

# **1. Исследование предметной области**

***Цель работы*** – необходимо проанализировать предметную область, а именно описание программы, состав участников, определения функций и подфункций, макет программы.

## **1.1 Описание программы**

Контролирующее – обучающая программа предназначена для внутреннего тестирования студентов «Название учебного заведения». Графическое приложение содержит тесты с разными нагрузками и вариантами. Для помощи в прохождении тестирования студентам предоставлен теоретический материал с формулами и примерами. Так же для понимания пользования приложения есть «справка», в которой предоставлена информация, как правильно пользоваться программой. После завершения тестирования результат студента будет сохранен в базе данных. Результаты всех студентов можно посмотреть в разделе «Результаты тестирований» в главном меню приложения. Данная программа является «ремейком» старой программы, которую использовали во времена *windows* XP. Новая версия программы будет запускаться на современных операционных системах (Windows 7/10).

## **1.2 Основные функции и подфункции**

* *Вход в систему* – для начала тестирования приложение предоставляет студенту возможность ввести свои пользовательские данные (ФИО, группа), после чего сохранить их для последующего вывода в базу данных результатов.
* *Выбор теста* – для начала тестирования приложение дает возможность студенту выбрать тему теста, тип чертежа положение нагрузок, вариант.
  + *Выбор времени* – подфункция, позволяющее студенту выбрать время тестирования (диапазон (в минутах): 10 - 60). Время тестирования решает преподаватель
  + *Выбор графической подсказки* – подфункция, позволяющее студенту выбрать, после какой неудачной попытки ответа на вопрос будет появляться подсказка на чертеже для помощи (диапазон: 0-3). Значение графической подсказки решает преподаватель.
* *Прохождение теста* – непосредственно само тестирование, которое должен пройти студент в отведенное для этого время. **За каждый правильный ответ на вопрос дается 5 баллов.**
  + *Проверка на синтаксические ошибки* – подфункция, которая проверяет введенных ответ студента на синтаксические ошибки. Если же программа их находит, то она информирует студента об ошибке, выводя соответствующее сообщение. **Баллы** **за синтаксические ошибки** **не снимаются**.
  + *Право на ошибку* – подфункция, дающее возможность ответить на вопрос еще раз, если студент неправильно дал ответ на вопрос. На каждый вопрос дается ровно 3 попытки. За каждый неправильный ответ **снимается один балл.**
  + *Сохранение результатов* – после завершения тестирования окончательный результат с оценкой сохраняется в базе данных студентов, с которыми в дальнейшим может ознакомиться преподаватель
* *Просмотр результатов студентов* – программа дает возможность преподавателю посмотреть список результатов всех студентов, проходивших тестирование.
  + *Сортировка результатов* – подфункция, которая дает возможность отсортировать список результатов студентов по 4-ем критериям: *группе, фамилии, имени, отчеству*.
  + *Поиск* – подфункция, которая дает возможность найти необходимый результат студента в списке результатов с помощью **поисковой строки**. Настроить поисковую строку можно по 5-и критериям: *всему, фамилии, имени, отчеству, группе*.
* *Просмотр теоретического материала и справки* – программа для помощи студентам при прохождении теста или пользования приложения предоставляет возможность просмотра теоретического материала и справки соответственно. Открыть их можно в любое время.

## **1.3 Распределение ролей в команде**

### 1.3.1 Синицын Иван Сергеевич

Студент ИСТ 1-19

**Тим-Лидер**, разработчик функционала программы (обработка ответа студента, использование исходных данных теста, проверка на синтаксические ошибки и т.д.).

### 1.3.2 Карпенко Александр Павлович

Студент ИСТ 1-19

*Программист*, разработчик графического интерфейса, разработчик функционала программы, UI/UX дизайнер (создания макета, методов, связок графического интерфейса).

### 1.3.3 Смирнов Михаил Александрович

Студент ИСТ 1-19

*Программист*, разработчик основного функционала программы (Создание классов и методов при работе тестирования, выгрузка из файлов, работа с репозиторием и слиянием веток, исправление ошибок программы).

### 1.3.4 Джагаев Вячеслав Андреевич

Студент ИСТ 1-19

*Программист*, разработчик базы данных (создание структуру базы данных результатов и тестов, создание класса обработки файлов базы данных)

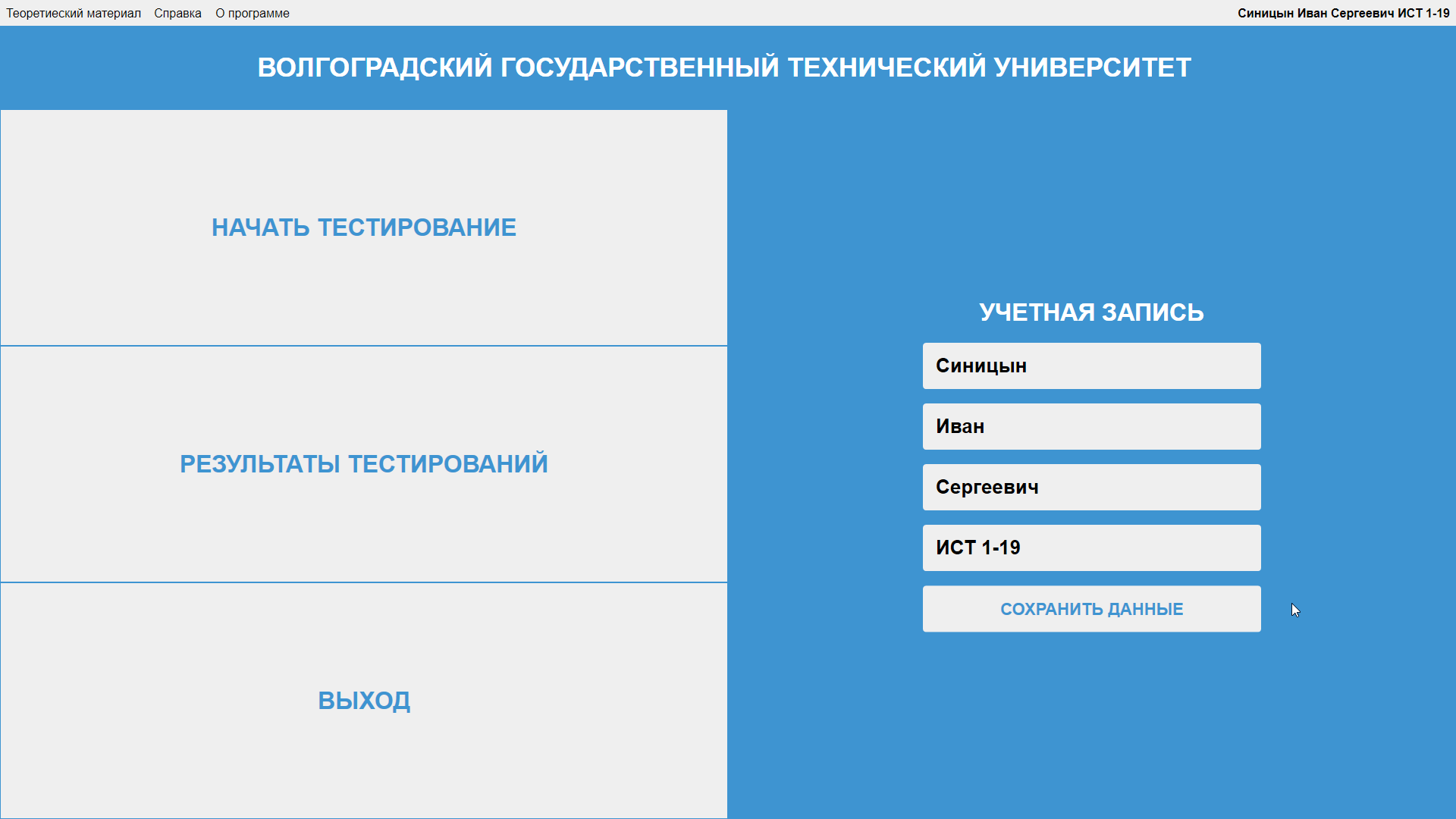
**1.4 Макет программы**

Ниже будет представлен макет «Контролируемо-обучающая программа».

