**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**

***Угадай число*** – логическая головоломка на скорость, которая предназначена для развития логического мышления и умения анализировать ситуацию у студентов Волгоградского Технического Колледжа. Графическое приложение содержит интерфейс угадывания числа с подсказками. После завершения результат студента будет сохранен в базе данных. Результаты всех студентов можно посмотреть при запуске приложения.

## **1.2 Основные функции и подфункции**

* *Ввод имени* – для начала игры приложение предоставляет студенту возможность ввести своё имя, которое сохраняется в базе данных.
* *Начало игры* – после ввода имени, у студента появляется возможность начать игру по нажатию кнопки. Активация игры запускает таймер и основной функционал.
* *Нахождение* – непосредственно сама игра, которую должен пройти студент менее чем за одну минуту. **За каждое введенное число студент получает подсказку от программы.**
* *Просмотр рекордов студентов* – программа дает возможность преподавателю посмотреть список результатов всех студентов, проходивших игру.
* *Запись рекорда* – после нахождения числа, студент имеет возможность записать свой рекорд в базу данных.

## **1.3 Распределение ролей в команде**

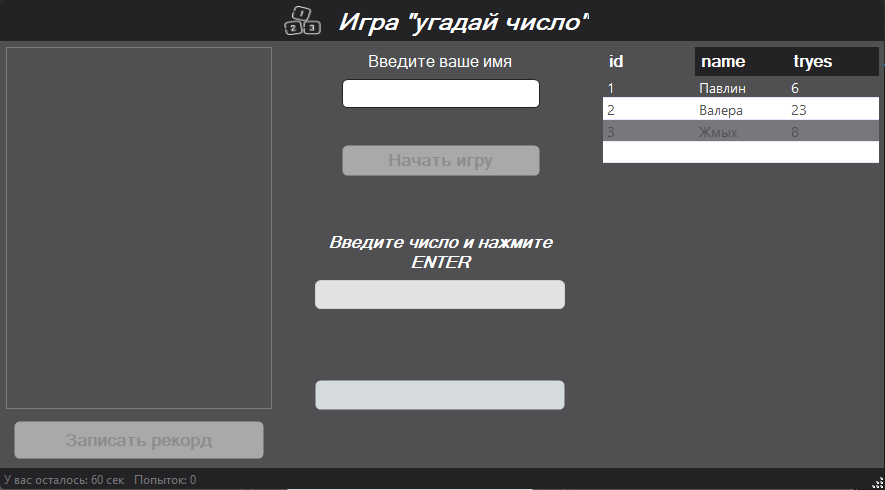
### Проект был разработан одним человеком – студентом группы ПКС-1-19, Шеховцовым Глебом Евгеньевичем.

В обязанности входили: разработка функционала программы, разработка графического интерфейса, работа с репозиторием github, создание базы данных.

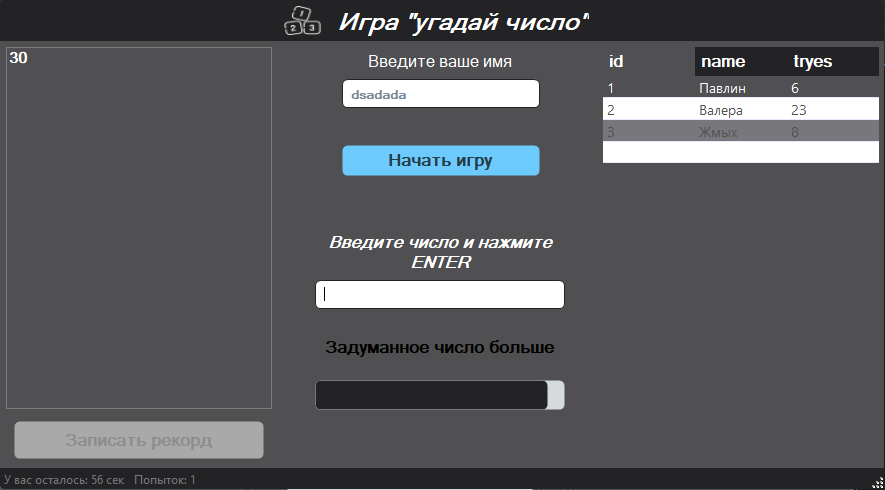
**1.4 Макет программы**

Ниже будет представлен макет программы «Угадай число».

1. Макет Главного меню



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



3. Макет тестирования

4. Макет списка результатов

Вывод: предложенный нами интерфейс программы удовлетворяет фактору качества разработки игрового приложения, 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**

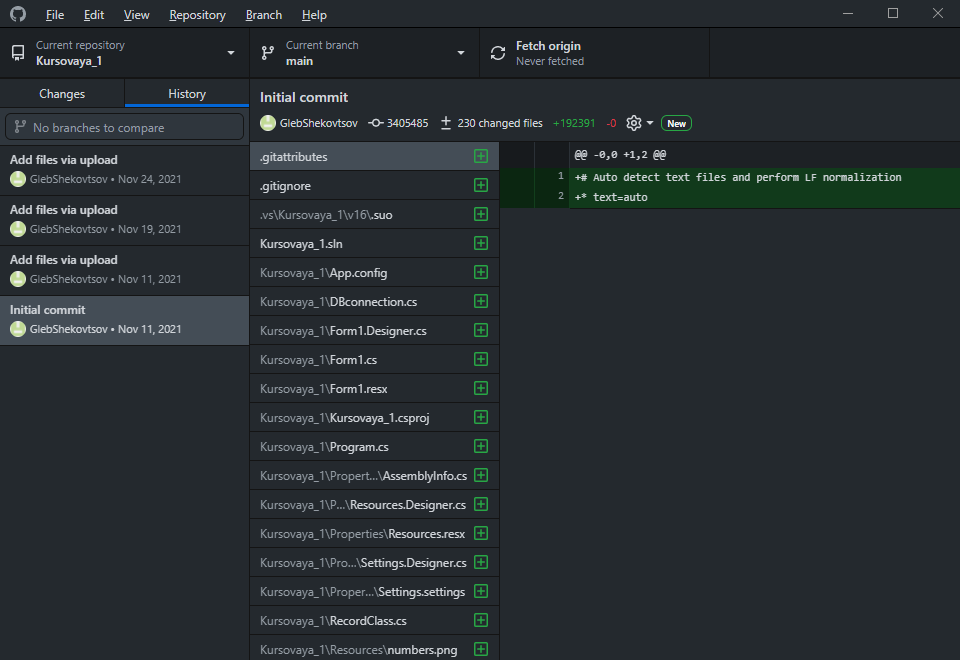
***Цель работы*** – работа и использование системы контроля версий, привести пример коммита, дерево ведения и ветки проекта.



