**Министерство образования и науки Российской федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)». Университетский колледж информационных технологий**

Специальность 09.02.03 «Программирование в компьютерных системах»

КУРСОВОЙ ПРОЕКТ

Модуль ПМ.01 Разработка программных модулей программного обеспечения

для компьютерных систем

МДК.01.02 Прикладное программирование

на тему «Обучающий калькулятор»

**Пояснительная записка**

**УКИТ 09.02.03.2017.304.06ПЗ**

|  |  |  |
| --- | --- | --- |
| Группа | П-304к |  |
| Студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Соловьев К. А. |
| Руководитель проекта | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Глускер А. И. |
| Нормативный контроль | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (личная подпись) | Гусева Е. Л. |

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc502079127)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc502079128)

[1.1 Введение в предметную область 5](#_Toc502079129)

[2 Спецификация 5](#_Toc502079130)

[3 Программа и методика испытаний 8](#_Toc502079131)

[4 Технический проект 17](#_Toc502079132)

[5 Тестирование программного продукта 17](#_Toc502079133)

ВВЕДЕНИЕ

Цель курсового проекта – разработка программы “Обучающий калькулятор”

Актуальность заключается в получении навыка работы с большим однообразным проектом, тренировки усидчивости и развитие прочих нужных характеристик для работы с нудными проектами.

Основные задачи, которые были поставлены при разработке:

* изучение предметной области;
* проектирование “Обучающий калькулятор”;
* разработка “Обучающий калькулятор”;
* тестирование “Обучающий калькулятор”;
* инсталлятор;
* руководство пользователя;
* презентация;

Структура курсового проекта:

1. Введение:

Данный раздел предназначен для того, чтобы описать текущий план работы по разработке «Обучающий калькулятор»

1. Основная часть:

Данный раздел описывает предметную область, для которой разрабатывается “Обучающий калькулятор”;

1. Программа и методика испытания:

Данный раздел описывает применяемые методы тестирования и тестовые примеры к разрабатываемому «Обучающий калькулятор»;

1. Спецификация:

Данный раздел описывает техническое задание, которое устанавливает требования к «Обучающий калькулятор»;

1. Технический проект:

Данный раздел описывает архитектурные решения, который применяются при разработке.

1. Реализация программного продукта:

Данный раздел описывает процесс разработки «Обучающий калькулятор»;

1. Тестирование программного продукта:

Данный раздел описывает результаты тестирования “Обучающий калькулятор” по программе и методике испытаний;

1. Заключение:

Данный раздел подводит итог по проделанной работе;

1. Список используемых источников:

В данном разделе приводиться список используемых источников, которые использовались при разработке “Обучающий калькулятор”;

1. Приложение:

В данном разделе приводятся дополнительная информация, которая включает в себя руководство оператора, исходный код приложения, протокол системы контроля версий, руководство программиста, доклад.

ОСНОВНАЯ ЧАСТЬ

1.1 Введение в предметную область

**1.1.1 Введение**

Обучающий калькулятор – самый простой калькулятор, который может выводить процесс выполнения всех действий с числами до десяти знаков в виде столбика на экран.

2 Спецификация

**2.1 Введение**

В данном разделе описана основная информация о программном продукте, требования по эксплуатации и разработке, а также выполняемый им функционал.

**2.3Наименование программы**

“Обучающий калькулятор»

**2.4 Область применения**

Младшие классы и подготовительные группы дошкольников

**2.5 Объект, в котором используют программу**

Программа «Обучающий калькулятор» сможет быть использована в любых объектах

**2.6 Основание для разработки**

**2.7 Документ, на основании которого ведется разработка**

Техническое задание на курсовой проект

**2.8 Наименование и (или) условное обозначение темы для разработки**

Курсовая работа – 66

**2.9 Назначение разработки**

**2.9.1 Функциональное назначение**

Обучение дошкольников и детей младших классов подсчету чисел

**2.9.1 Эксплуатационное назначение**

Программа предназначена для детей школьного и дошкольного возраста

**2.10 Требования к программе или программному изделию**

**2.10.1 Требования к функциональным характеристикам**

**2.10.2 Требования к составу выполняемых функций**

Программа должна обеспечивать выполнение следующих функций:

* функция ввода чисел;
* функция выбора действий;
* функция подсчета чисел;
* функция вывода столбика для каждого действия.

**2.11 Требования к временным характеристикам**

1200 микросекунд на срабатывание нажатия кнопки

**2.12 Требования к надежности**

Другие требования к надежности не предъявляются

**2.13 Условия эксплуатации**

**2.14 Персонал**

Минимальное количество персонала – один человек, владеющий русским языком, обладающий практическими навыками по использованию программ.

**2.14. Специальные требования к эксплуатации**

Другие специальные требования к условиям эксплуатации не предъявляются

**2.15 Требования к составу и параметрам технических средств**

В состав технический средств должен входить компьютер, включающий:

– процессор Pentium или совместимый с ним;

– клавиатуру;

– видеокарту;

– монитор;

– жесткий или SSD-диск.

**2.16 Требования к информационной и программной совместимости**

**2.16.1 Требования к информационным структурам на входе и выходе**

Требования к информационным структурам не предъявляются

**2.16.2 Операционная система**

Программа должна работать под управлением операционной системы Windows 10.

**2.16.3 Язык программы**

Программа должна быть написана на языке Паскаль и компилироваться в трансляторе Free Pascal 2.6 или выше.

**2.16.4 Требования к защите**

Требования к защите информации и программ не предъявляются

**2.17 Требование к пользовательскому интерфейсу**

**2.17.1 Интерфейс**

Графический

**2.17.2 Управление**

Управление происходит манипулированием мышкой, а ввод чисел производится с клавиатуры

**2.18 Требования к программной документации**

**2.18.1 Состав программной документации**

Состав программной документации должен включать:

– техническое задание;

– пояснительную записку;

– текст программы;

–текст программы, осуществляющей автоматическое тестирование

программы «Обучающий калькулятор»;

– программу и методику испытаний;

– Руководство пользователя;

**2.18.2 Специальные требования к пояснительной записке**

Пояснительная записка должна содержать блок-схему (блок-схемы) алгоритма(-ов), используемых в программе.

**2.19 Требование к исходным кодам**

Требования к исходным кодам изложены в документе: А. И. Глускер

«Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011 /раздел 3.1/ (п. 4.5.3 технического задания).

**2.19.1 Язык программы**

Программа должна быть написана на языке Паскаль и компилироваться транслятором Free Pascal 2.6 (п. 4.5.4 технического задания).

**2.20 Стадии и этапы разработки**

**2.20.1 Стадии разработки**

Разработка осуществляется в три стадии:

* техническое задание;
* технический проект;
* рабочий проект.

**2.20.2 Этапы разработки**

На стадии техническое задание осуществляется разработка, согласование и утверждение технического задания в срок до 25 сентября 2017 года Глускером А. И.

На стадии технический проект осуществляется разработка, согласование и утверждение пояснительной записки в срок до 2 октября 2017 года Глускером А. И.

На стадии рабочий проект осуществляется разработка текста программы, осуществляющей автоматическое тестирование программы «Обучающий калькулятор», разработка, согласование и утверждение программы и методики испытаний, текста программы в срок до 20 декабря 2017 года Глускером А. И., после чего осуществляются испытания по результатам которой возможно будет проводиться корректировка программной документации в срок до 31 декабря 2017 года.

**2.21 Порядок контроля и приемки**

Приемосдаточные испытания должны проводиться в соответствии с программой и методикой испытаний, разработанной, согласованной и утвержденной не позднее 31 декабря 2017 года.

**2.22 Вывод**

В данном разделе были изложены все основные особенности программного продукта и требования к нему. Стал более умело составлять ТЗ.

3 Программа и методика испытаний

**Введение**

В данном разделе описана “Программа и методика испытаний”

**3.1.1. Объект испытаний**

**3.1.1.1 Наименование**

Программа “Обучающий калькулятор”

**3.1.1.2 Область применения**

Младшие классы и подготовительные группы дошкольников.

**3.1.1.3 Обозначение программы**

Курсовой – 66.

**3.1.2. Цель испытаний**

Проверка соответствия программного продукта требованиям технического задания.

**3.1.3. Требования к программе**

**3.1.3.1 Требования к функциональным характеристикам**

**3.1.1 Требования к составу выполняемых функций**

Программа должна обеспечивать выполнение следующих функций (п. 3.1.4.1.1 технического задания):

* функция ввода чисел;
* функция выбора действий;
* функция подсчета чисел;
* функция вывода столбика для каждого действия.

**3.1.5.1.2 Требования к временным характеристикам**

1200 микросекунд на срабатывание нажатия кнопки

**3.1.6.2 Требования к информационной и программной совместимости**

Программа должна работать под управлением операционной системы Windows 10.

**3.1.4. Требования к программной документации**

**3.1.4.1 Состав программной документации**

Состав программной документации должен включать:

* техническое задание;
* пояснительную записку;
* текст программы;
* текст программы, осуществляющей автоматическое тестирование программы «Обучающий калькулятор»;
* программу и методику испытаний;
* руководство пользователя;

**3.1.4.2 Специальные требования к пояснительной записке**

Пояснительная записка должна содержать блок-схему (блок-схемы) алгоритма(-ов), используемых в программе

**3.1.4.3 Специальные требования к тексту программы**

**3.1.4.3.1 Требования к исходным кодам**

Требования к исходным кодам изложены в документе: А. И. Глускер «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011 /раздел 3.1/ (п. 4.5.3 технического задания).

**3.1.4.3.2 Язык программы**

Программа должна быть написана на языке Паскаль и компилироваться в трансляторе Free Pascal 2.6 или выше.

**3.1.5 Средства и порядок испытаний**

**3.1.5.1 Технические средства, используемые при проведении испытаний**

В состав технический средств входит IBM-совместимый компьютер, включающий:

– процессор Pentium или совместимый с ним;

– клавиатуру;

– видеокарту;

– монитор;

– жесткий или SSD-диск.

**3.1.5.2 Программные средства, используемые при проведении испытаний**

* В состав программных средств входит: лицензионная копия операционной системы Windows 10 (любой версии), локализованная для работы в Российской Федерации;
* Lazarus 1.6.0 (вариант, предназначенный для работы в

среде Windows).

