|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **РОСЖЕЛДОР**  **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  **ВЫСШЕГО ОБРАЗОВАНИЯ**  **СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  **ПУТЕЙ СООБЩЕНИЯ (СГУПС)**   |  |  |  |  | | --- | --- | --- | --- | | **К защите:** |  |  | | | **Заведующий кафедрой** | **Информационные** | | | **технологии транспорта** | | | |  | д-р техн. наук, проф. | | |  |  | В. И. Хабаров | | | *подпись* |  | *инициалы, фамилия* | | |  |  |  | | | *дата* |  |  | |   **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Тема:** | Разработка обучающей программы с веб-интерфейсом | | | | | | |  | по внутренним форматам простых типов данных | | | | | | |  | |  | БР.БИСТ.12.2022 |  |  | |  | |  | *шифр документа* |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  | **Выполнил** |  |  |  | **Руководитель** | |  |  | А. С. Малдаван |  |  |  | канд. техн. наук, доц.  В. Г. Кобылянский | | *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* | |  |  |  |  |  |  |  | | *дата* |  |  |  | *дата* |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Консультанты по разделам** |  |  |  |  | | 1 |  |  |  |  | |  |  |  |  |  | | 2 |  |  |  |  | |  |  |  |  |  | | 3 |  |  |  |  | |  |  |  |  |  | | Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина | |  |  | *подпись* |  |  | |  |  |  |  |  | |  |  | *дата* |  |  |   **2022 г.** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  **ПУТЕЙ СООБЩЕНИЯ (СГУПС)**  Факультет: Бизнес-информатики  Кафедра: Информационные технологии транспорта  Направление: 09.03.02 «Информационные системы и технологии»  Профиль: Интеллектуальные транспортные системы   |  |  | | --- | --- | |  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров | |  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |   **З А Д А Н И Е**  **на выполнение выпускной квалификационной работы**   |  |  |  |  | | --- | --- | --- | --- | | студенту | Малдаван Александру Сергеевичу | | | |  |  | | | | 1. Тема «Разработка обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных» утверждена приказом № 203/с от «30» мая 2022 г. | | | | | 2. Задание выдано «12» мая 2022 г. | | | | | 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | | | 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | | | 5. Содержание расчетно-пояснительной записки | | | | | Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения | | Введение | | 2 | 13.05.2022 | | Аналитическое исследование | | 8 | 13.05.2022 | | Проектирование информационной системы | | 12 | 15.05.2022 | | Реализация веб-тренажера | | 17 | 28.05.2022 | | Заключение | | 1 | 01.06.2022 |   6. Содержание и объемы графической части   |  |  |  | | --- | --- | --- | | Наименование графического документа (чертежа, схемы, графика) | Количество  слайдов | График  (сроки)  выполнения | | Презентация PowerPoint | 23 | 05.06.2022 |   7. Консультанты по разделам   |  |  |  | | --- | --- | --- | | Наименование  раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания | |  |  |  |  |  |  |  | | --- | --- | --- | | Руководитель |  | В. Г. Кобылянский | |  | *(подпись, фамилия, И.О.)* |  | | Задание к использованию принял |  | А. С. Малдаван | |  | *(подпись студента)* |  | |

УДК 004.588

**АННОТАЦИЯ**

В работе 49 страниц, 26 рисунков, 14 источников.

Ключевые слова: *Веб интерфейс, типы данных, обучение, алгоритмы перевода*.

Предметная область – Веб-тренажер. Цель работы – создать прототип веб тренажера по внутренним форматам простых типов данных являющимся помощником в обучении.

Во время выполнения работы проведен анализ предметной области, описаны бизнес-процессы, обоснован выбор языка и программного обеспечения, рассмотрено поэтапное проектирование программы.

Результатом выполнения работы является программа, разработанная в среде Visual Studio Code, на языке JavaScript с точечным применением языка PHP.

**ABSTRACT**

The work has 49 pages, 26 drawings, 14 sources.

Keywords: *Web interface, data types, training, translation algorithms*.

The subject area is a web simulator. The purpose of the work is to create a prototype of a web simulator using internal formats of simple data types that are an assistant in training.

During the execution of the work, the analysis of the subject area was carried out, business processes were described, the choice of language and software was justified, the phased design of the program was considered.

The result of the work is a program developed in the Visual Studio Code environment, in JavaScript with a point application of the PHP language.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

БД – это поименованная собрание данных, которое отражает состояние объектов и их отношений в рассматриваемой предметной области.

ИС – информационная система, это система, которая предназначена для хранения, обработки и поиска информации, а также персонал, который обеспечивает и распространяет информацию.

ПО – программное обеспечение, это программа или множество программ, которые используются для управления компьютером.

Хостинг – услуга по предоставлению ресурсов для размещения информации на сервере, постоянно имеющем доступ к сети (обычно Интернет).

IDE – Integrated Development Environment, это система программных средств, используемая программистами для разработки программного обеспечения.