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

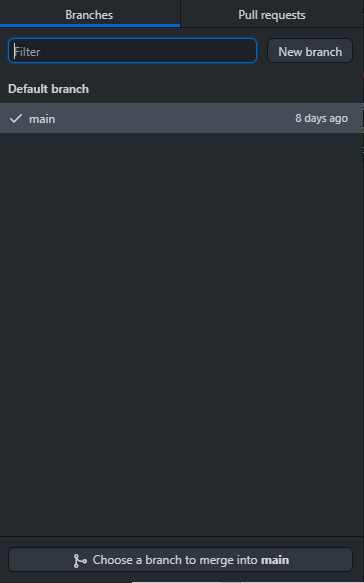
Ветка

Название Коммита

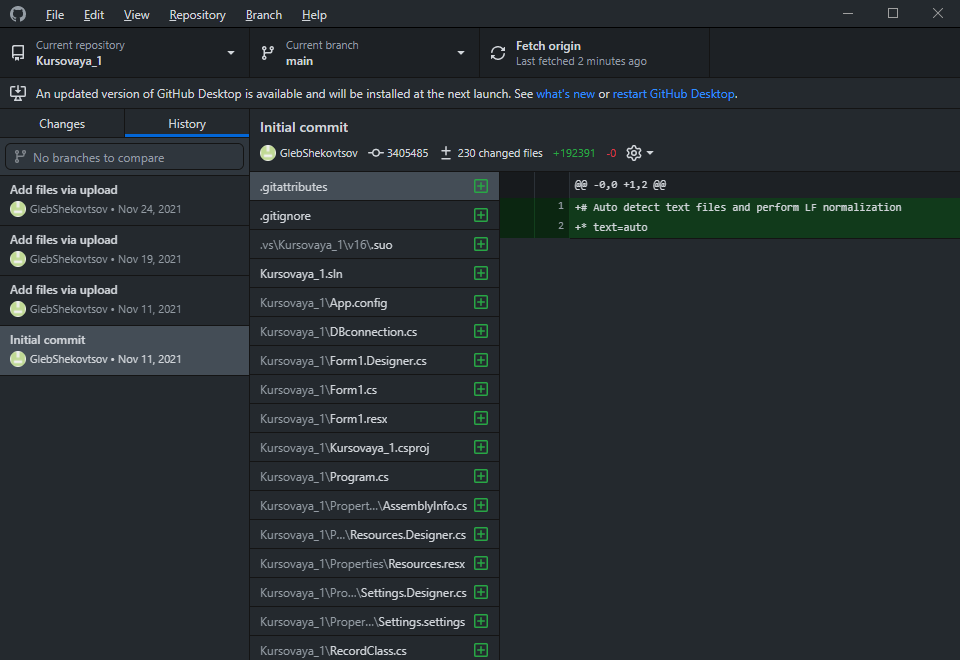
Описание Коммита

Ниже представлены ветки проекта

* Ветка main – основная ветка



2. Пример веток

2.1 Ветка ***main –*** основная ветка

Ветка Development

# **3. Тестирование информационной системы**

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

## **3.1 Тестируемые функции**

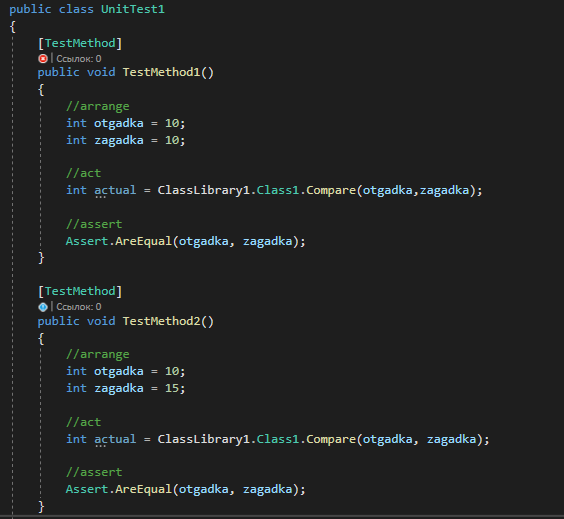
* Compare check – check

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

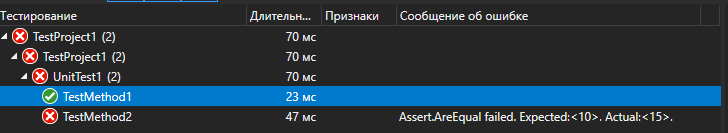
* check – Проверка равны ли числа
* check – Проверка не равны ли числа

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

Для Юнит-теста мы взяли проверку функцию check класса Compare



Результат работы Юнит-теста:



## **3.4 Пример сценария тестирования**

Для сценария тестирования был взять ввод чисел и имени в главном меню приложения.

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **Действия** | **Ожидание** | **Результат** |
| 1 | Ввод букв в поле с отгадкой цифр | Не даст ввести | Совпадает с ожиданием |
| 2 | Ввод только больших букв | Даёт ввести | Совпадает с ожиданием |
| 3 | Ввод латиницы | Даёт возможность ввода | Совпадает с ожиданием |
| 4 | Ввод цифр в поле имени | Даёт ввести | Совпадает с ожиданием |
| 5 | Ввод спец. символов в поля | Не даёт ввести | Совпадает с ожиданием |
| 6 | Нажать кнопку “начать игру” без ввода имени | Не позволяет | Совпадает с ожиданием |
| 7 | Нажать кнопку “начать игру” с двумя символами в поле ввода | Не позволяет | Совпадает с ожиданием |
| 8 | Нажать на кнопку “записать рекорд” до завершения игры | Не позволяет | Совпадает с ожиданием |