**3.1.5.3 Порядок проведения испытаний**

**3.1.5.3.1 Подготовка к проведению испытаний**

Подготовка заключается в обеспечении наличия компьютера, описанного в п. 3.1.5, и программных средств, указанных в п. 5.2, установленных на этом компьютере

**3.1.5.3.2 Ход проведения испытаний**

Проведения испытаний документируется в протоколе, где указывается перечень проводимых испытаний, результат каждого испытания и возможно замечания.

**3.1.5.3.3 Состав испытаний**

**3.1.5.3.3.1 Проверка состава программной документации в соответствии с методом, описанном в п. 6.2**

**3.1.5.3.3.2 Проверка требований к программе**

Проверка обеспечения требований к программе (п. 3) в соответствии с методом, описанным в п. 6.1

**3.1.5.3.3.3 Проверка требований к программной документации**

**3.1.5.3.3.3.1 Проверка пояснительной записки**

Проверка производится в соответствии с методом, описанным в п. 6.3

**3.1.5.3.3.3.2 Проверка текстов программ (п. 3.1.4.3.1)**

Проверка производится в соответствии с методом, описанным в п.6.4

**3.1.5.3.3.3.3 Проверка текстов программ (п. 3.1.4.3.2)**

Проверка производится в соответствии с методом, описанным в п. 6.5

**3.1.6. Методы испытаний**

**3.1.6.1 Метод проверки требований к программе**

Проверка осуществляется путем запуска программы и сравнения результатов ее запуска с ожидаемыми, при этом осуществляется измерения интервала времени между окончанием ввода и окончанием вывода результатов (Таблица 1).

Таблица 1 - Список тестовых действий

|  |  |  |
| --- | --- | --- |
| № | Действия | Результат |
| 1 | Выбор действия | Доступно: деление, умножение, вычитание, сложение |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| № | Действия | Результат |
| 2 | Ввод числа в первое поле | Использование этого числа в качестве первого параметра Пример: Для деления это будет делимое |
| 3 | Ввод числа во второе поле | Использование этого числа в качестве второго параметра.  Пример:  Для деления это будет делитель |
| 4 | Кнопка “=” | Выводит результат выбранного действия на экран. Результат должен совпадать с действительностью и все действия должны работать правильно |
| 5 | Ввод символов | Должны вводиться только числа и символ «,»; |
| 6 | Проверить результат  при пустых полях | Появляется окно с ошибкой |
| 7 | Деление: 1/1111 |  |
| 8 | Деление: 1/1 |  |
| 9 | Деление: 1,1/1,2 |  |
| 10 | Деление X/0 | Появляется окно с ошибкой |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| № | Действия | Результат |
| 11 | Умножение: 1234/221 |  |
| 12 | Умножение: 23,5/71,8 |  |
| 13 | Вычитание: 245 - 100 |  |
| 14 | Вычитание: 34,7 – 17,2 |  |
| 15 | Сложение: 345 + 100 |  |
| 17 | Сложение: 71,3 + 23,6 |  |

— Если все тестовые примеры совпадают с контрольными, то в протокол вносится запись: «Требования к составу выполняемых функций» – соответствует; в противном случае: «Требования к составу выполняемых функций» – не соответствует.

— В случае, если длина временного интервала от окончания ввода входных данных до окончания вывода результатов, не превышает 1200 микросекунд для каждого тестового примера, то в протокол вносится запись: «Требования к временным характеристикам» – соответствует; в противном случае «Требования к временным характеристикам» – не соответствует

**3.1.6.2 Метод проверки требований к составу программной документации**

Проверка состава программной документации осуществляется визуально путем сравнения набора предъявленных документов (в форме распечатки или в рукописной форме) списку, приведенному в п. 4.1. При этом исходные тексты программ должны быть предоставлены так же и в электронной форме. В случае если набор предъявленных документов соответствует списку, а исходные тексты предоставлены также в электронной форме, то в протокол заносится запись: «Состав программной документации» – соответствует; в противном случае: «Состав программной документации» – не соответствует.

**3.1.6.3 Метод проверки требований к пояснительной записке**

Проверка состоит из следующих этапов:

* проверка наличия блок-схемы (блок-схем) в пояснительной записке;
* проверка соблюдения требований ГОСТ 19.701-90 для каждой блок-схемы;
* проверка соблюдения локальных стандартов для блок-схем;
* проверка соответствия каждой блок-схемы алгоритму, закодированному

в программе. Проверка соблюдения требований ГОСТ 19.701-90 состоит из следующих работ:

* проверка использования только тех символов, которые указаны как применимые к схемам программ в п. 5 ГОСТ 19.701-90;
* проверка соответствия символов их назначению (экспертная оценка лица, проводящего испытания);
* проверка правильности выполнения соединения линий (п. 4.2.3 ГОСТ 19.701-90);
* проверка того, что линии потока управления, выходящие из символа «решение» подписана (п. 4.3.1.2 ГОСТ 19.701-90); Проверка соблюдения локальных стандартов для блок-схем состоит из следующих работ:
* проверка того, что все символы (кроме терминаторов, соединителей, линий и комментариев) имеют одинаковые размеры;
* проверка того, что терминаторы имеют ту же ширину, что и другие символы; проверка того, что отношение ширины к высоте составляет 2 к 1 для каждого символа, кроме терминаторов, комментариев и линий
* проверка того, что отношение ширины к высоте составляет 4 к 1 для терминаторов;
* проверка того, что высота соединителей совпадает с высотой терминаторов;
* проверка того, что линии потока управления входят в символ слева или сверху, а выходят снизу или справа;
* проверка того, что подписи к линиям не находятся на самих линиях. Проверка соответствия каждой блок-схемы алгоритму, закодированному в программе, осуществляется путем экспертной оценки лицом, осуществляющим проведение испытаний. В случае, если все вышеприведенные проверки прошли успешно, в протокол заносится запись: «Специальные требования к пояснительной записке» – соответствует; в противном случае «Специальные требования к пояснительной записке» – не соответствует.

**3.1.6.4 Метод проверки требований к исходным кодам**

Изложенный ниже метод применяется ко всем файлам, содержащим исходный текст, и входящим в состав программной документации по отдельности. Для каждого файла вносится в протокол запись: «Требования к исходным кодам для файла ####» – соответствует/не соответствует (где вместо #### указывается название файла). Проверка состоит из следующих этапов:

* наличие комментария в начале файла, содержащего автора работы, номера задания и варианта, краткой формулировки задания (или его части);
* наличие комментариев к неочевидным действиям (проверяется методом экспертной оценки лицом, осуществляющим испытания);
* для каждой подпрограммы наличие комментария, содержащего полное описание ее работы, описание всех аргументов и результатов. Достаточность этого комментария для возможности использовать подпрограмму в других программах (без изучения собственно текста подпрограммы);
* для каждой глобальной переменной указание ее назначения;
* для всех переменных, кроме переменных цикла, использование «говорящих» названий;
* для всех подпрограмм использование говорящих названий;
* использование одного оператора на одной строке программы;
* количество пробелов перед строкой программы должно соответствовать уровню вложенности (по два пробела на уровень вложенности);
* слова begin и end, соответствующие друг другу, располагаются строго с одной и той же позиции по вертикали;
* количество строк в подпрограмме и в самой программе (между begin и end) – не более 25 строк;
* использование модулей для трех и более сходных по назначению подпрограмм;
* отсутствие в подпрограммах использования глобальных переменных (напрямую);
* разделение подпрограмм на предназначенные для вычислений (в них не должно быть ввода-вывода) и на предназначенные для ввода-вывода (в них вычисления должны быть только такие, что нужны для ввода-вывода);
* отсутствие операторов goto, break, continue; процедур halt и exit;
* проверка того, что вместо явно указанных значений чисел, в тексте программы используются константы.

В случае, если все перечисленные этапы пройдены, то в протокол о соответствии файла требованиям, в противном случае – о несоответствии.

**3.2 Вывод**

В данном разделе была описана “Программа и методика испытаний”

4 Технический проект

**4.1 Введение**

В данном разделе описан «Технический проект»

**4.1.1 Наименование программы**

«Обучающий калькулятор»

**4.1.2 Условное обозначение темы**

Курсовая работа – 66

**4.1.3 Документ, на основании которого ведется разработка**

Техническое задание на курсовой проект

**4.2 Назначение и область применения**

**4.2.1 Назначение программы**

Обучение дошкольников и детей младших классов подсчету чисел

**4.2.2 Краткая характеристика области применения**

Программа может использоваться любыми лицами

**4.3 Технические характеристики**

**4.3.1 Постановка задачи на разработку программы**

Программа для обучения детей считать

**4.3.2 Описание структуры // ПУСТО**

5 Тестирование программного продукта

**5.1 Введение**

В данном разделе проводиться тестирование, продукта и проверка на соответствие требований, заявленных в программе и методике испытания.

**5.2 Протокол тестирования**

* функция ввода чисел – присутствует;
* функция выбора действий – присутствует;
* функция подсчета чисел – присутствует;
* функция вывода столбиков – присутствует;
* ограничения – присутствуют;
* требование к временным характеристикам – соответствует;
* требования к информационной и программной совместимости – выполнено;
* состав программной документации – соответствует заявленному требованию.

Исходный код был проверен, они соответствуют заявленным требованиям.

Была произведена проверка программы.

Выбор действий – присутствует (Рисунок \*\*\*).