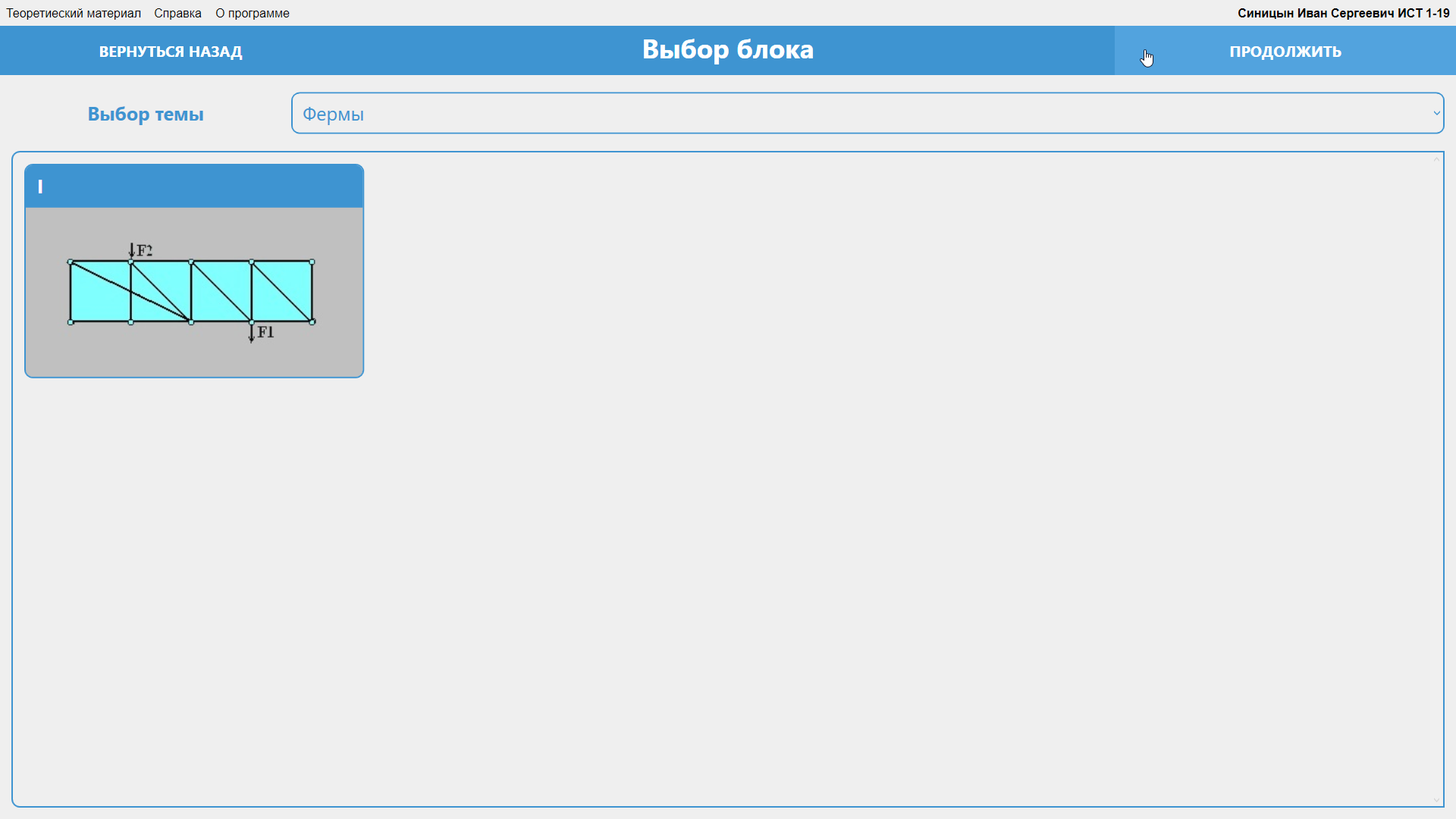
1. Макет Главного меню

Вверху слева идет плашка, которая содержит в себе «Теоретический материал», «Справка» по пользованию программой и общая информация «О программе». Это сделано для того, чтобы они находились в одном и том же месте во время работы программы.

Вверху справа находятся *пользовательские данные* студента, которые он вводит в *учетной записи –* это сделано для того, чтобы дать понять студенту, что его данные были сохранены.

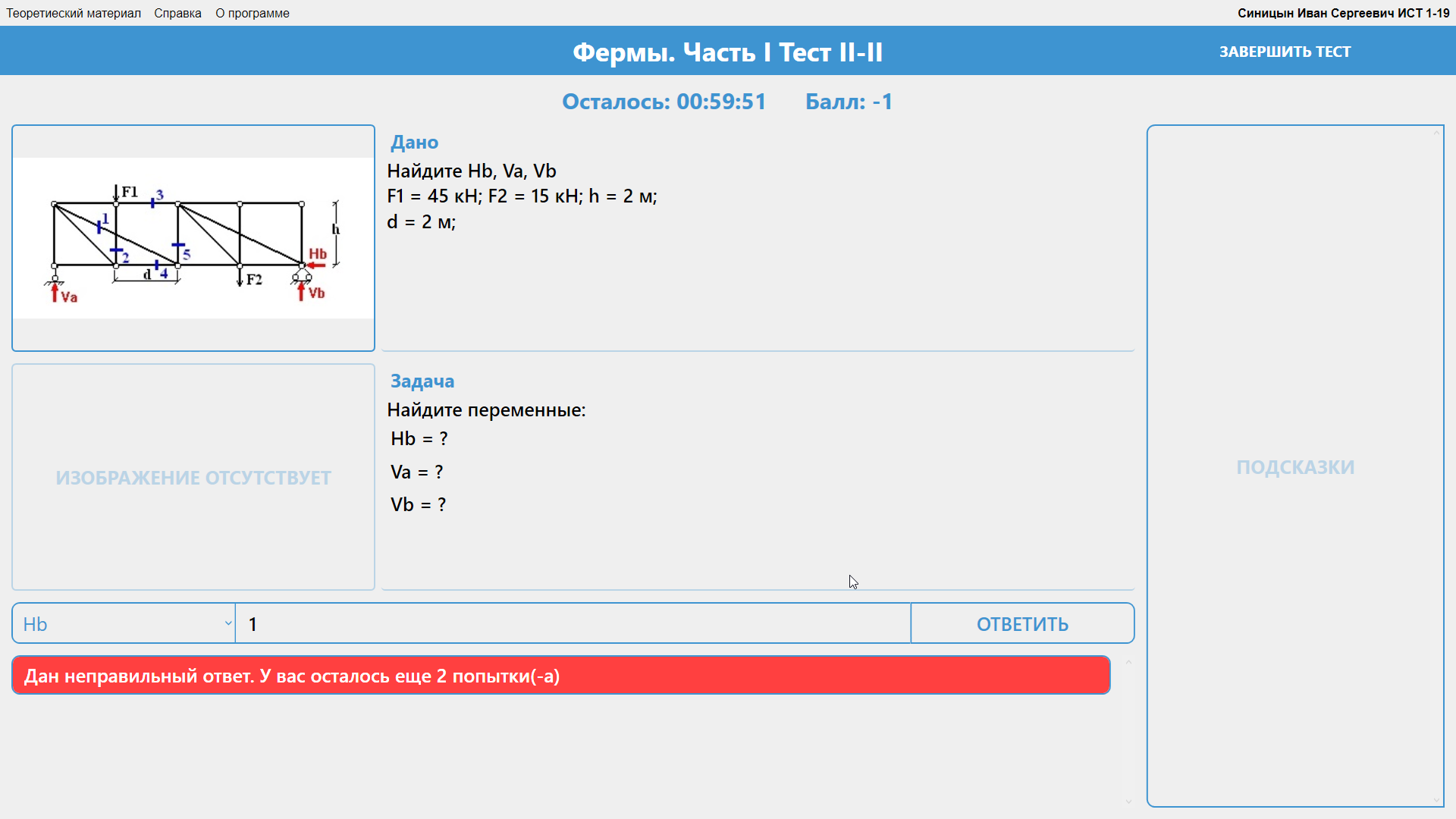


2. Макеты программы после нажатия на кнопку «Начать тестирование»

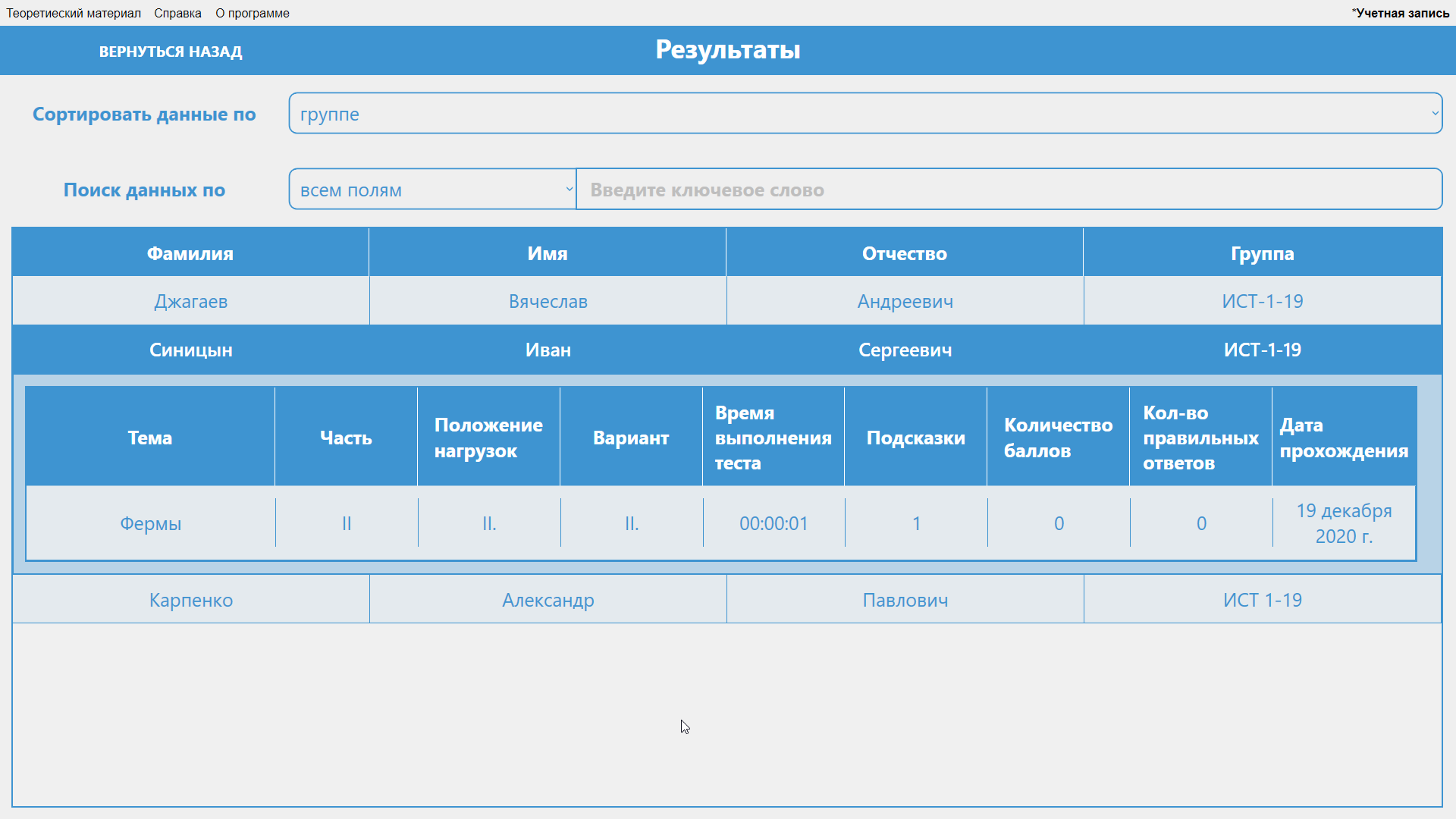




3. Макет тестирования



4. Макет списка результатов



Вывод: предложенный нами интерфейс программы удовлетворяет фактору качества разработки ИС, как удобство пользования потому что он выполнен в минималистическом стиле и поэтому такой вид интерфейса не будет нагружать студента лишней информацией и «простой вид» интерфейса позволяет легче его понять.