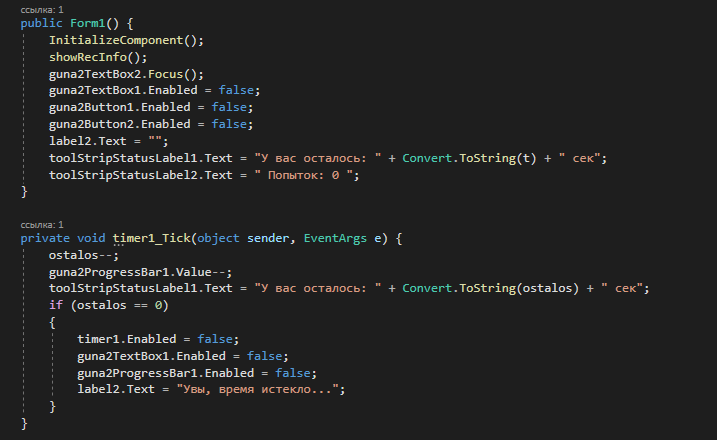
# **4. Кодирование**

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

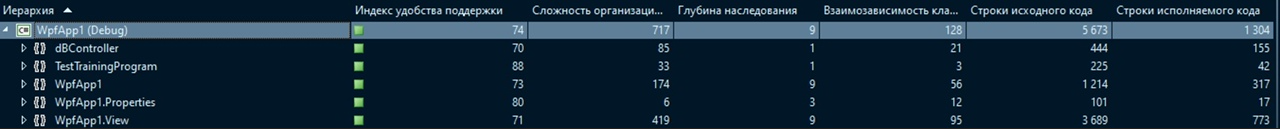
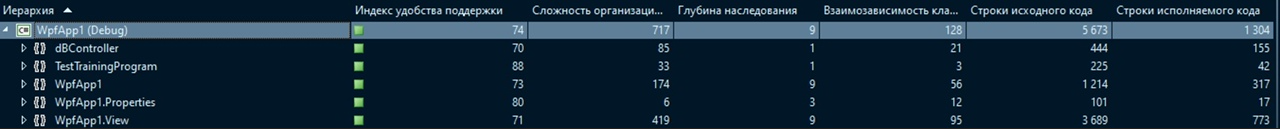
## **4.1 Правила форматирования кода**

* 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



## **4.2 Оценка цикломатической сложности**

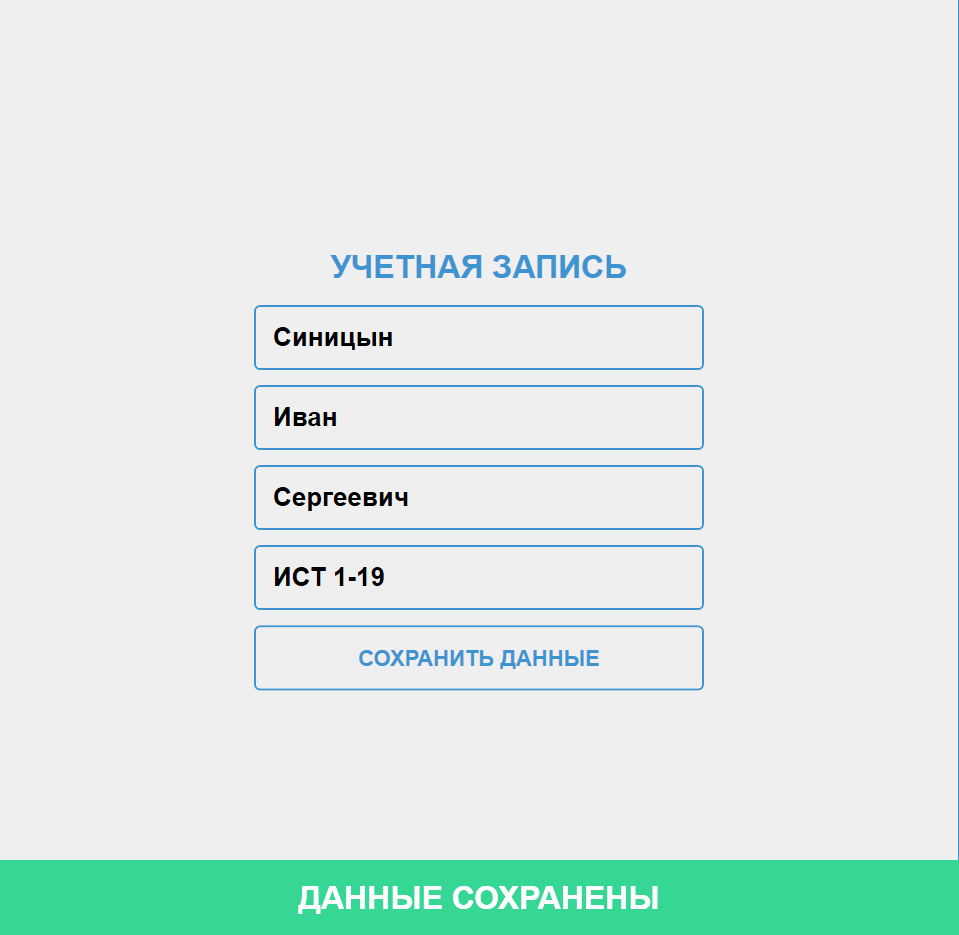
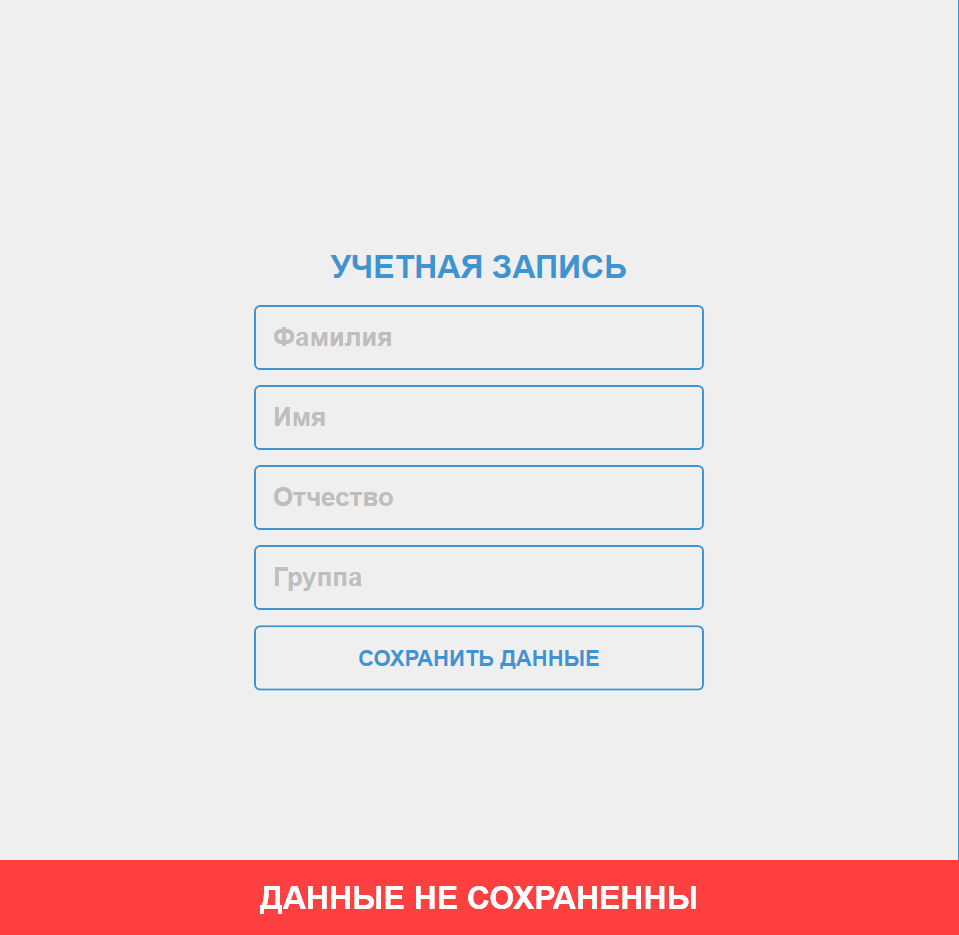
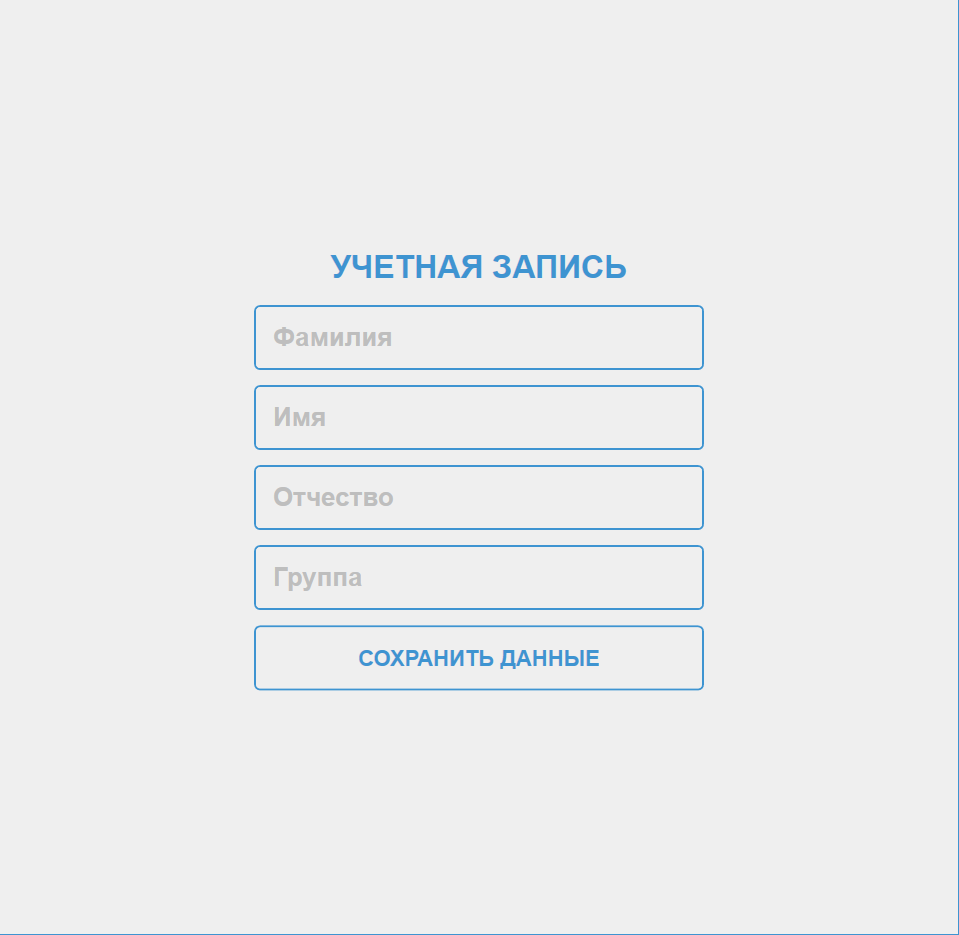