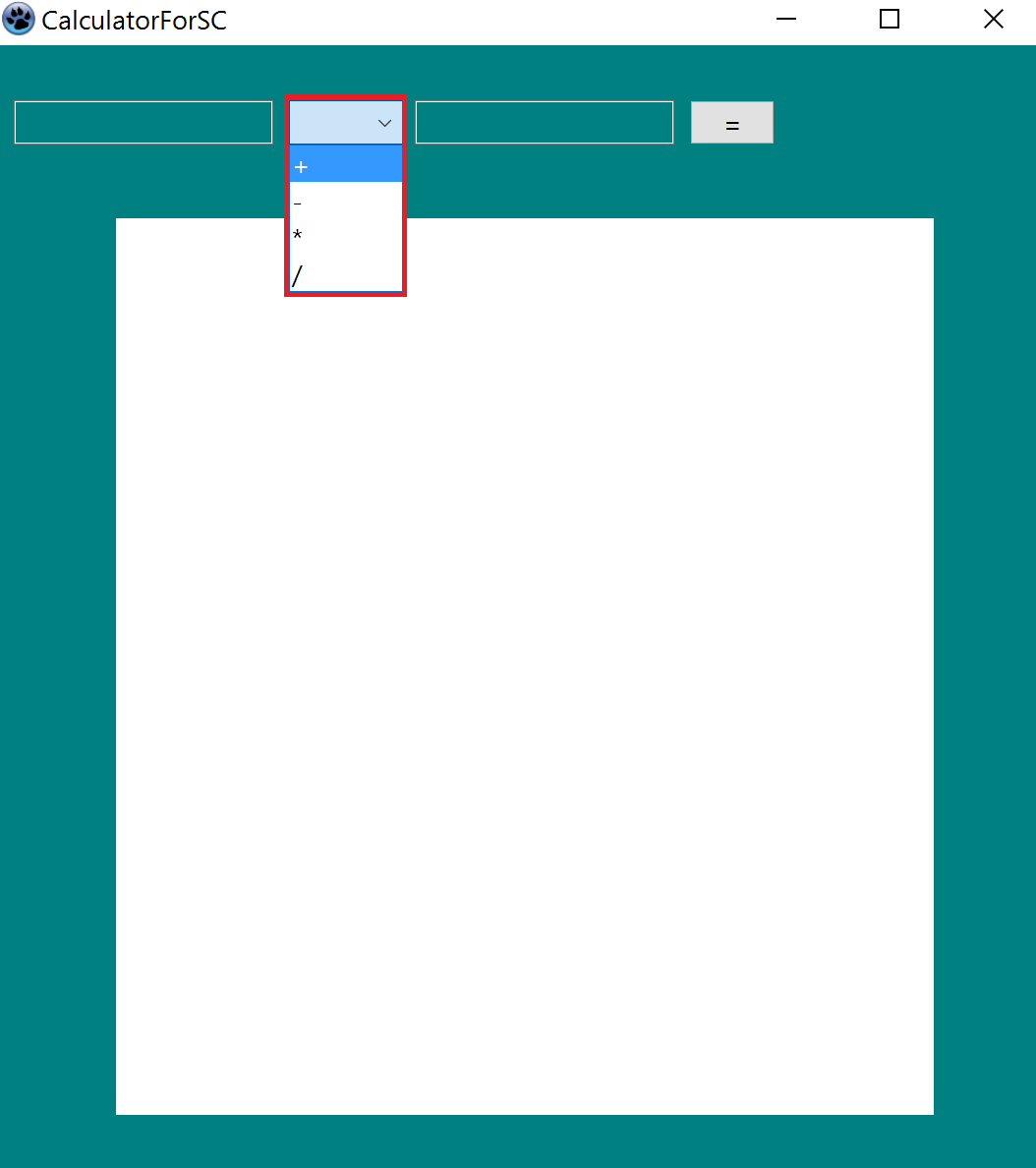


Рисунок \*\*\* – Выбор действий

Ввод числа в первое поле работает исправно (Рисунок \*\*\*).

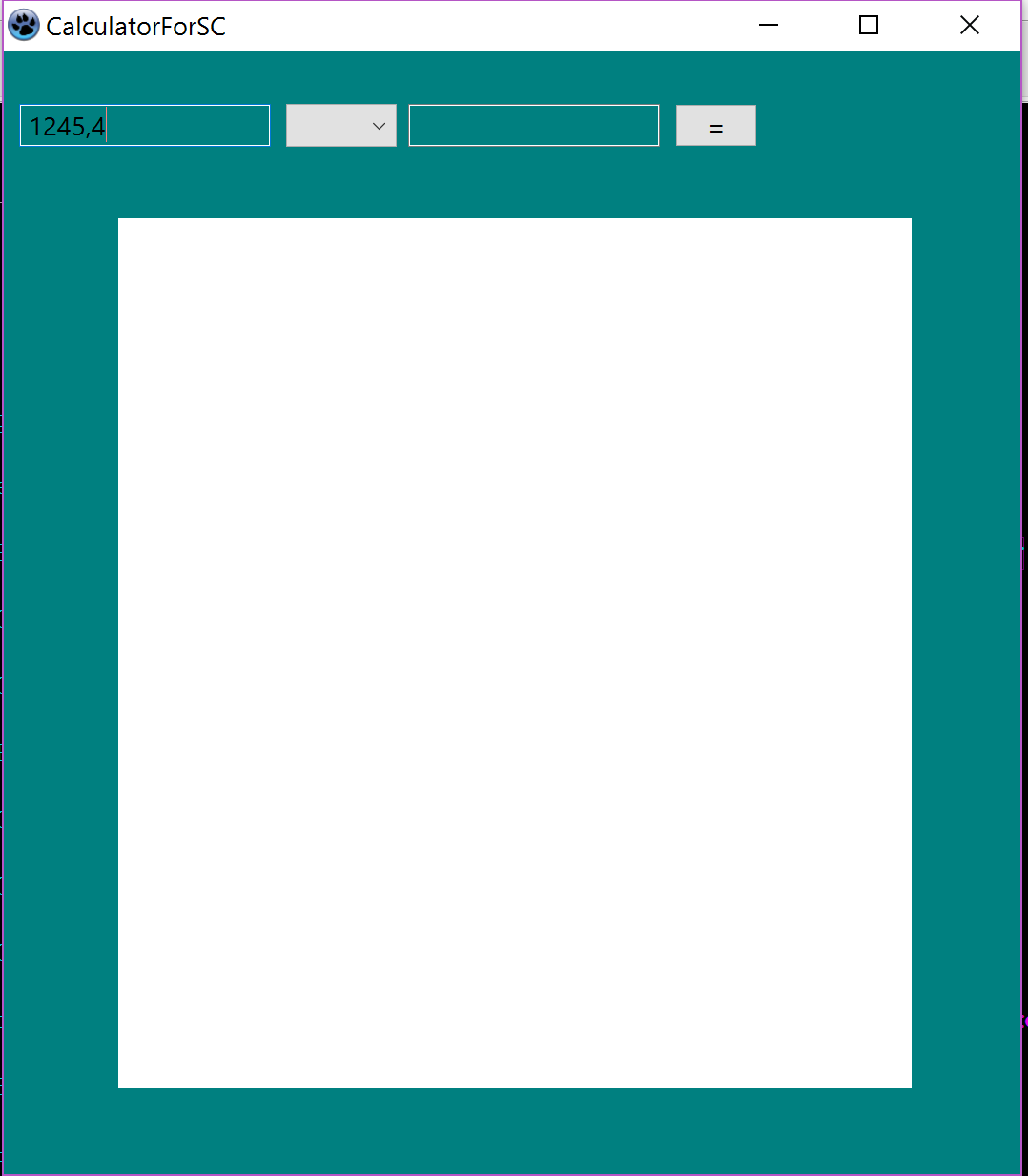


Рисунок **\*\*\*** – Ввод первого числа

Ввод числа во второе поле работает исправно (Рисунок \*\*\*).

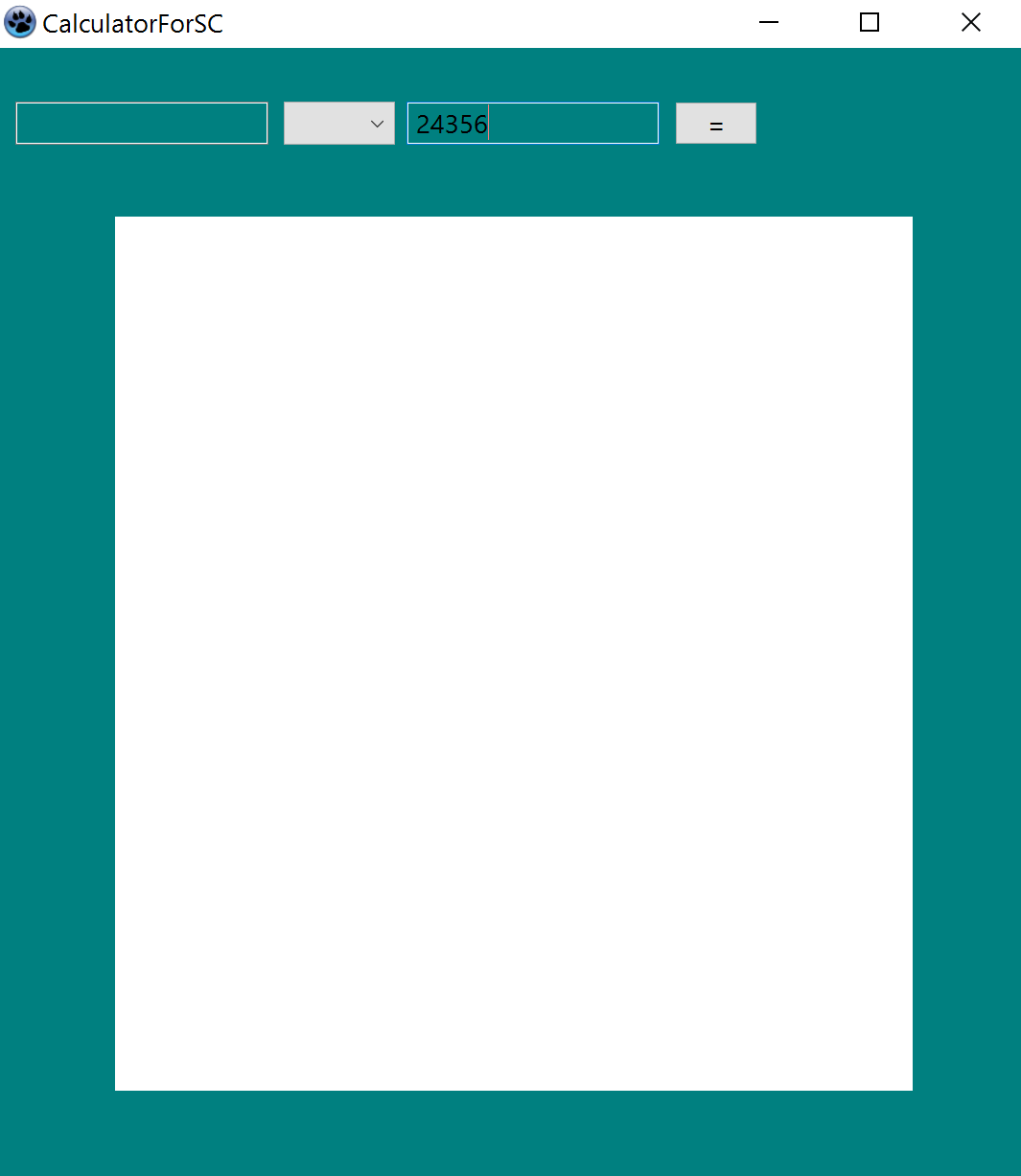


Рисунок **\*\*\*** – Ввод второго числа

Кнопка “=” (кнопка вычисляющая результат) работает правильно (Рисунок \*\*\*)