# **2. Использование система контроля версий**

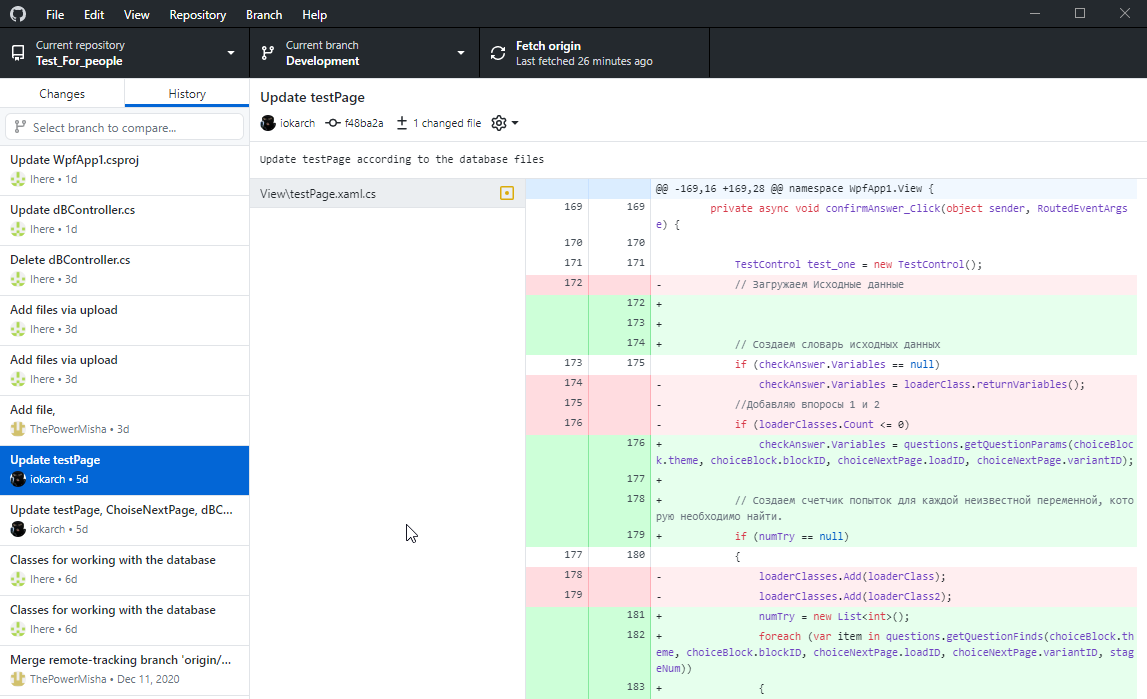
***Цель работы*** – работа и использование системы контроля версий, привести пример коммита, дерево ведения и ветки проекта.



Любая программа должна реализовываться с использованием распределенной системы контроля версий и наша не исключение. Для нашего приложения мы используем бесплатный и крупнейший веб-сервис GitHub (<https://github.com>).

Логотип GitHub

1. Пример коммита программы:



Коммит

Какие файлы были изменены

Что было изменено

Репозиторий

Ветка

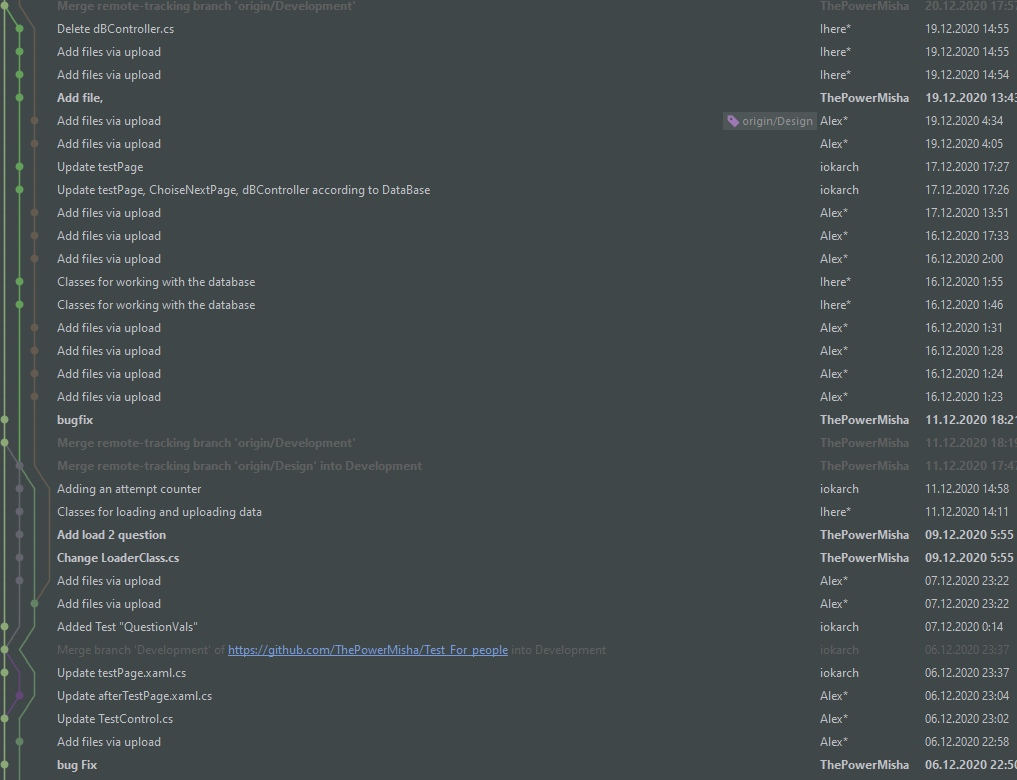
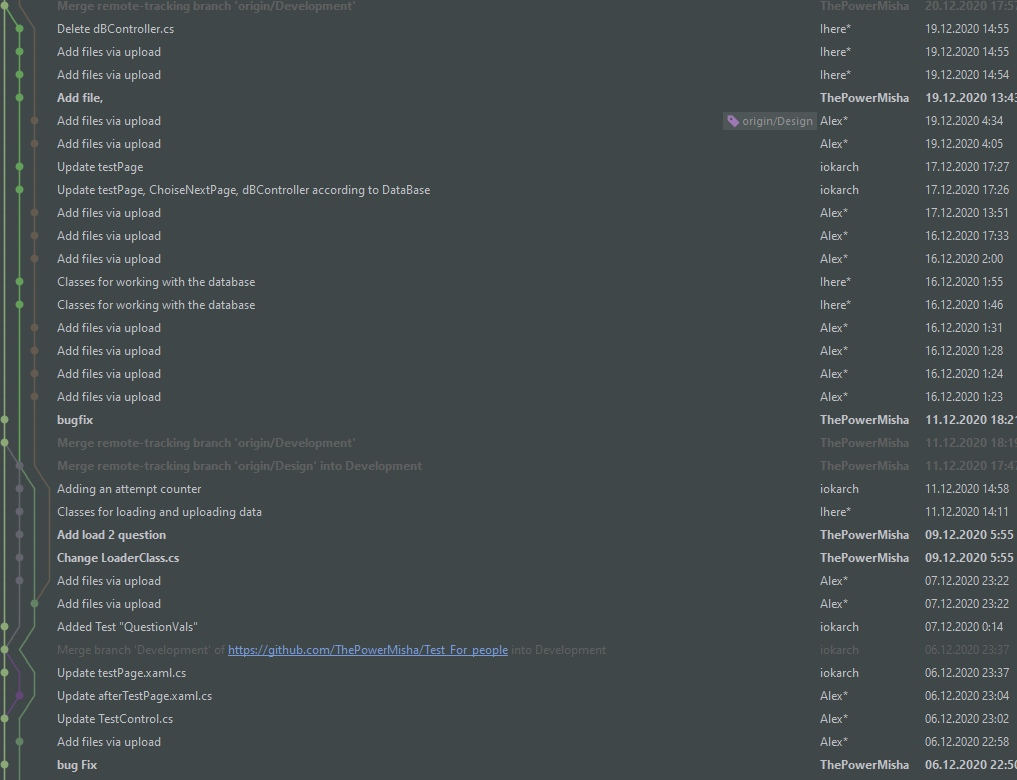
Название Коммита

Описание Коммита

8ш

Ниже представлено дерево веток проекта

* Ветка master – основная ветка, в которую сливаются ветки
* Ветка Development – ветка, в которой реализуются основные функции программы
* Ветка Design – ветка, в которой реализуется графический интерфейс