## **4.3 метрика Холстеда**

# **5. Результат работы программы**

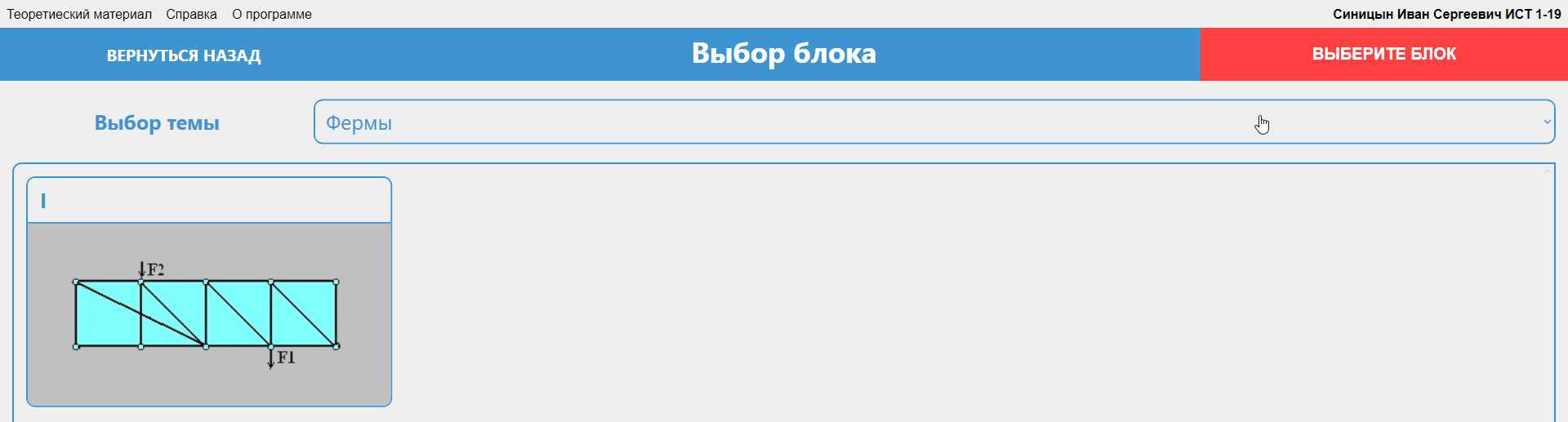
***Цель работы*** – показать работоспособность программы в целом или отдельных функций.

5.1. Пример работы «Пользовательский данных»