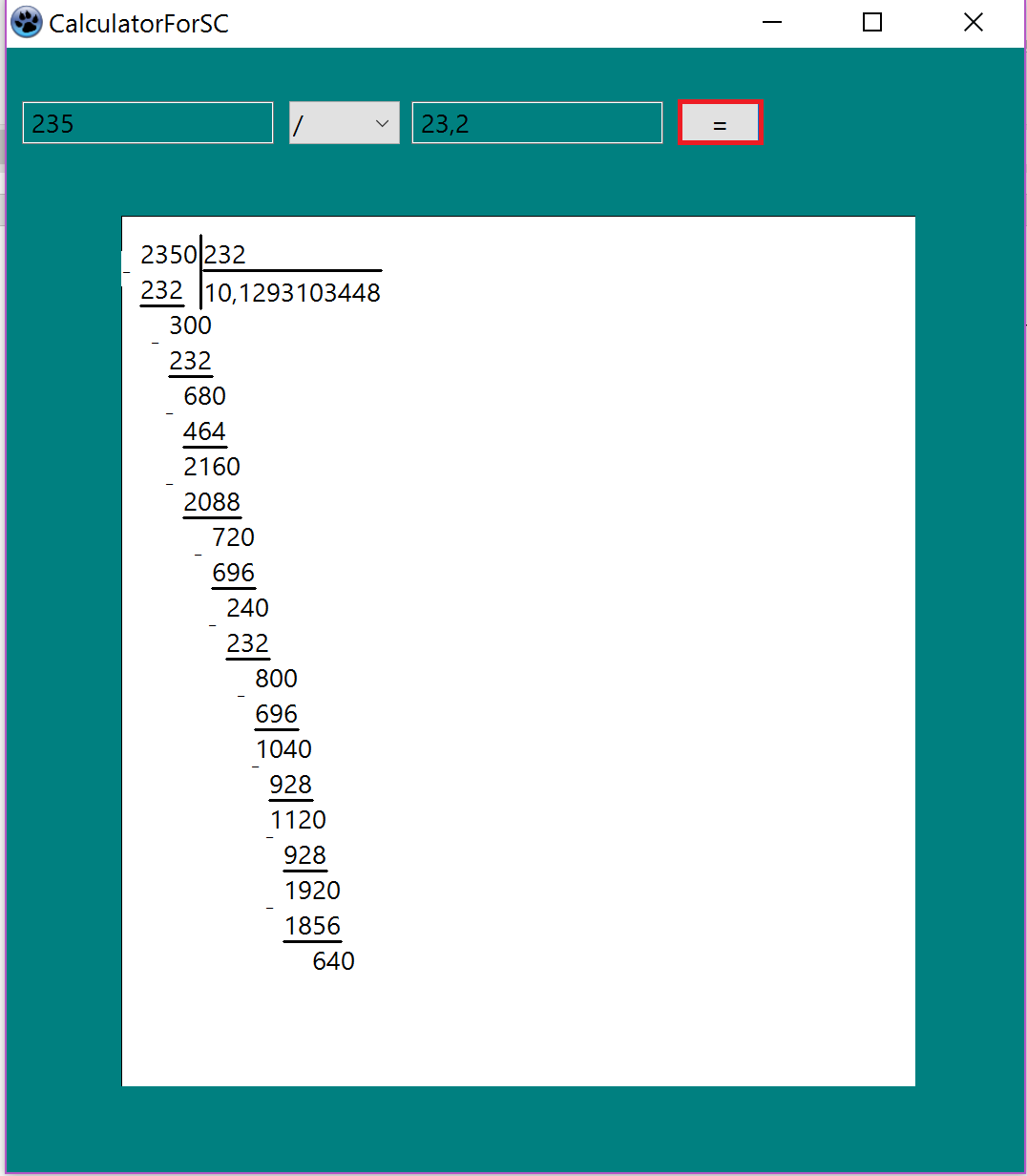


Рисунок – \*\*\* (Работа кнопки показана на примере деления)

Ограничение ввода символов работает исправно (Рисунок \*\*\*)

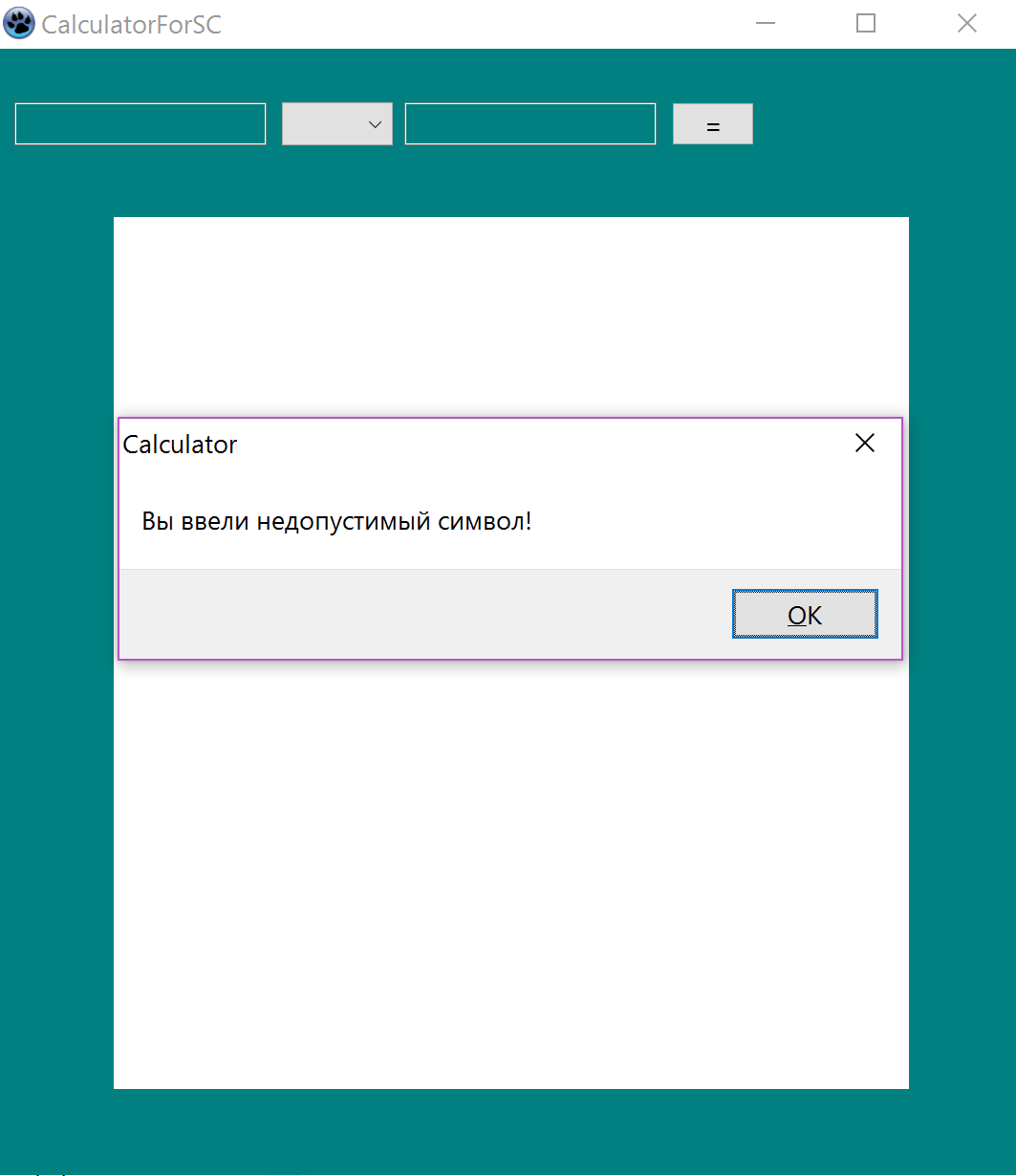
****

Рисунок – **\*\*\***

Проверка на пустые поля работает исправно (Рисунок \*\*\*)