master

Development

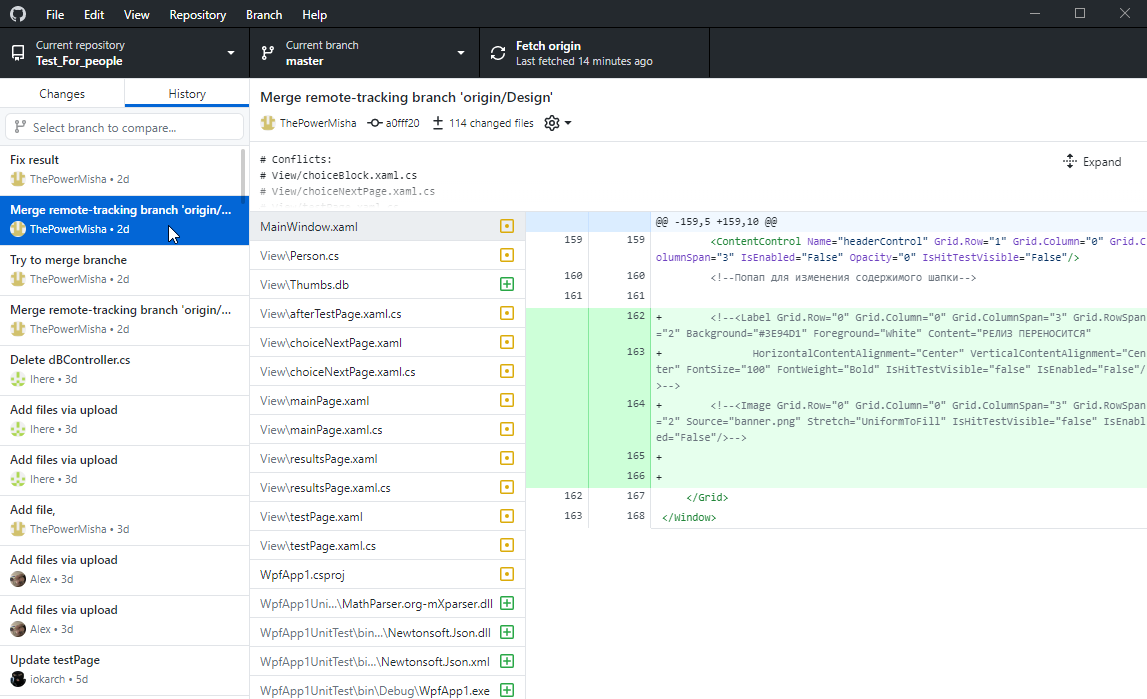
Design

Название коммитов

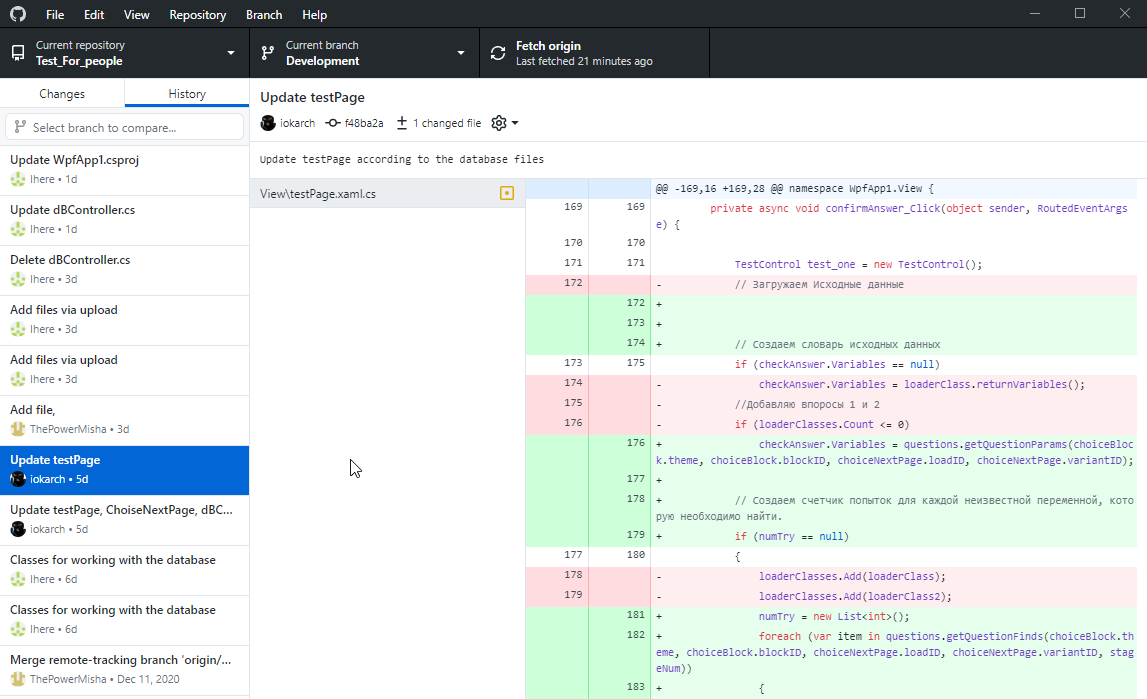
Автор коммита

Дата коммита

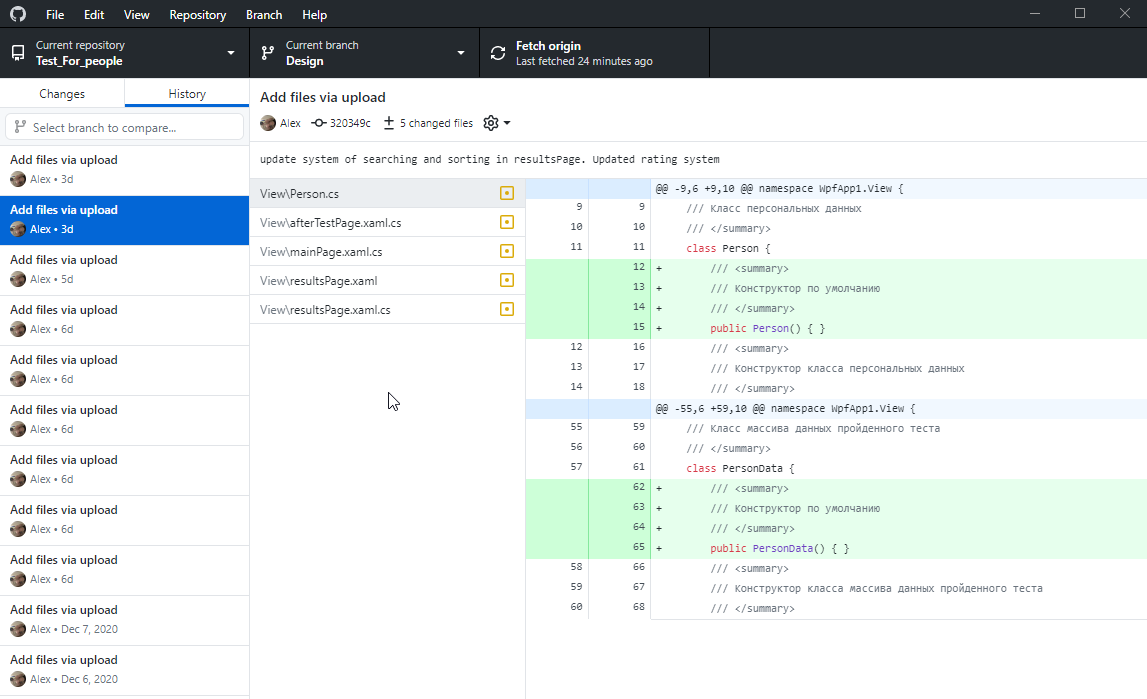
3. Пример веток

 3.1 Ветка ***master –*** основная ветка, в которую сливаются ветки

Ветка master

 3.2 Ветка Development – ветка, в которой реализуются основные функции программы

Ветка Development

 3.3 Ветка Design – ветка, в которой реализуется графический интерфейс

Ветка Design

# **3. Тестирование информационной системы**

***Цель работы*** – Протестировать программный код с помощью технологий автоматизированного и ручного тестирования.

## **3.1 Тестируемые функции**

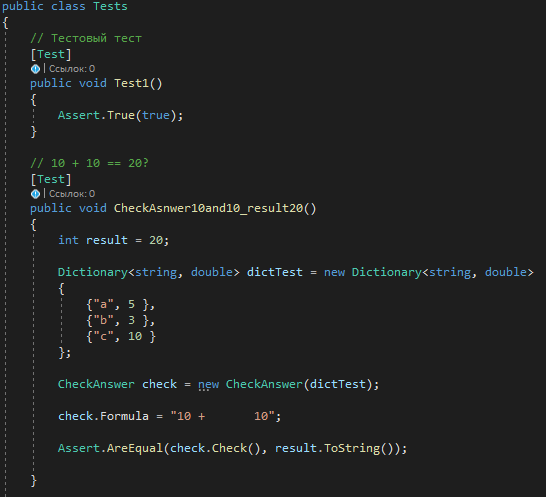
* CheckAnswer – check
* CheckAnswer – SyntaxСheck
* CheckAnswer – BracketCheck
* dBController – getQuestionText
* dBController – getQuestionFinds
* dBController – getMainVariantImgPath

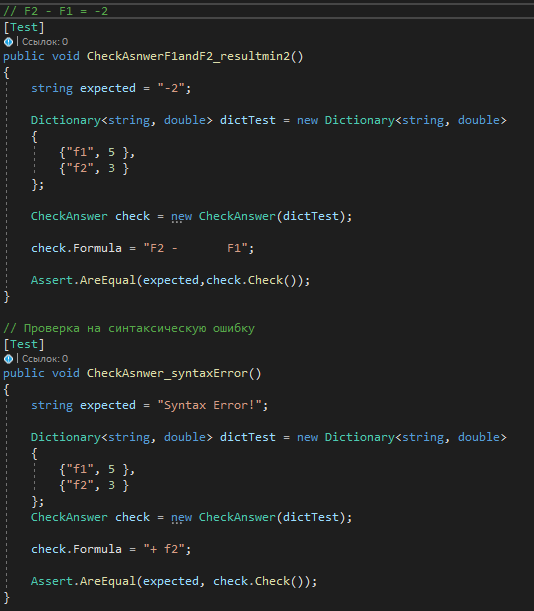
## **3.2 Наборы тестов для тестируемых функций**

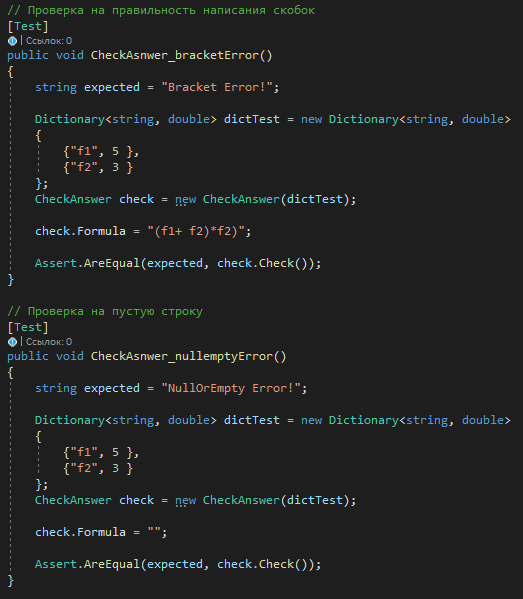
* check – Проверка на расчет введенной функции
* check – Проверка на пустую строку
* SyntaxСheck – Проверка на синтиаксическую ошибку
* BracketCheck – Проверка на правильность написания скобок
* getQuestionText – Получить текст из БД
* getQuestionFinds – Получить список списков переменных
* getMainVariantImgPath – Получить список картинок для вопросов

## **3.3 Пример фрагмента кода с использованием Юнит-тестов**

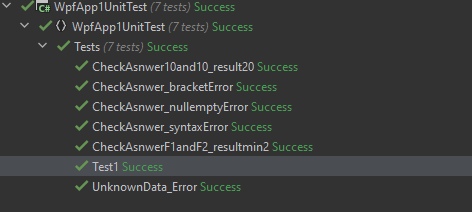
Для Юнит-теста мы взяли проверку функцию check класса CheckAnswer





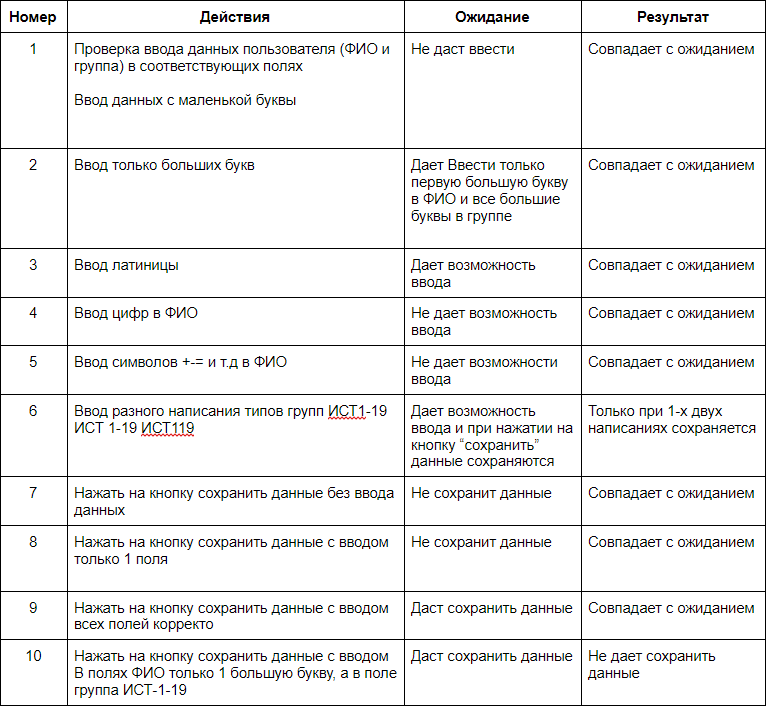


Результат работы Юнит-теста:



## **3.4 Пример сценария тестирования**

Для сценария тестирования мы взяли *ввод пользовательских данных* в главном меню приложения.