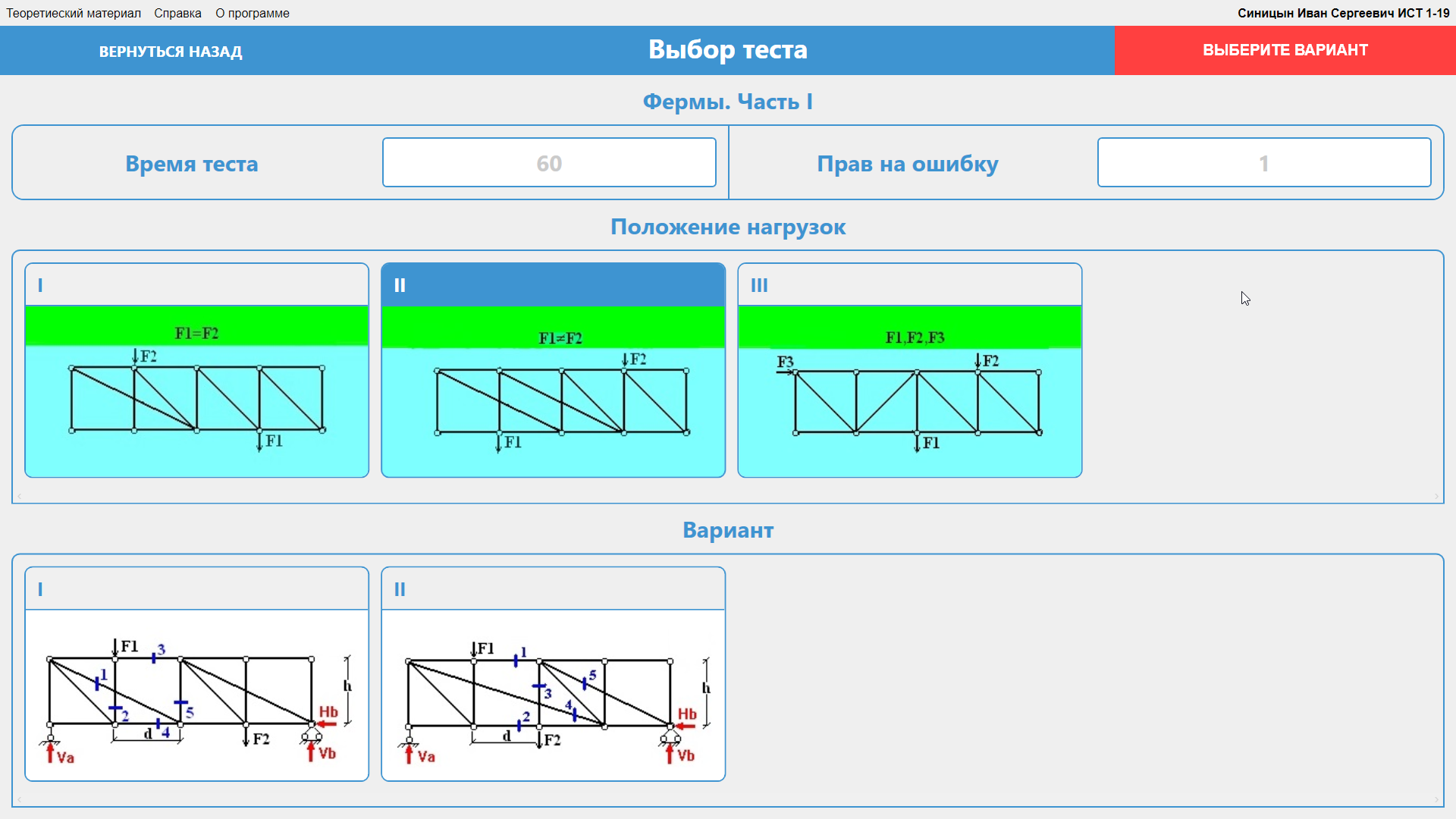


Не вводим данные Данные не сохранены Данные сохранены

5.2 Пример работы выбора части/положения/варианта

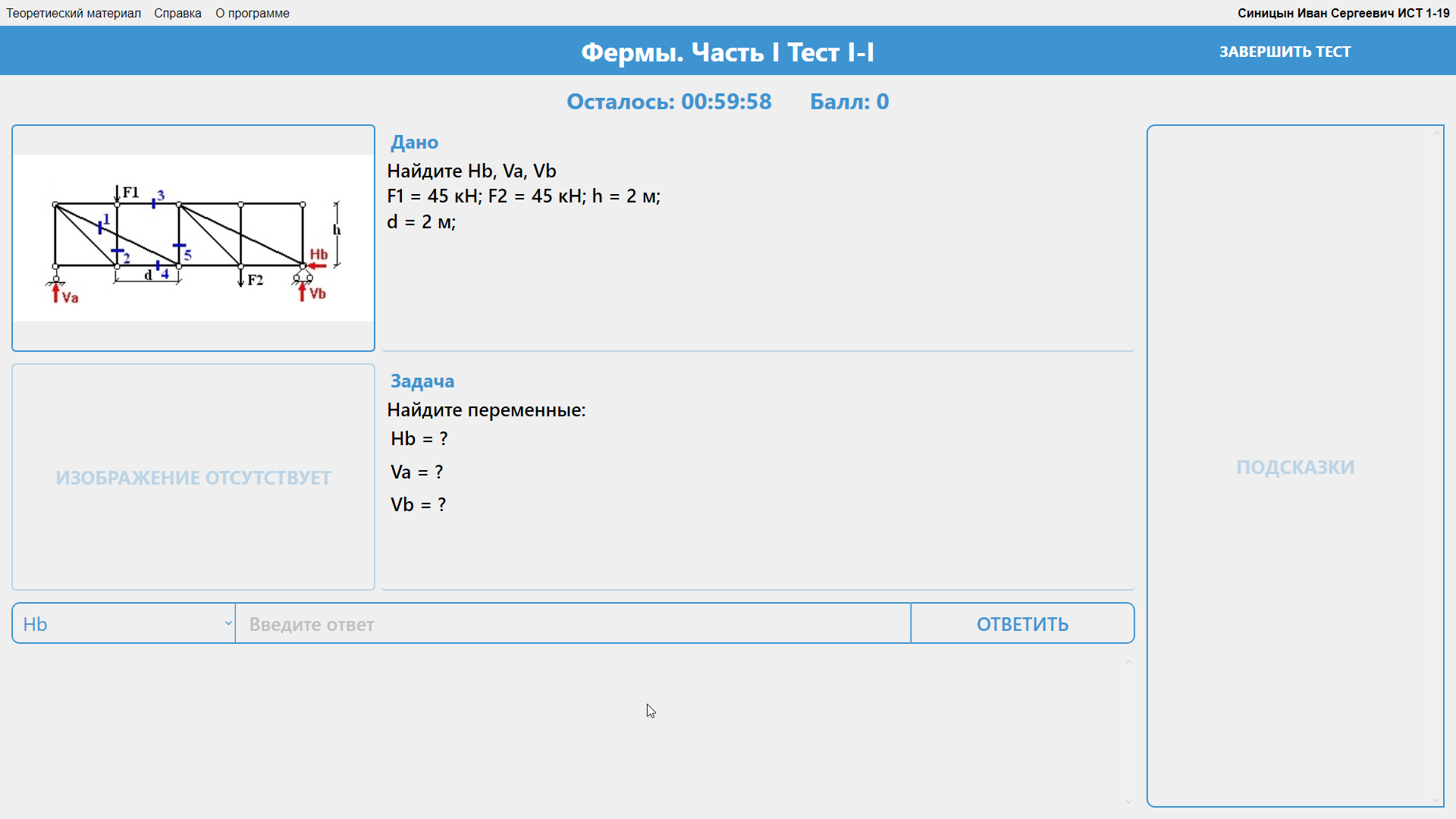


Если блок не будет выбран, то выведется ошибка при нажатии на кнопку «Продолжить»