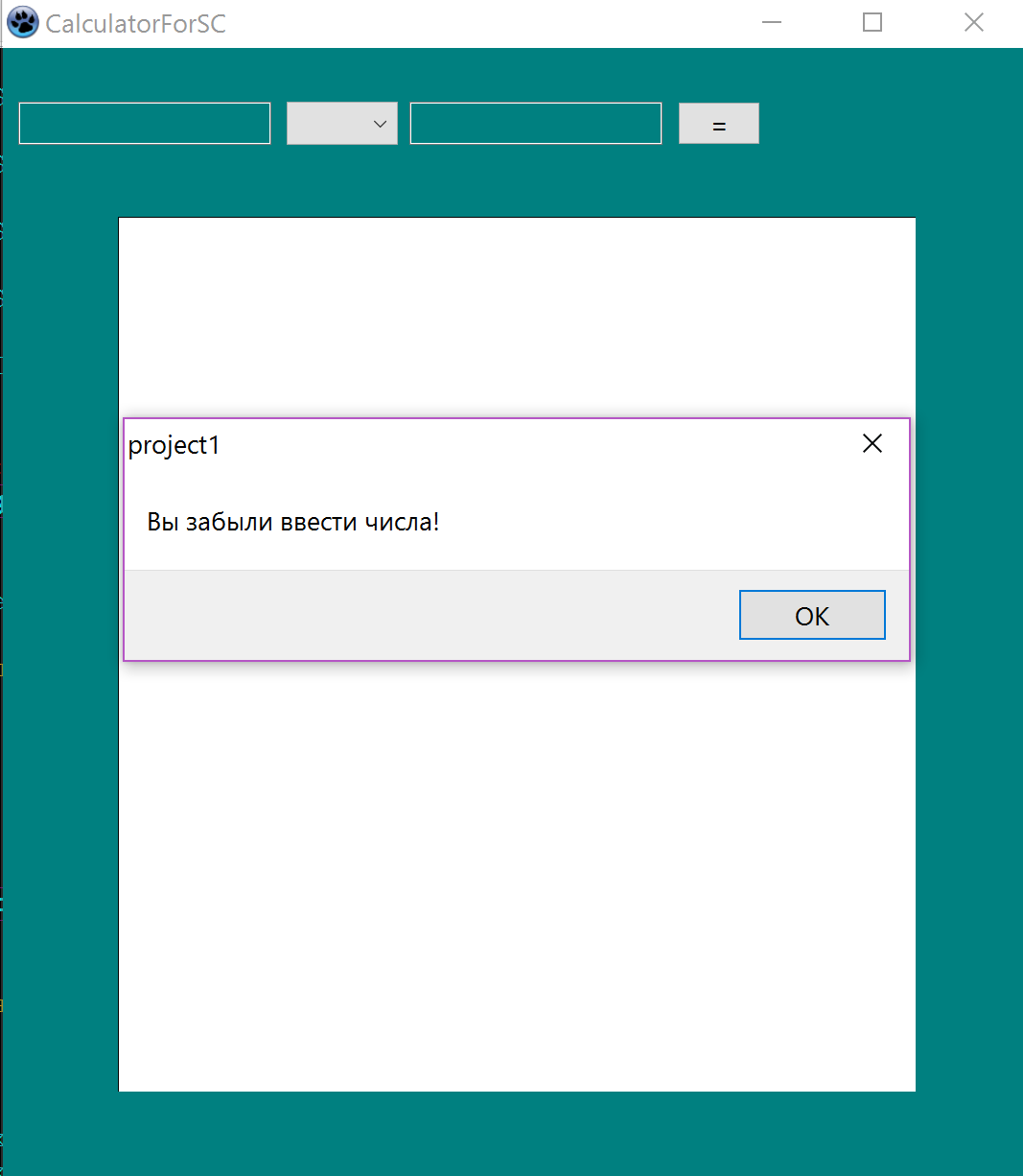


Рисунок – \*\*\*

Деление меньшего числа на большее в некоторых случаях работает некорктно

(Рисунок \*\*\*)

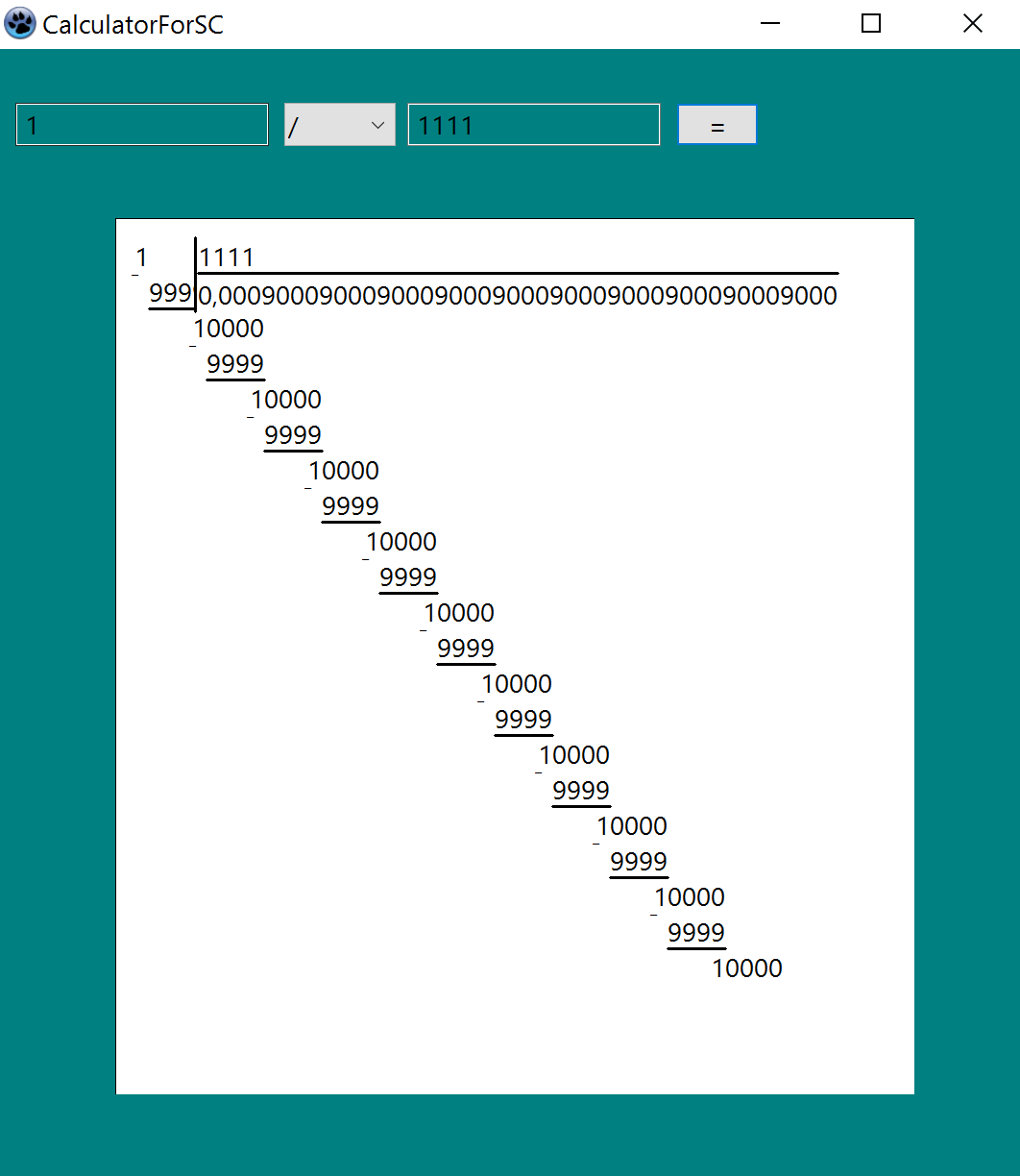


Рисунок – \*\*\*

Деление числа без остатка и с одним действием работает правильно (Рисунок \*\*\*)

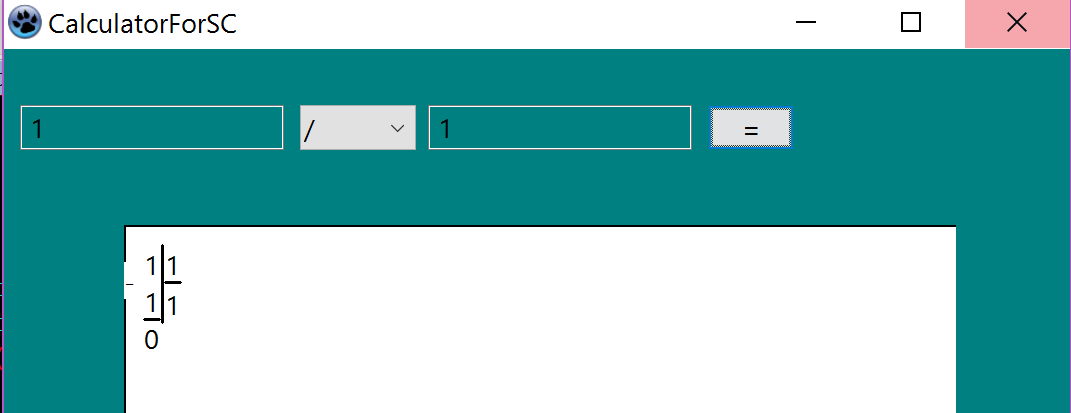


Рисунок – \*\*\*

Деление дробных чисел работает правильно (Рисунок \*\*\*)