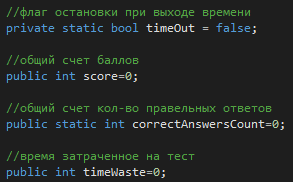
# **4. Кодирование**

***Цель работы*** – соблюдать кодирование согласно правилам форматирования кода. Выполнить оценку цикломатической сложности и использование метрики Холстеда.

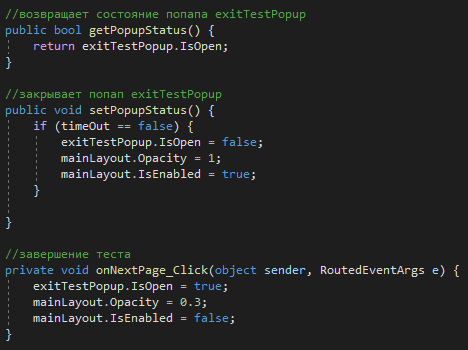
## **4.1 Правила форматирования кода**

* Нотация кода – camelCase.
* Фигурные скобки на той-же строке что и функция
* Табуляция в 4 пробела
* Между разными функциями интервал 1 строка
* Стиль написания комментариев – над функциями, объединение общих элементов
* Краткое описание всех открытых функциях и входящих в них переменных через *///summary*

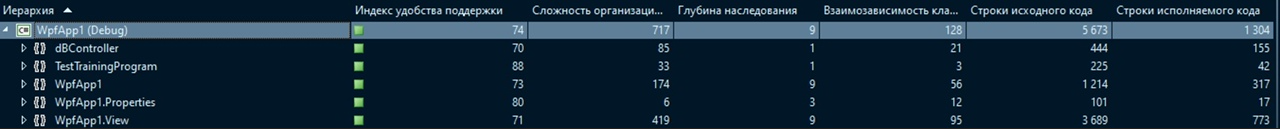
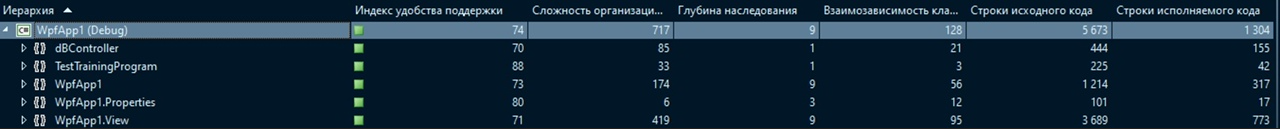
Прим. 1



Прим. 2



## **4.2 Оценка цикломатической сложности**

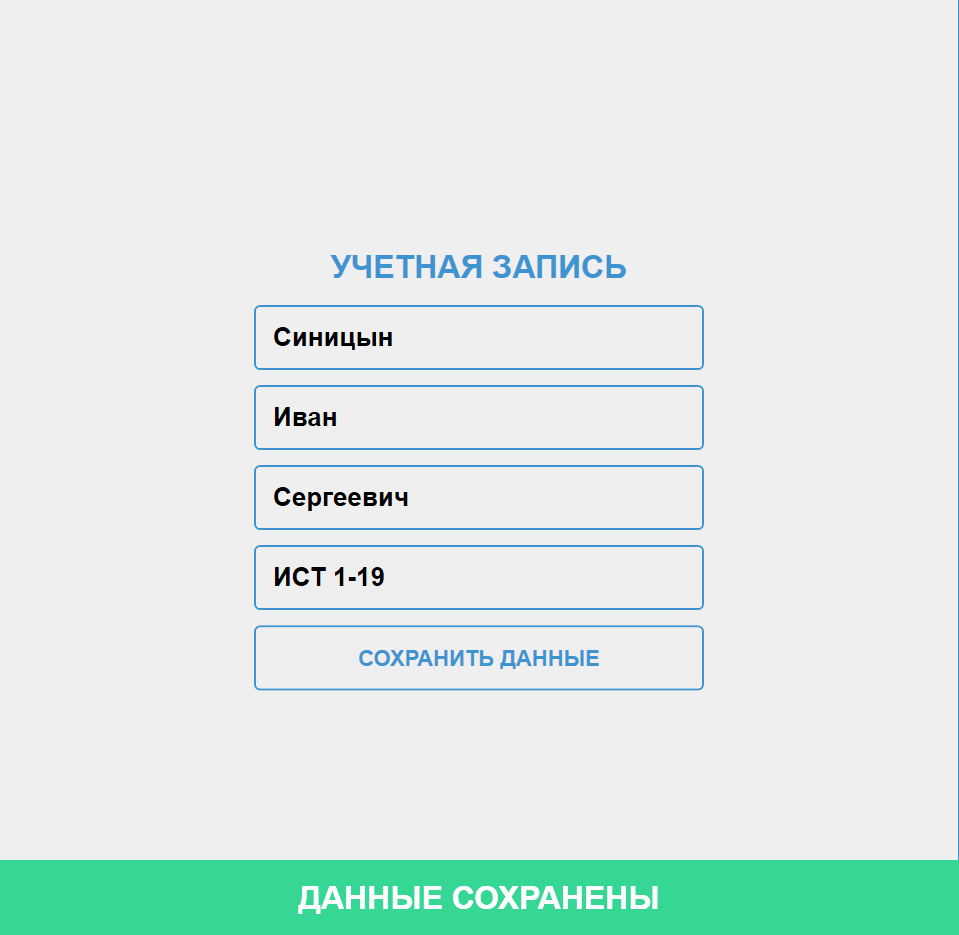
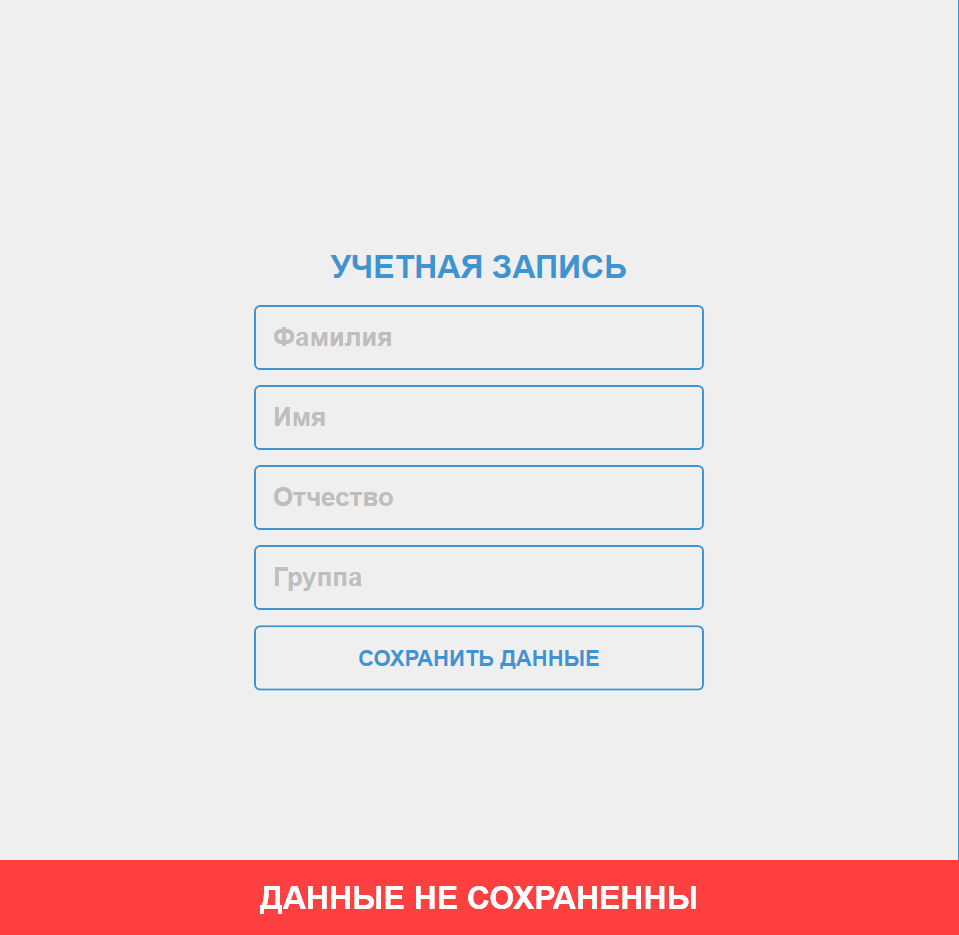
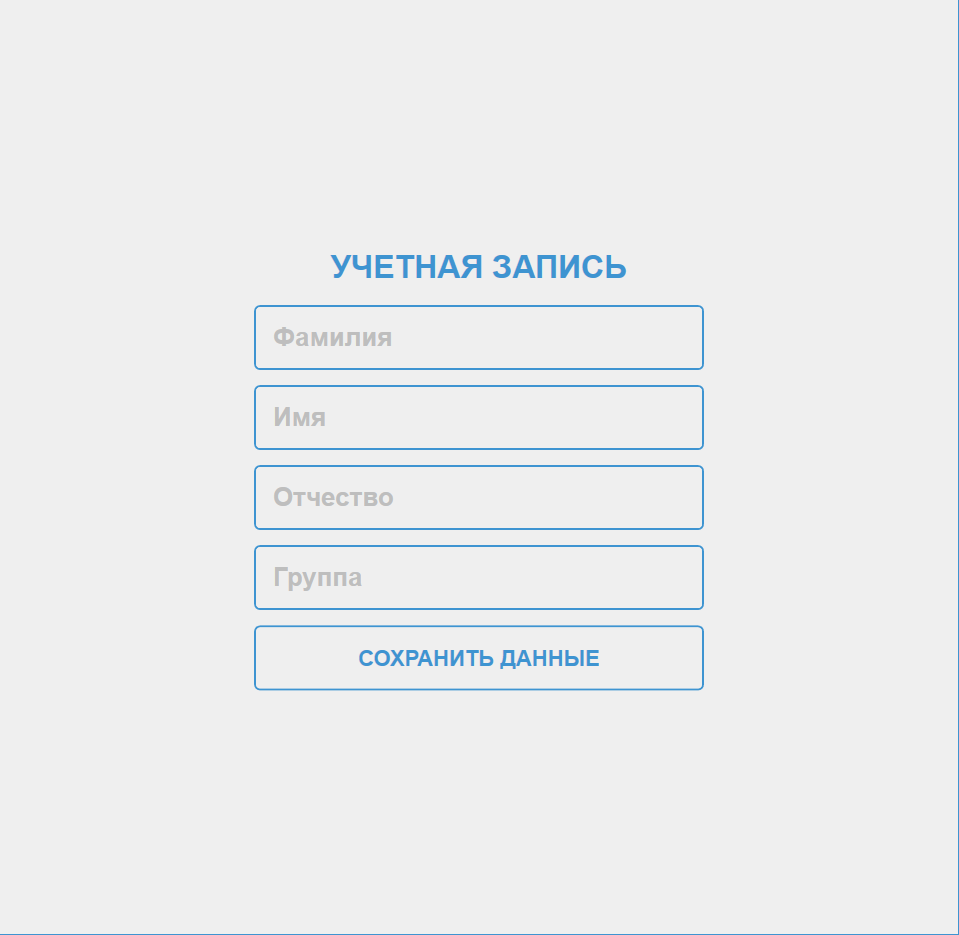