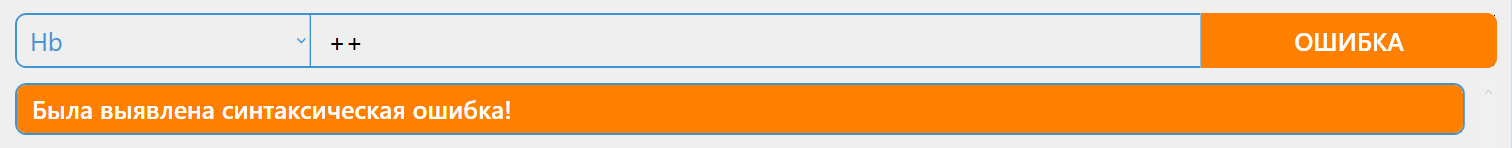


То же самое работает с положением нагрузок и вариантами

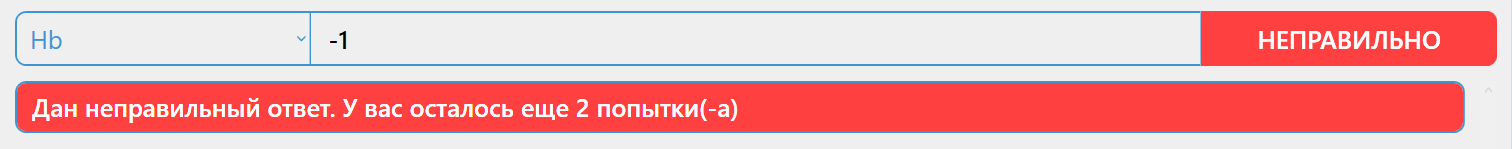
5.3 Пример тестирования



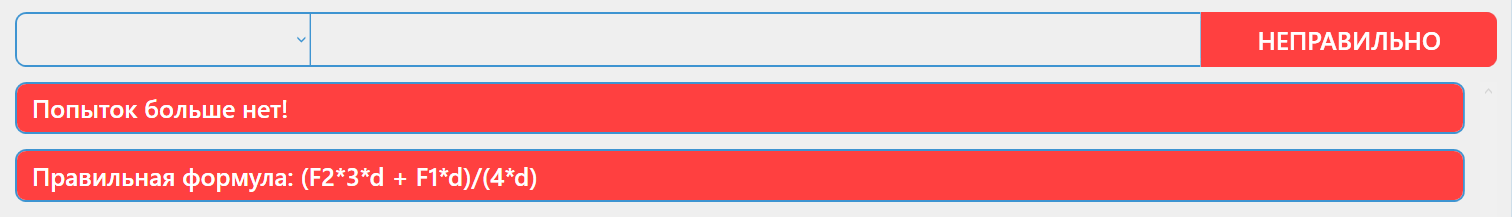
Окно тестирования



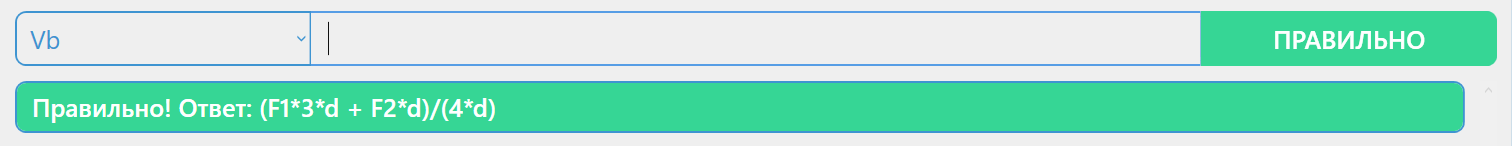
Если допущена синтаксическая ошибка, то будет выведено сообщение (баллы не снимаются)