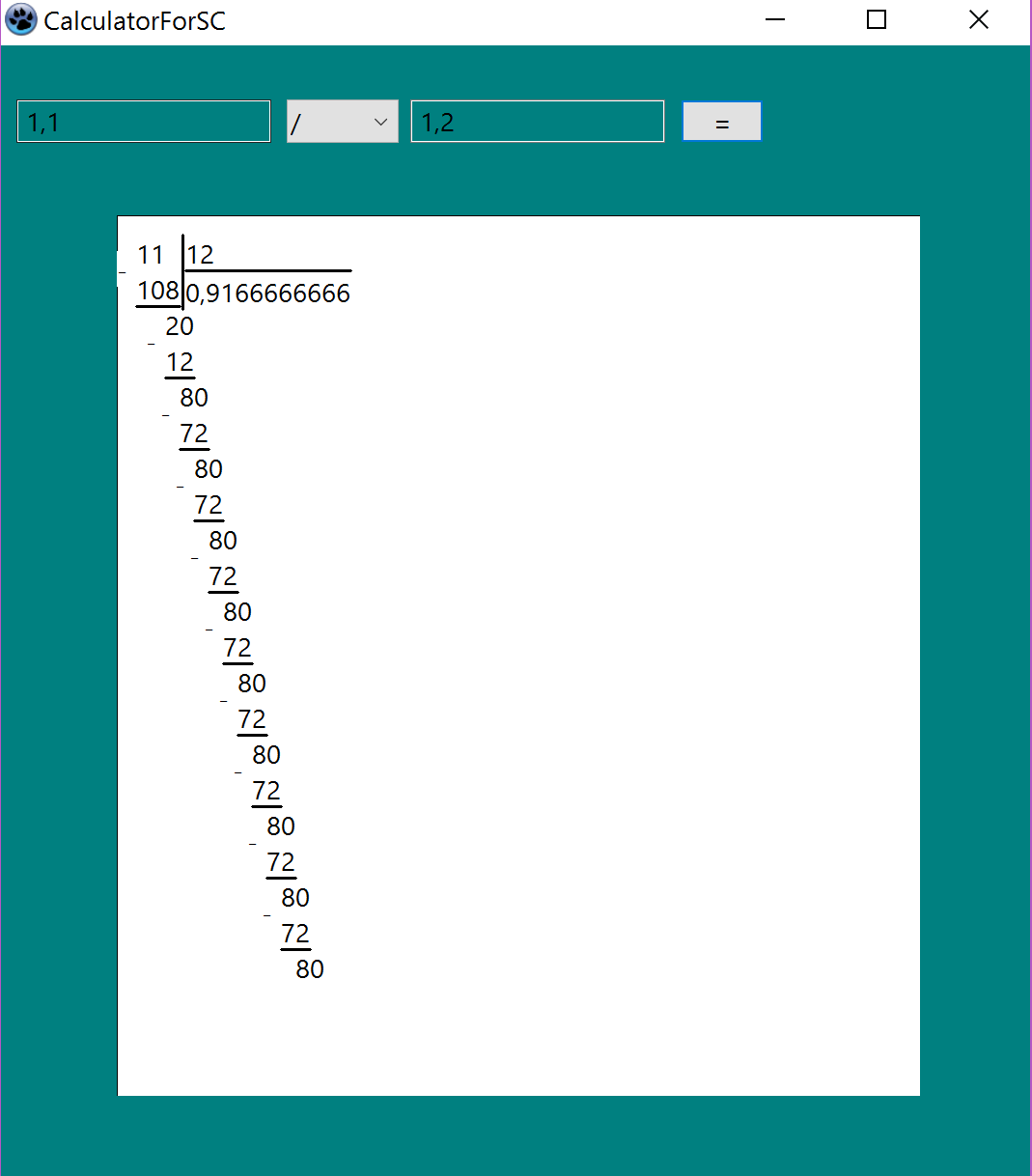


Рисунок – \*\*\*

Деление на ноль было запрещено (Рисунок \*\*\*)

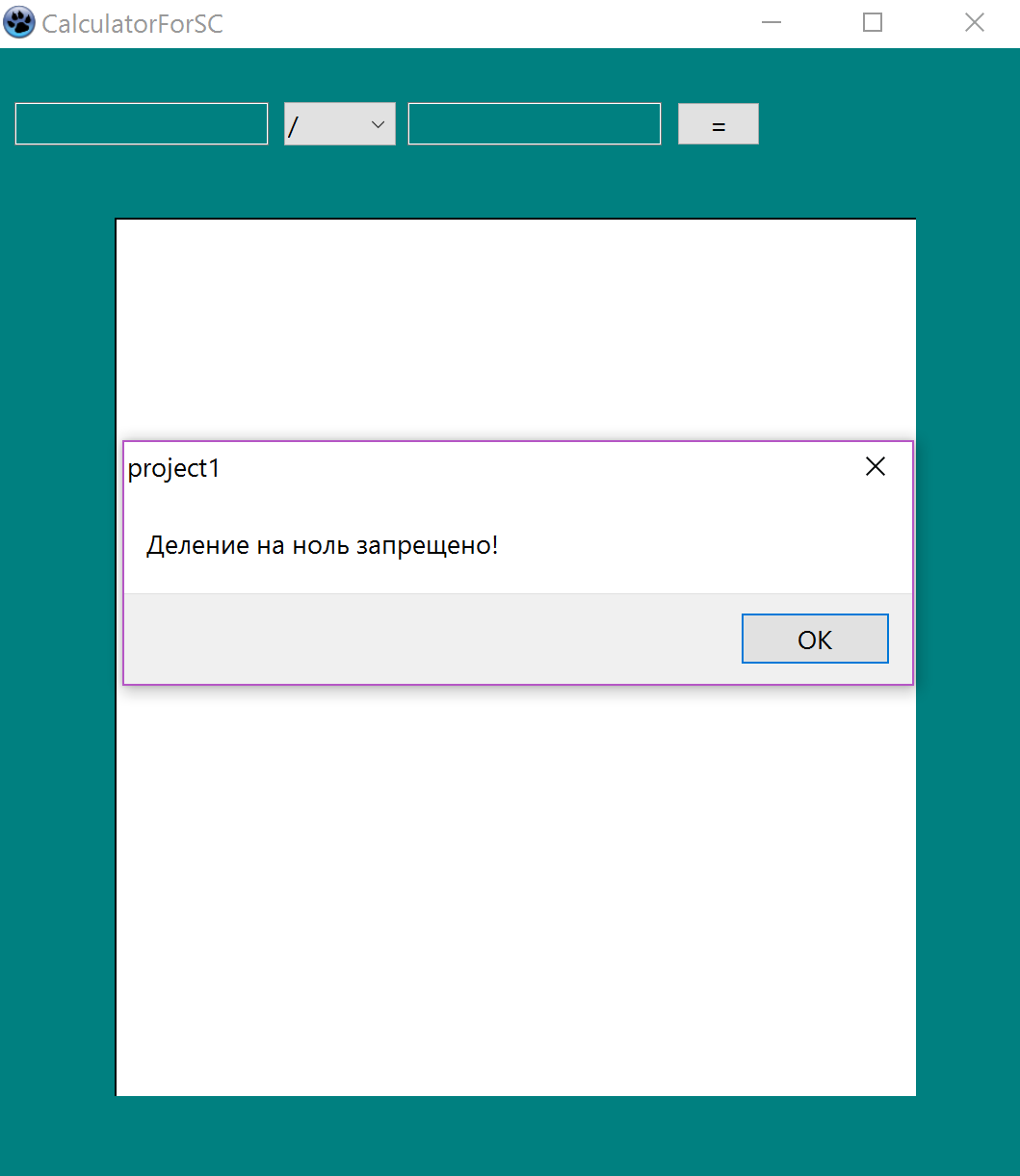


Рисунок – \*\*\*

Умножение обычных чисел работает исправно (Рисунок \*\*\*)

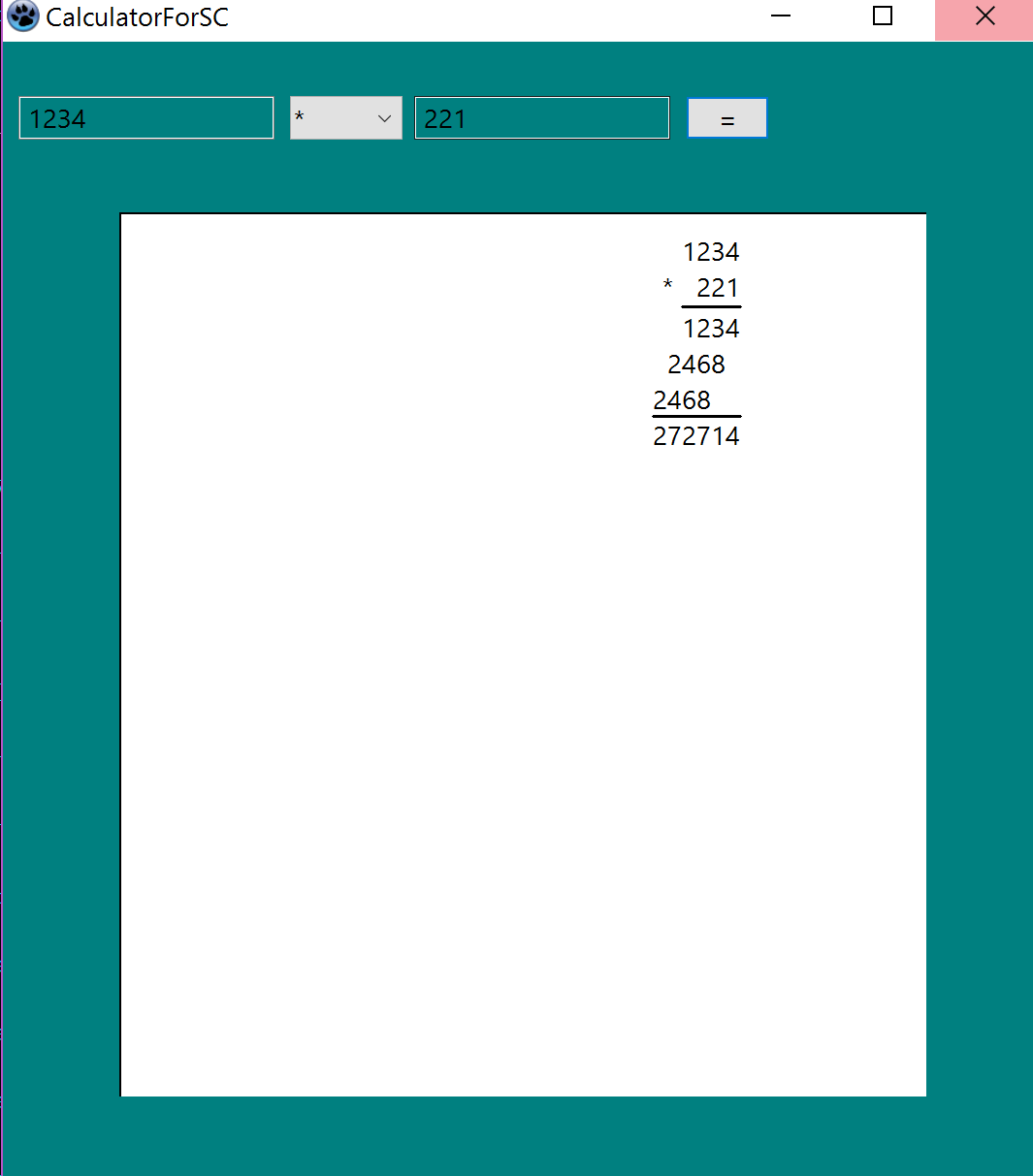


Рисунок – \*\*\*

Умножение дробных чисел работает исправно (Рисунок \*\*\*)