## **4.3 метрика Холстеда**

# **5. Результат работы программы**

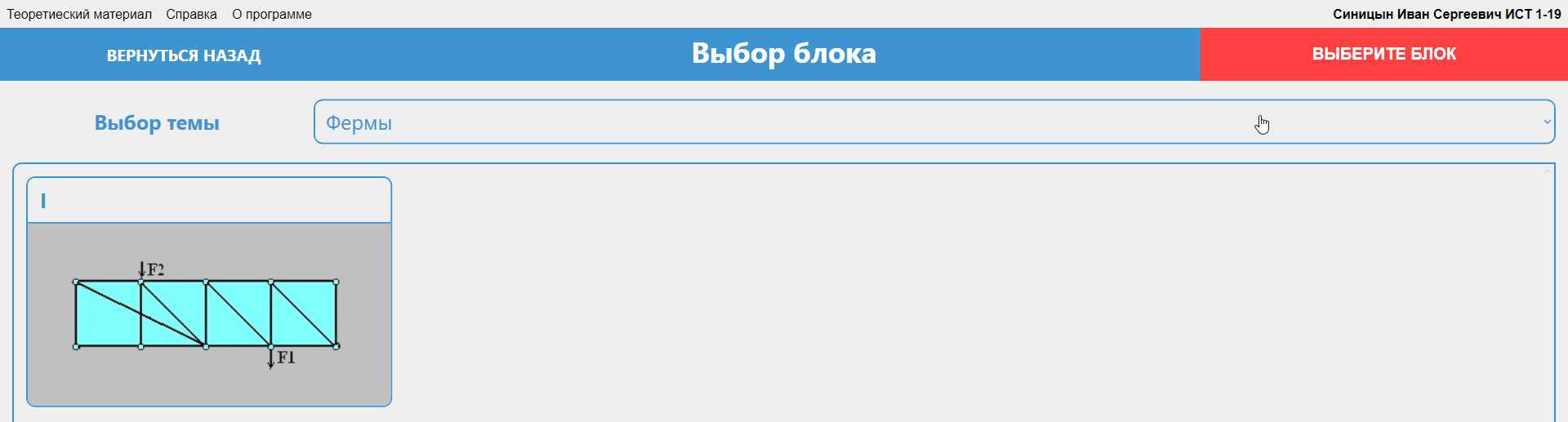
***Цель работы*** – показать работоспособность программы в целом или отдельных функций.

5.1. Пример работы «Пользовательский данных»