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

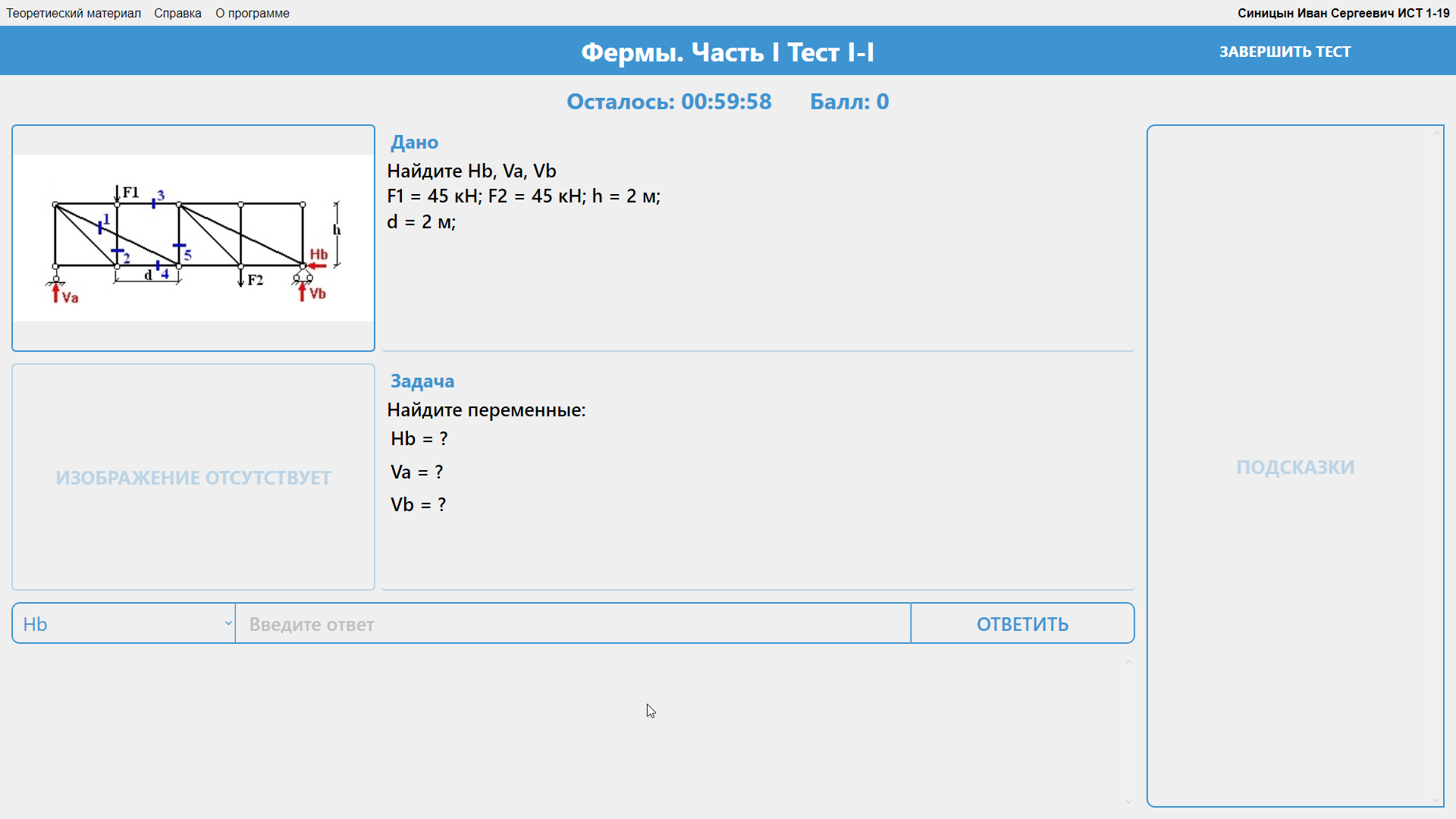


Если попыток больше нет

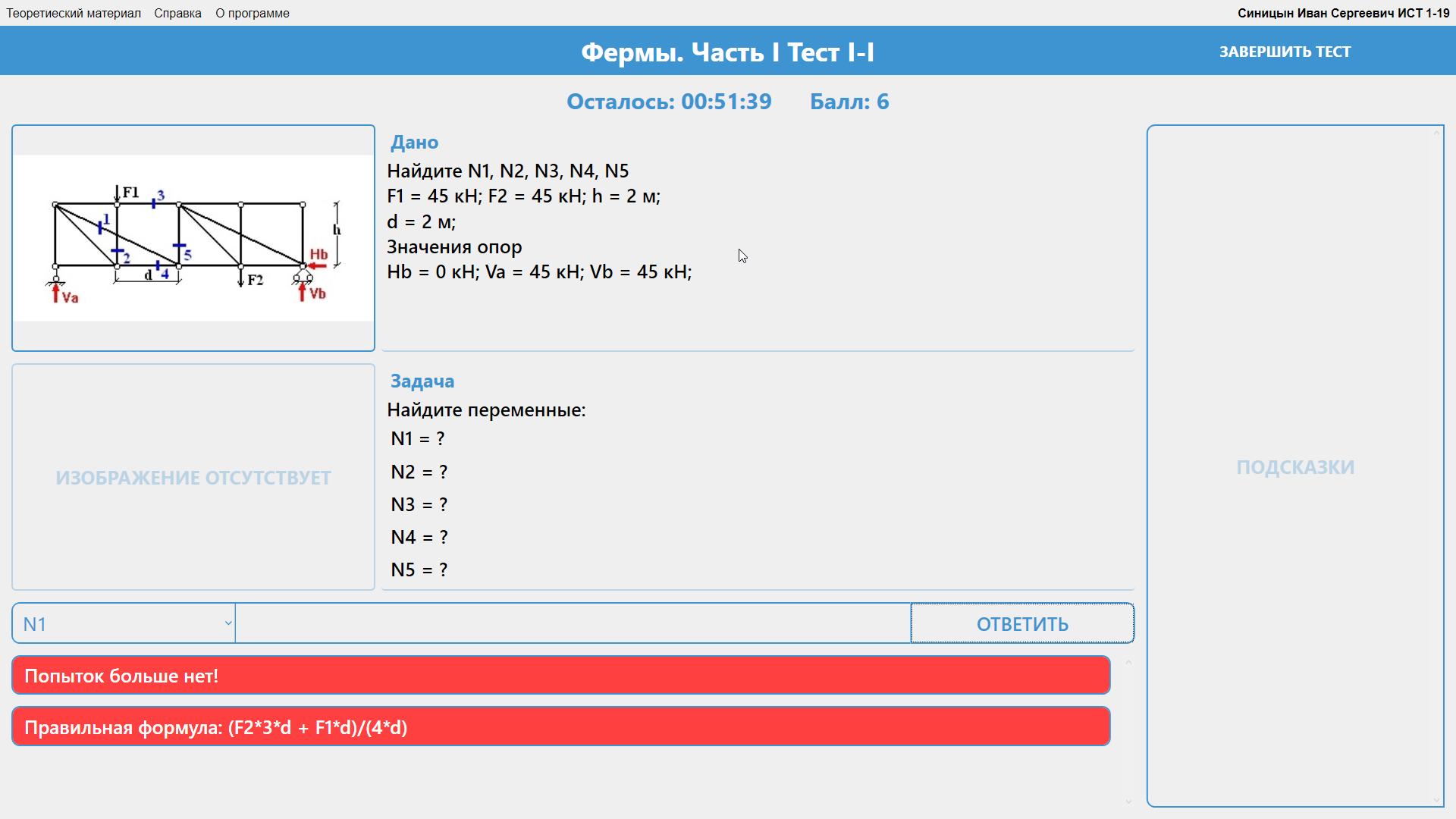


За правильный ответ дается 5 баллов

5.4 Пример смены вопросов