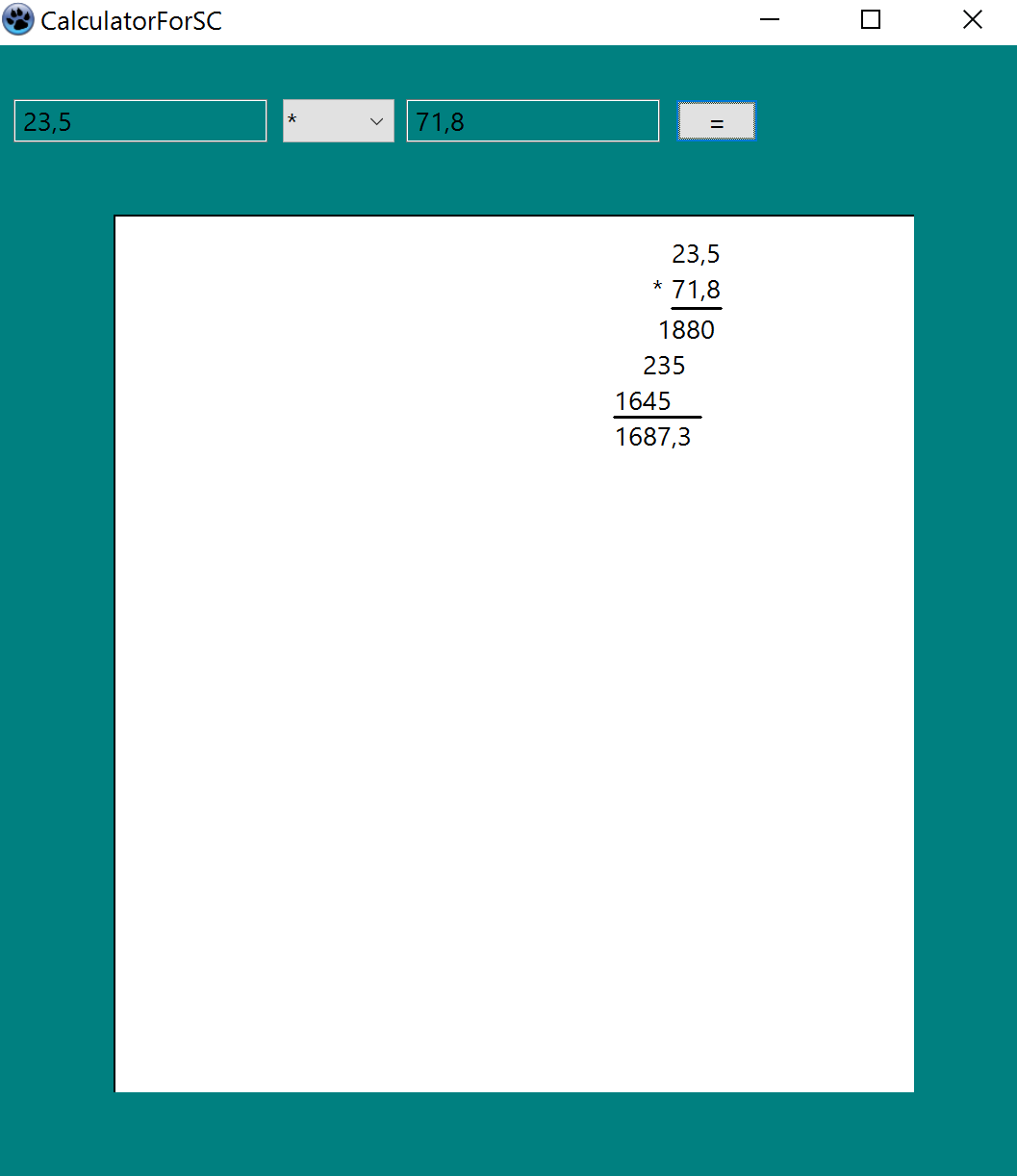


Рисунок – \*\*\*

Вычитание обычных чисел работает исправно (Рисунок \*\*\*)

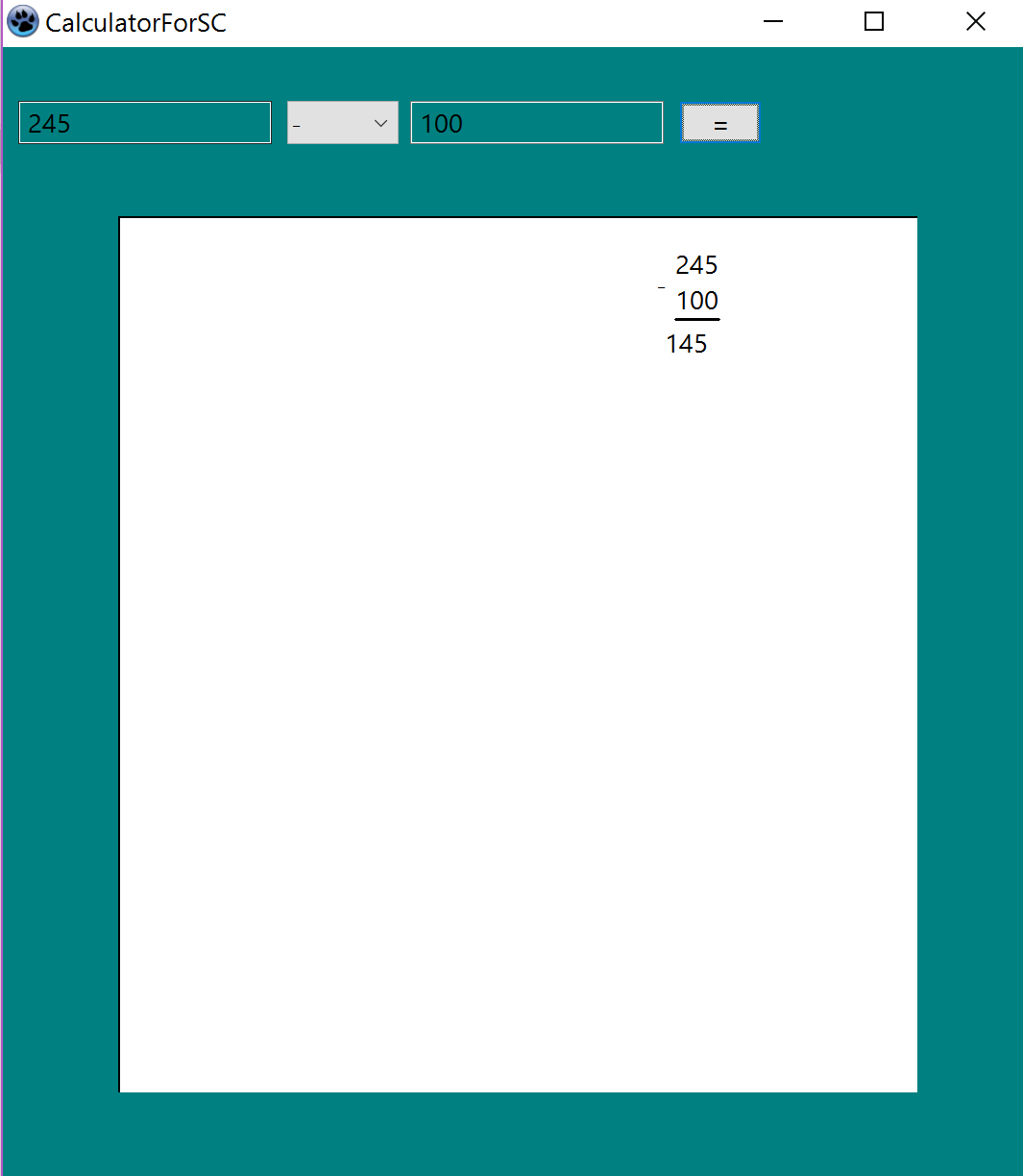


Рисунок – \*\*\* (Вывод на экран работает не совсем корректно)

Вычитание дробных чисел работает исправно (Рисунок \*\*\*)

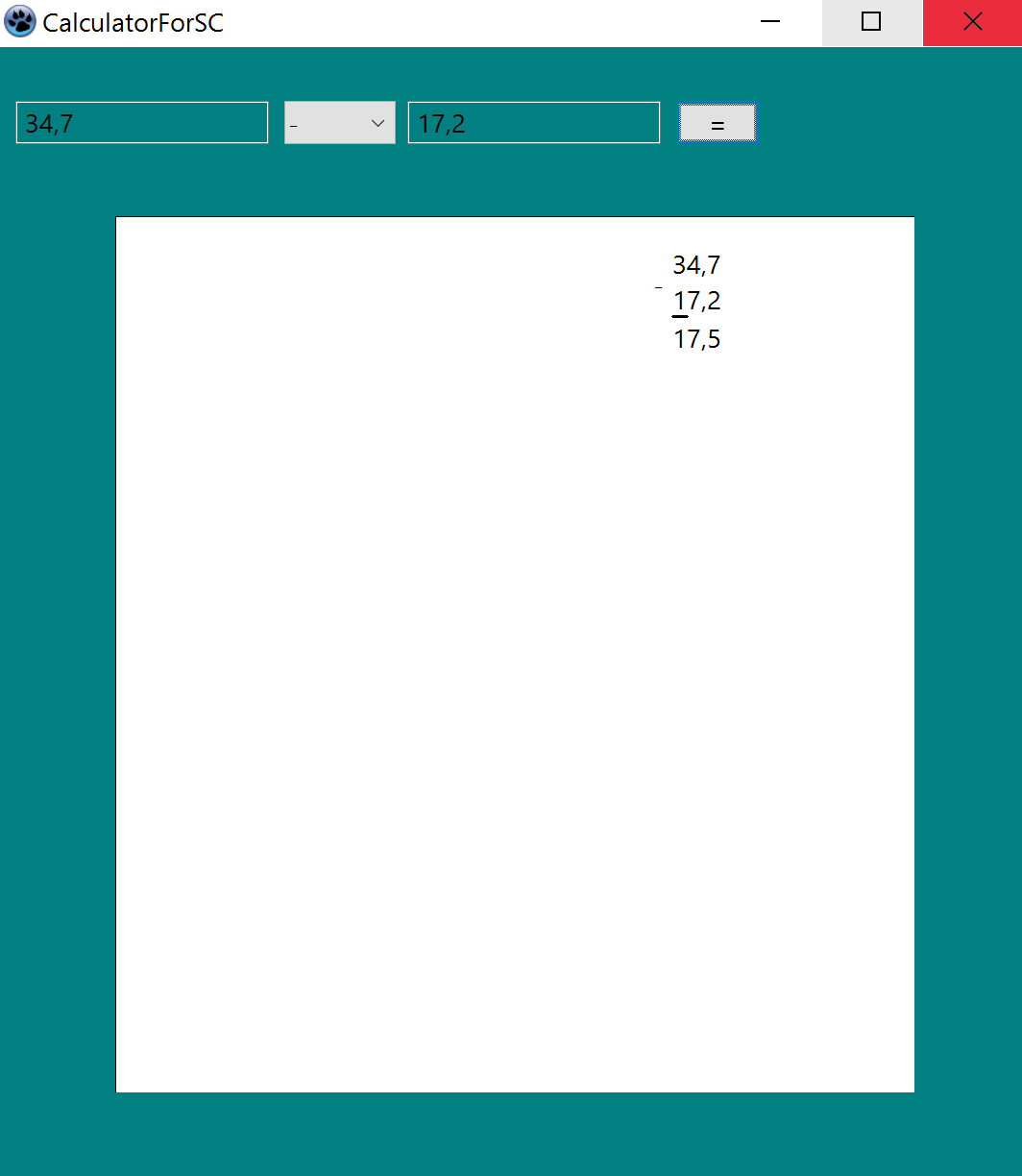


Рисунок – \*\*\* (Вывод на экран работает не совсем корректно)