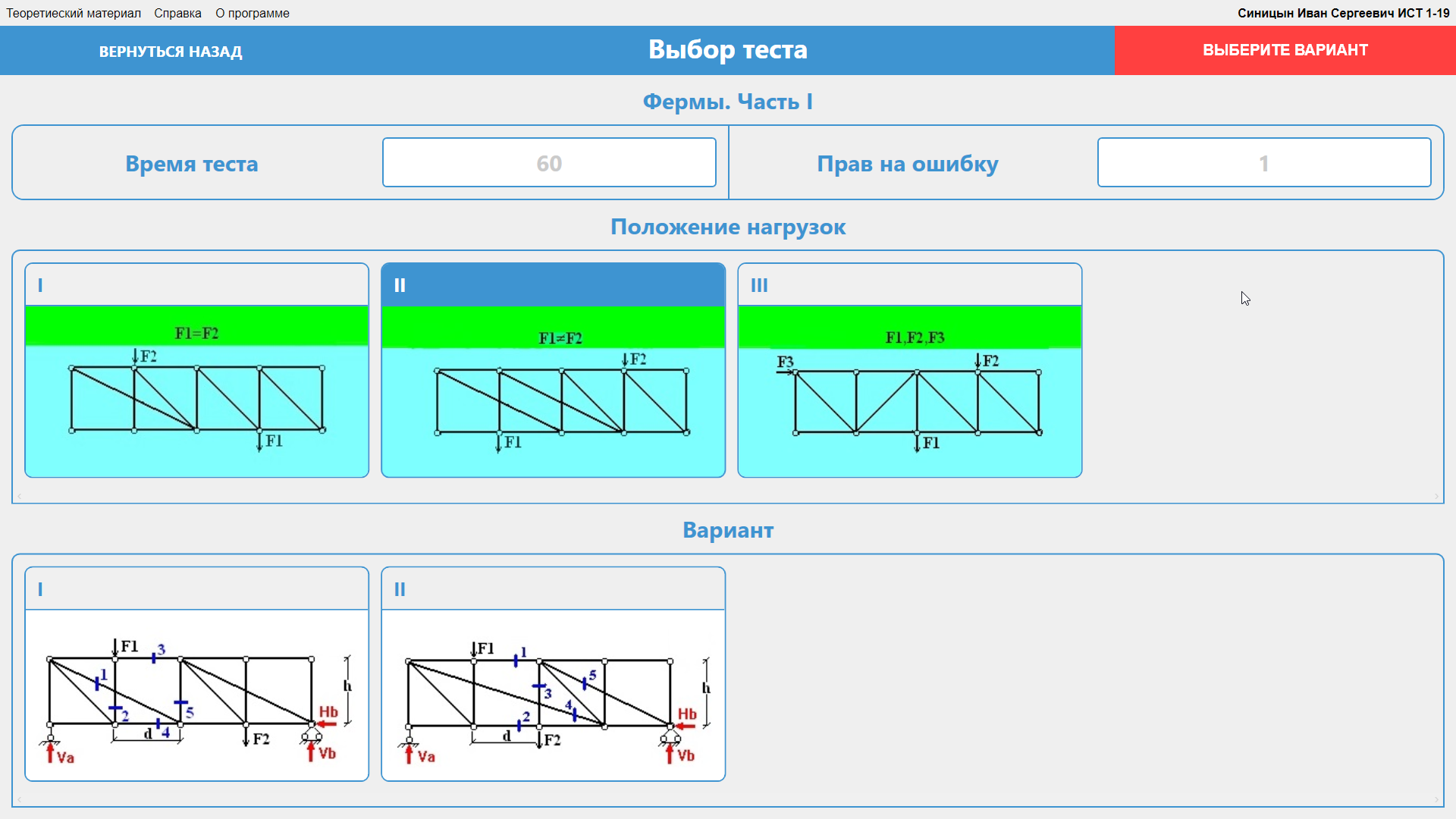


Не вводим данные Данные не сохранены Данные сохранены

5.2 Пример работы выбора части/положения/варианта

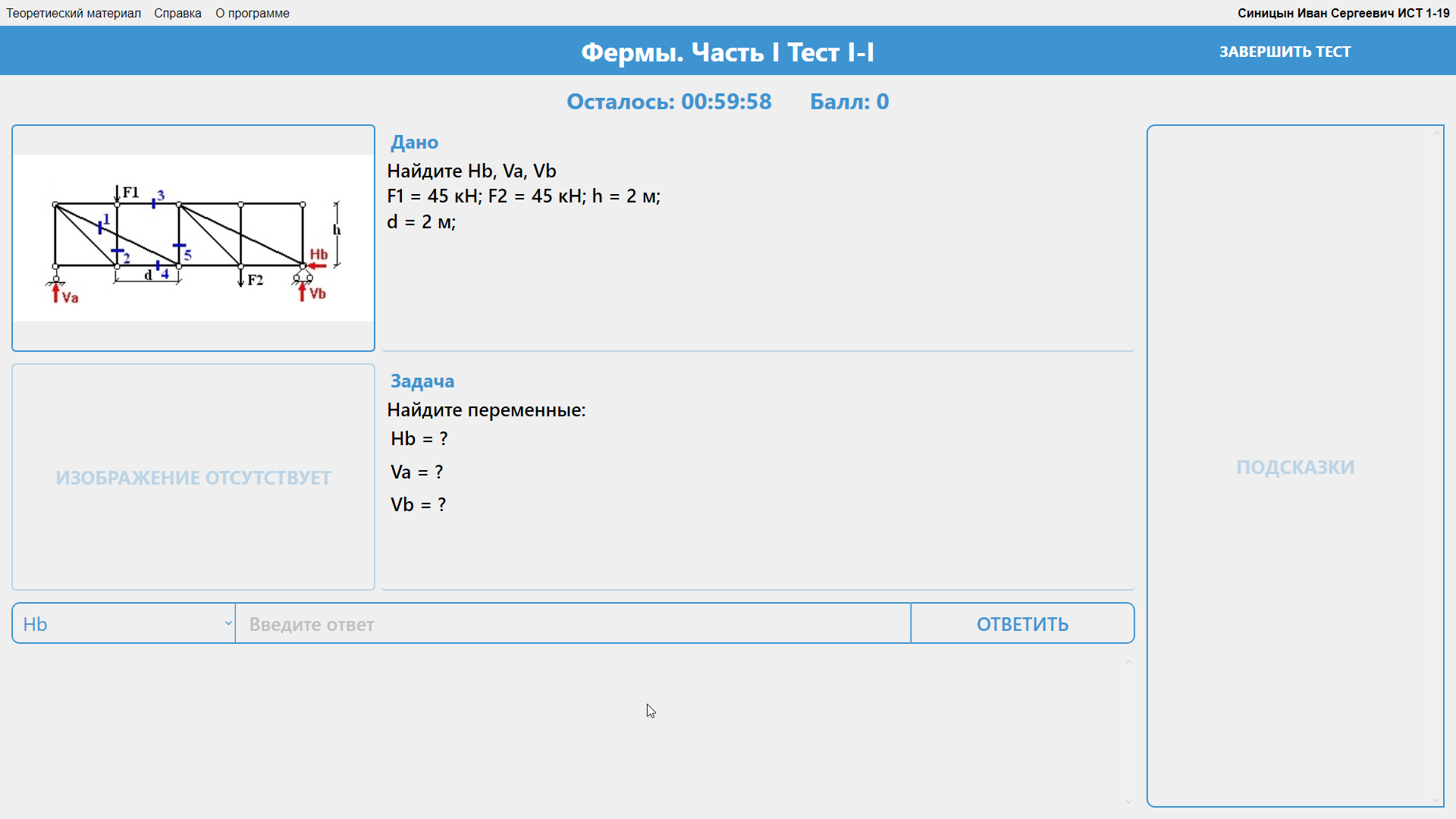


Если блок не будет выбран, то выведется ошибка при нажатии на кнопку «Продолжить»

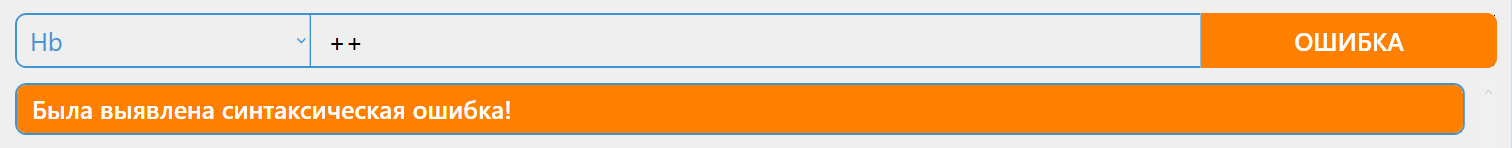


То же самое работает с положением нагрузок и вариантами

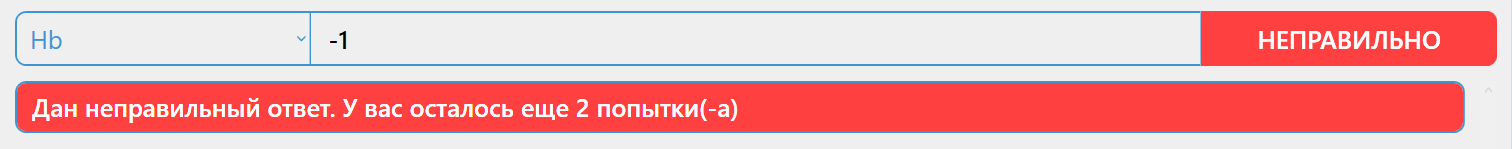
5.3 Пример тестирования



Окно тестирования



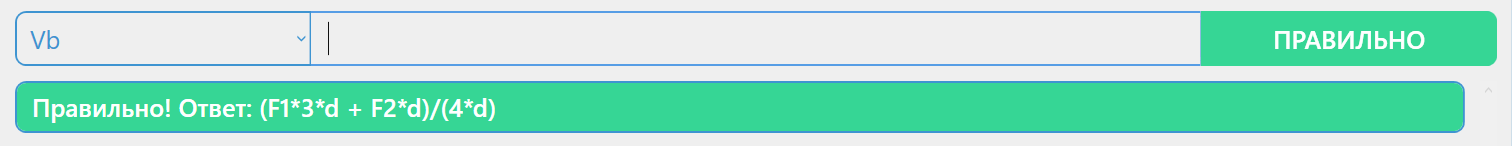
Если допущена синтаксическая ошибка, то будет выведено сообщение (баллы не снимаются)



Если был введен неправильный ответ, то дается еще n-попыток на ответ (за каждый неправильный ответ баллы снижаются на 1)