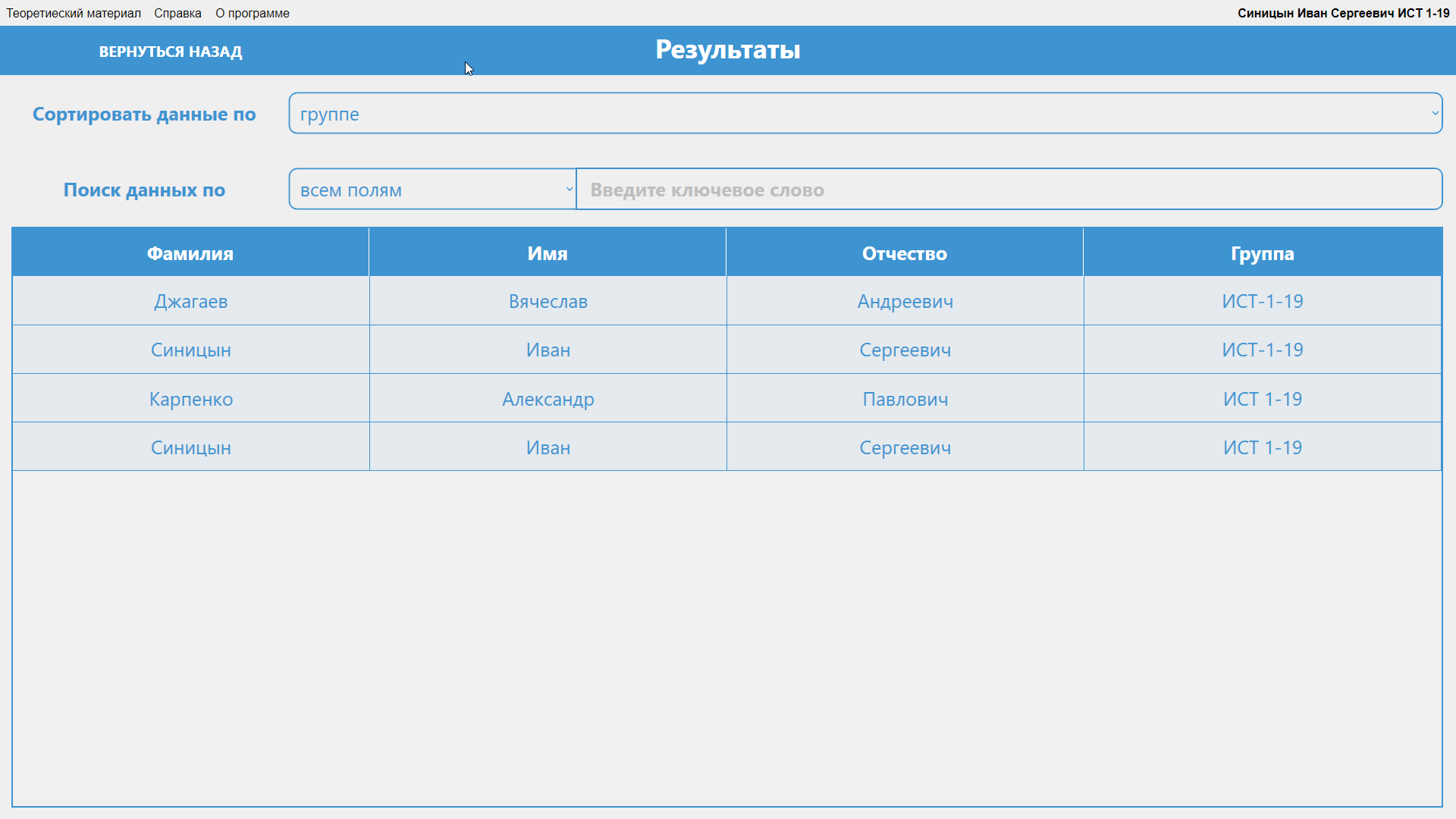


1-ый этап вопросов (нахождение опор)

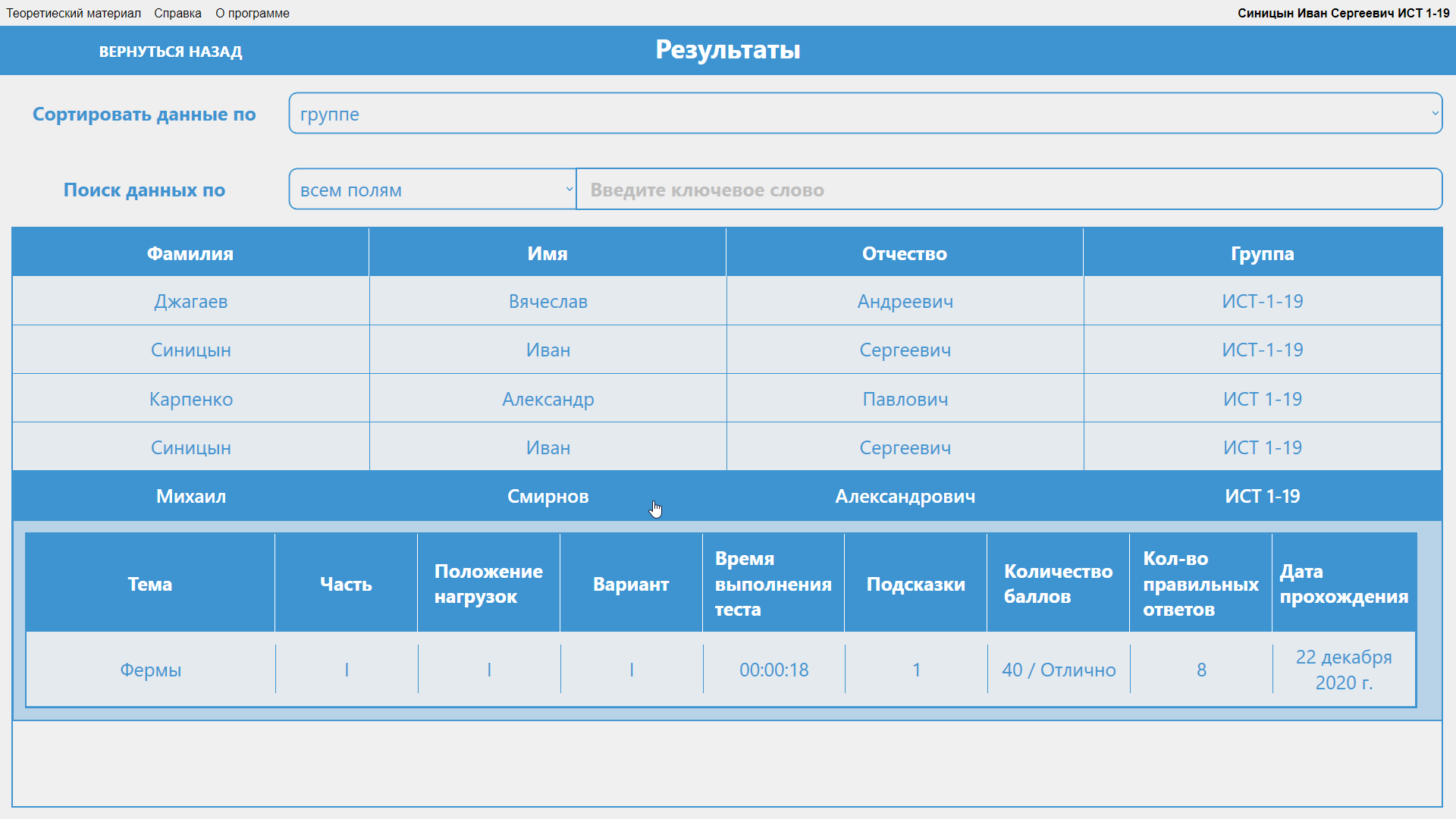


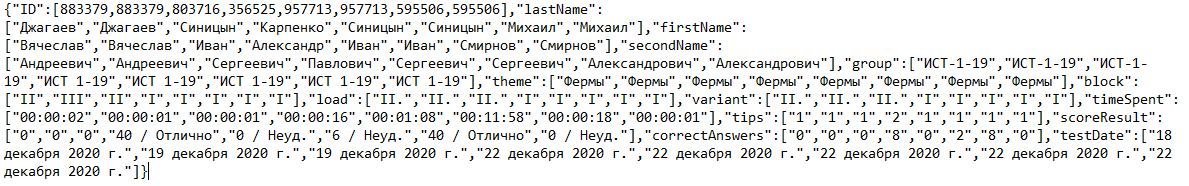
2-ой этап вопросов (нахождение N1,N2,N3,N4,N5)

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