Сложение обычных чисел работает исправно (Рисунок \*\*\*)

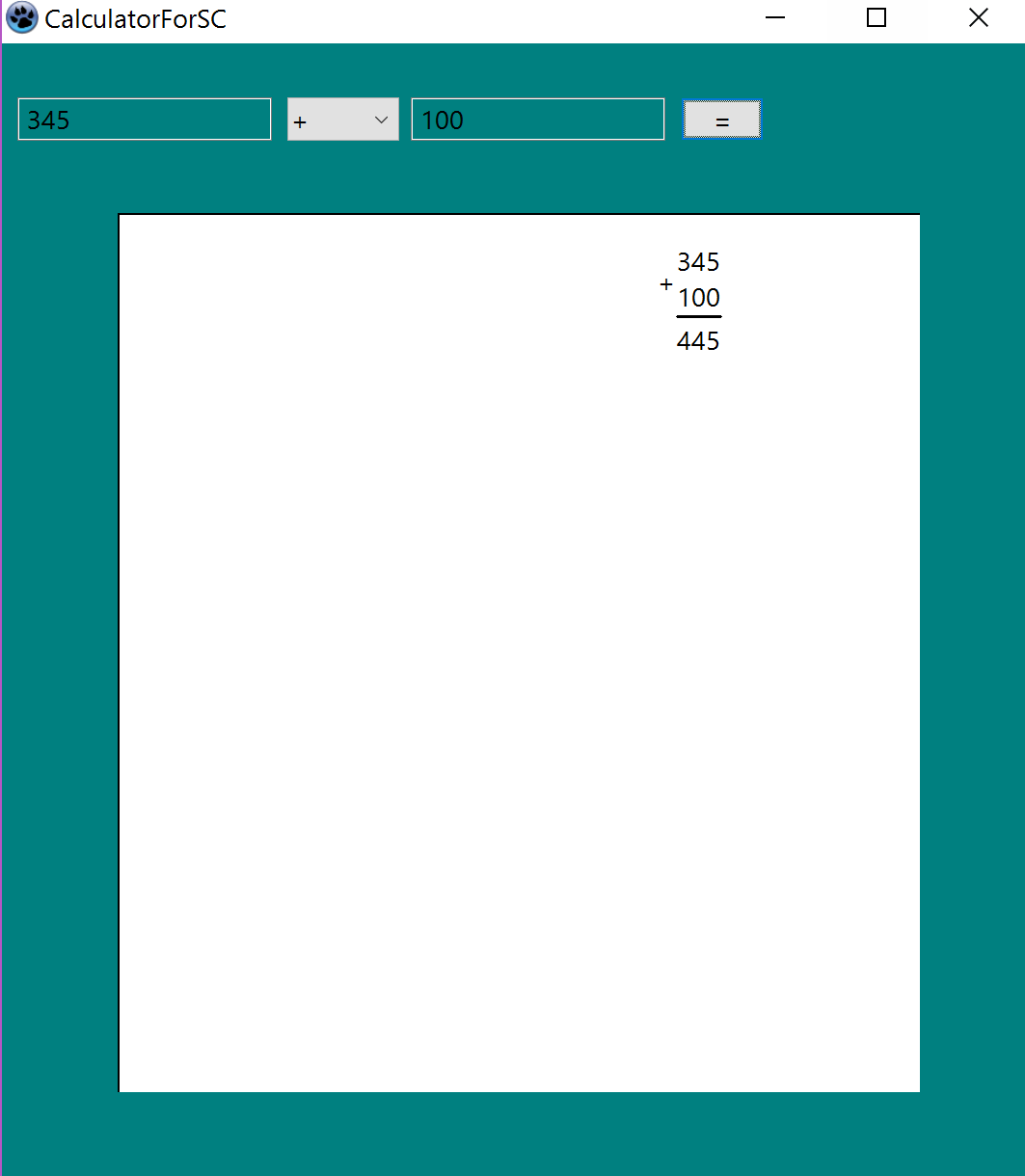


Рисунок – \*\*\*

Сложение обычных чисел работает исправно (Рисунок \*\*\*)

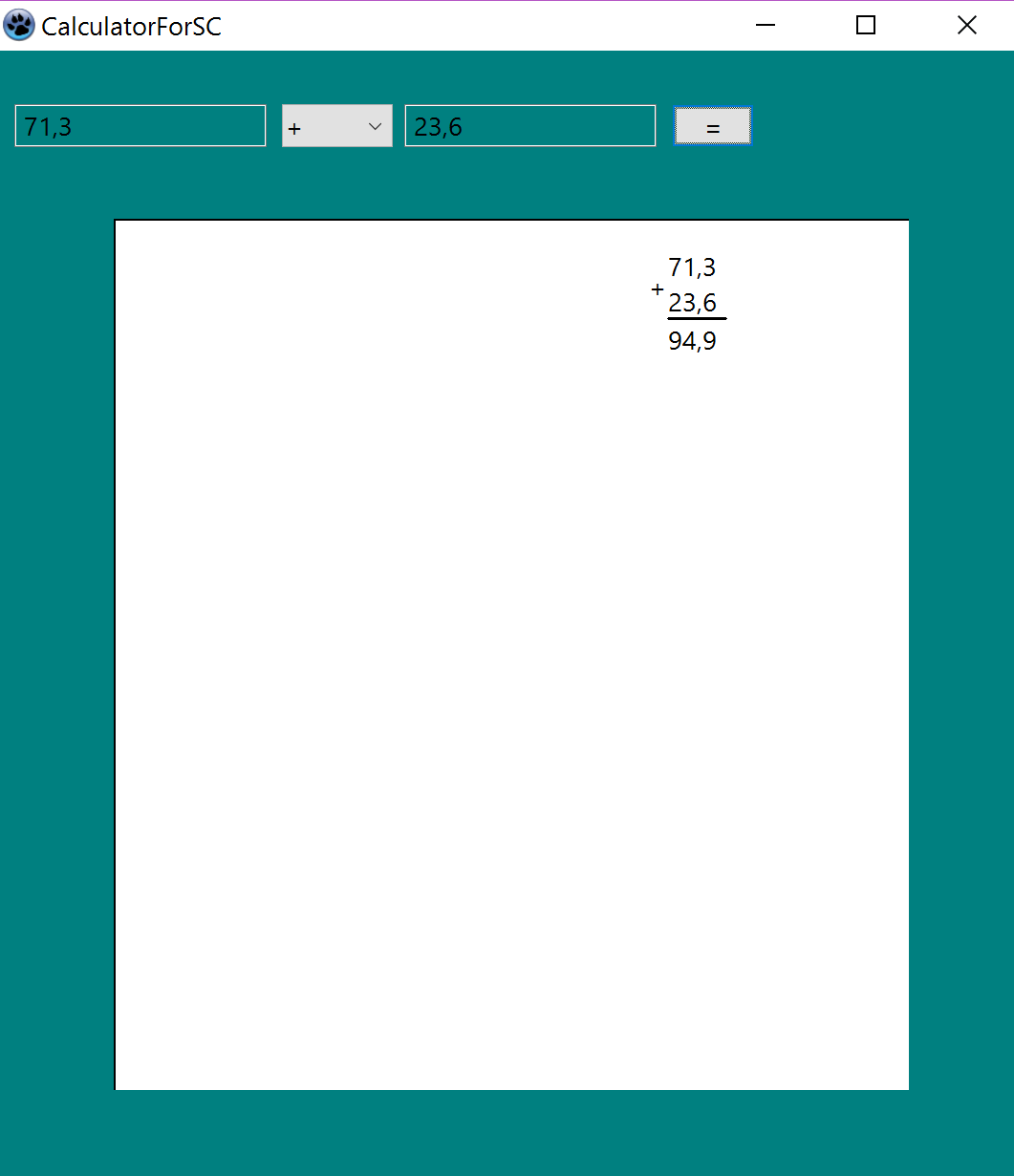


Рисунок – \*\*\*

ЗАКЛЮЧЕНИЕ

Результатом работы оказался разработанная программа «Обучающий калькулятор», который включает в себя следующее:

Достоинства программы «Обучающий калькулятор»:

1. Доступность, так как является некоммерческим проектом;
2. Помогает детям легко обучиться считать на примерах.

Недостатки разработанного игры «Самолетик»:

1. Максимальная простота программы;
2. Вывод вычитания и сложения сделан не лучшим образом.

В итоге, задача оказалась чересчур простой. Ничего нового изучено не было.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1 А. И. Глускер «Сборник задач по курсу “Основы алгоритмизации и программирования” [Электронный ресурс] – 2011/раздел 3.1 – Режим доступа: <https://drive.google.com/drive/folders/0BzMEQjYKTHyOd3Q1ck1kQjFUMFU> (21.12.1017)