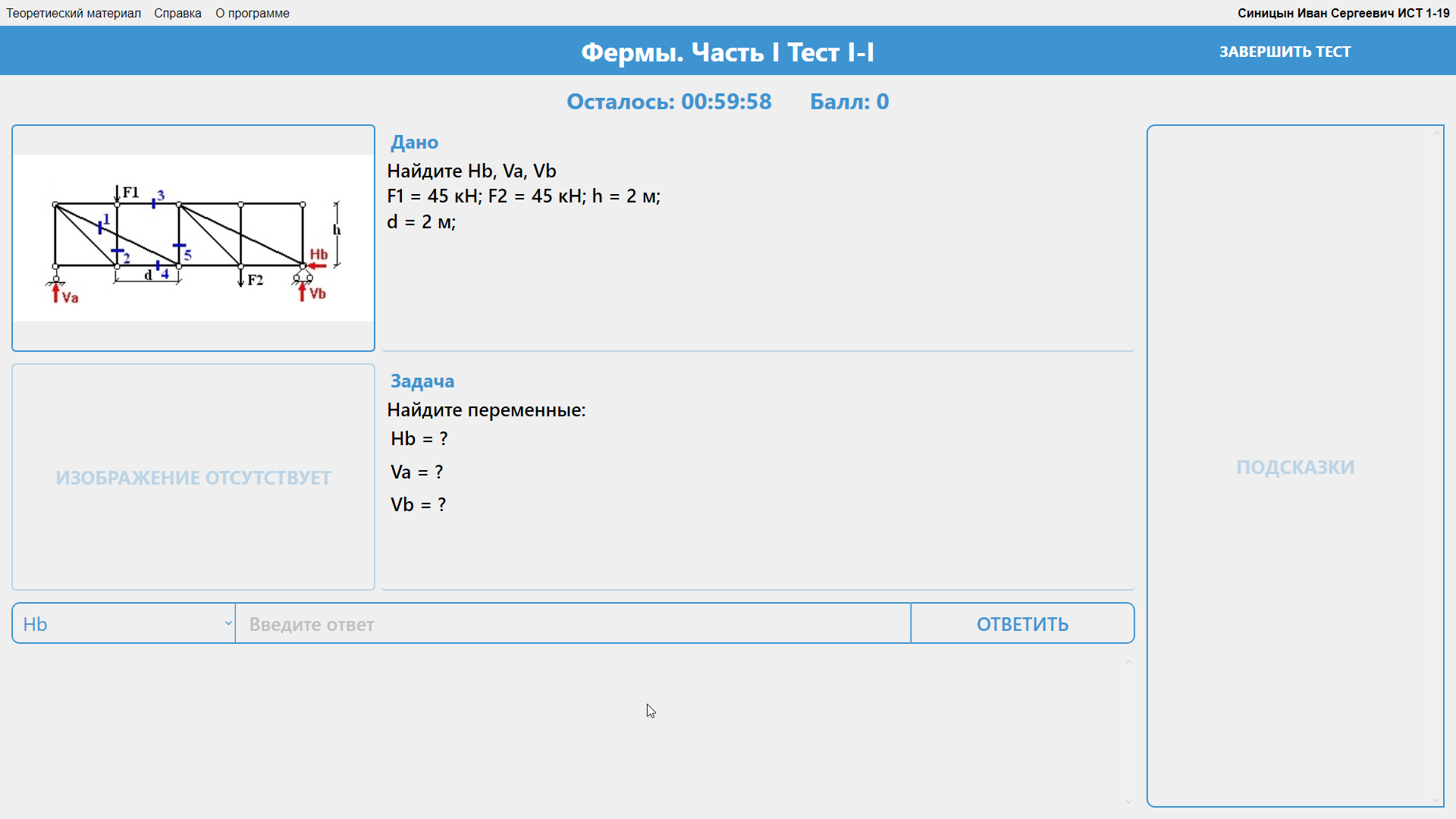


Если попыток больше нет

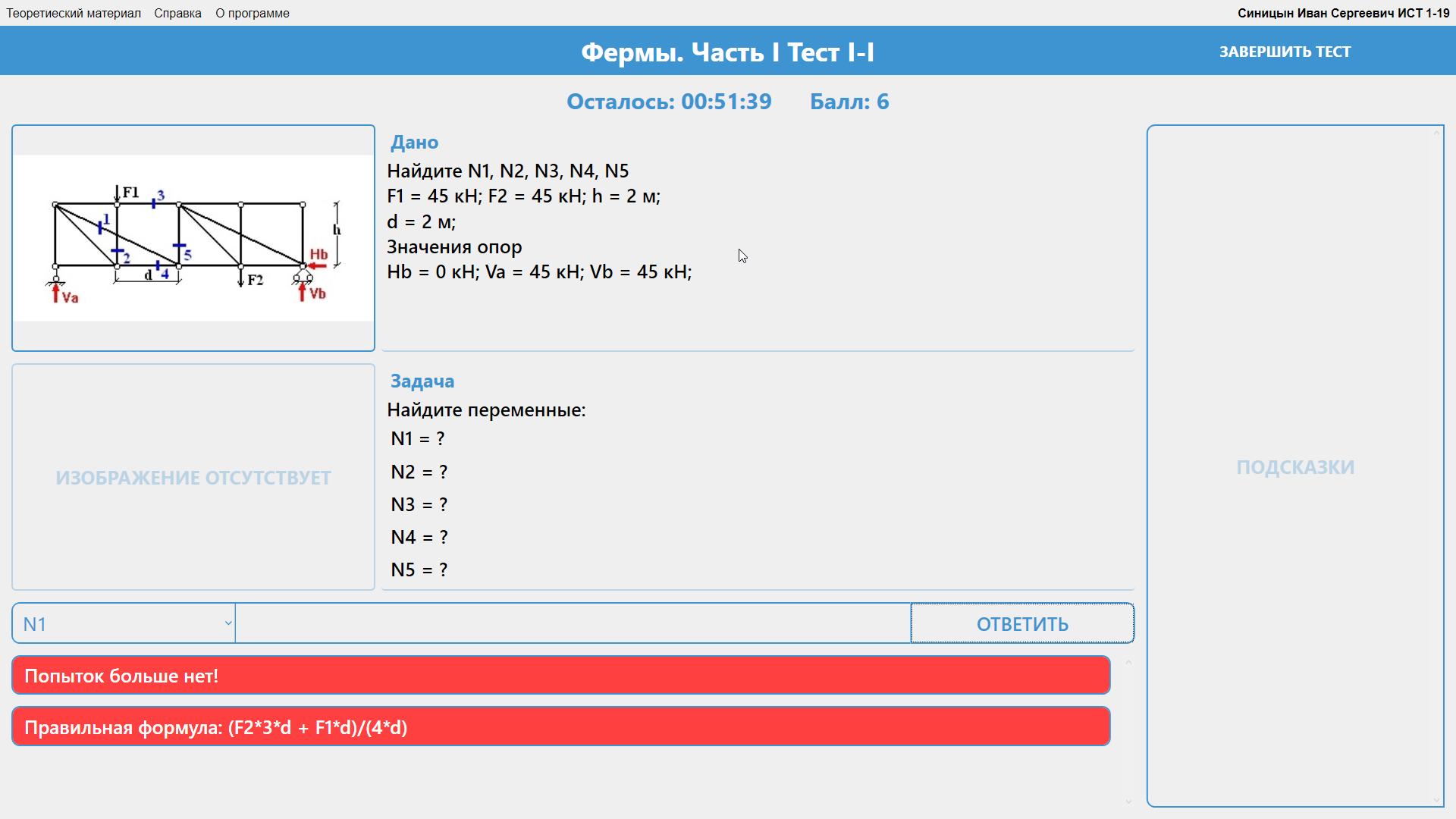


За правильный ответ дается 5 баллов

5.4 Пример смены вопросов

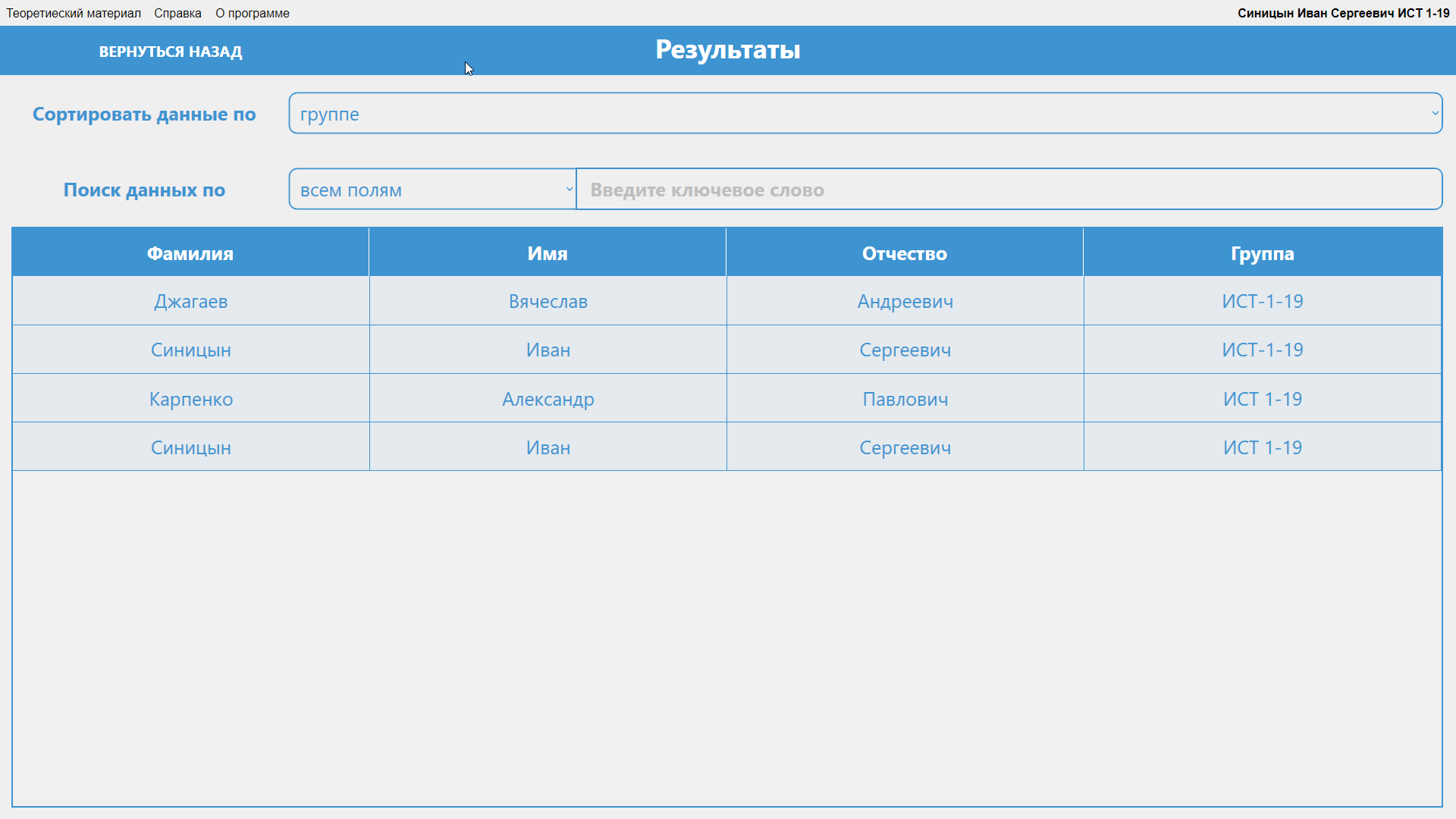


1-ый этап вопросов (нахождение опор)

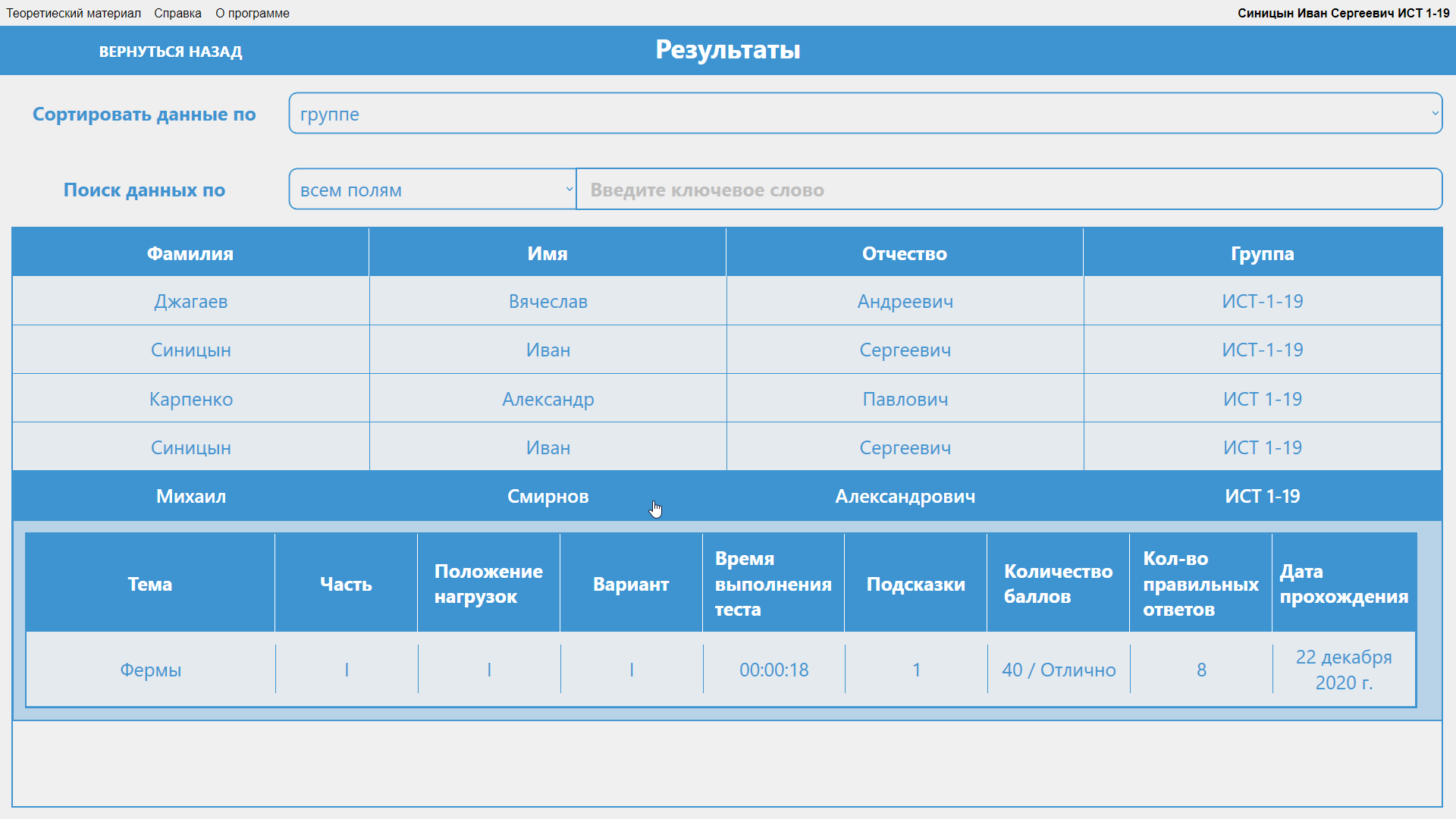


2-ой этап вопросов (нахождение N1,N2,N3,N4,N5)

5.5 Пример работы добавление результатов в список результатов



Страница результатов до добавления нового студента



Страница результатов после добавления нового студента (Михаил Смирнов Александрович ИСТ 1-19)

Как выглядит БД страницы результатов