**СОДЕРЖАНИЕ**

[Введение 7](#_Toc106057399)

[1 Аналитическое исследование 9](#_Toc106057400)

[1.1 Определение предметной области 9](#_Toc106057401)

[1.2 Постановка целей и задач 9](#_Toc106057402)

[1.3 Анализ бизнес-процессов 10](#_Toc106057403)

[1.4 Выбор среды и языка разработки 12](#_Toc106057404)

[1.5 Вывод по разделу 16](#_Toc106057405)

[2 Проектирование информационной системы 18](#_Toc106057406)

[2.1 Обзор методологий моделирования бизнес-процессов 18](#_Toc106057407)

[2.2 Общие сведения о языке моделирования UML 23](#_Toc106057408)

[2.3 Диаграмма вариантов использования 23](#_Toc106057409)

[2.4 Диаграмма последовательности 24](#_Toc106057410)

[2.5 Диаграмма активностей процесса 25](#_Toc106057411)

[2.6 Диаграмма состояний объекта число 26](#_Toc106057412)

[2.7 Диаграмма обзора взаимодействия 27](#_Toc106057413)

[2.8 Диаграмма классов предметной области 28](#_Toc106057414)

[2.9 Описание структуры базы данных 28](#_Toc106057415)

[2.10 Вывод по разделу 29](#_Toc106057416)

[3 Реализация веб-тренажера 30](#_Toc106057417)

[3.1 Выбор и обоснование типа информационной системы 30](#_Toc106057418)

[3.2 Система управления базами данных 30](#_Toc106057419)

[3.3 Хостинг 30](#_Toc106057420)

[3.4 Модель данных (ER-модель) 30](#_Toc106057421)

[3.5 Проектирование алгоритмов перевода 33](#_Toc106057422)

[3.6 Графический интерфейс веб-тренажера 37](#_Toc106057423)

[3.7 Средства и методы тестирования 41](#_Toc106057424)

[3.8 Развертывание, установка, настройка информационной системы 43](#_Toc106057425)

[3.9 Описание функциональных особенностей информационной системы 43](#_Toc106057426)

[3.10 Руководство администратора 44](#_Toc106057427)

[3.11 Руководство пользователя 44](#_Toc106057428)

[3.12 Условия обеспечения работоспособности информационной системы 46](#_Toc106057429)

[3.13 Результаты реализации веб тренажера 46](#_Toc106057430)

[Заключение 47](#_Toc106057431)

[Список использованных источников 48](#_Toc106057432)

# **ВВЕДЕНИЕ**

В начале изучения программирования, школьников и студентов обучают переводить числа из одной системы счисления в другую. Система счисления определяет алфавит языка, на котором «разговаривает» программа. Люди привыкли использовать в повседневной жизни цифры от нуля до десяти, то есть используют десятичную систему счисления. Однако компьютер – это лишь машина, работающая с набором бит, в то время как в одном бите может храниться одно из двух чисел – ноль или один. Это означает, что для «общения» компьютер использует двоичную систему счисления. Чтобы понимать друг друга, человек или машина должны представлять свои сообщения в системе счисления собеседника.

Переводить числа из десятичной системы счисления в двоичную и обратно – работа, хоть, и требующая определенного времени, но не такая сложная. Однако системы счисления не ограничиваются двумя вышеописанными. Их существует огромное множество, и перевод сообщения из системы с одним основанием в систему с другим основанием становится проблематичным. Еще более проблематичным становится тот факт, что разные числа кодируются по-разному. Так различными способами кодируются:

* знаковые числа – положительные и отрицательные целые числа в заданном диапазоне;
* беззнаковые числа – целые числа от ноля до границы диапазона;
* числа с плавающей точкой – числа, которые хранятся в виде мантиссы и показателя степени;
* числа с фиксированной точкой – дробные числа с нефиксированным количеством знаков до и после запятой;
* символы – отдельные элементы алфавита;
* строки – набор символов.

Для оптимизации процесса уже давно в машинах используют встроенные функции перевода чисел в разные системы счисления. Тем не менее, школьников и студентов все еще обучают навыку перевода чисел, чтобы они понимали, как это работает. Для ускорения обучения и проверки их знаний были разработаны соответствующие калькуляторы.

Для разработки обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных:

* проанализировать предметную область;
* проанализировать программы перевода данных;
* проанализировать требования к обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных;
* проанализировать имеющуюся бета-версию системы;
* выбрать технологию проверки перевода чисел тренажером;
* выбрать способ хранения обращений обратной связи;
* разработать обучающую программы с веб-интерфейсом по внутренним форматам простых типов данных с учетом всех требований;
* проверить соответствие системы всем требованиям;
* протестировать разработанную систему и при необходимости исправить недостатки системы.

В ходе разработки системы будут использованы:

* интегрированная среда разработки программного обеспечения Visual Studio Code;
* инструмент для моделирования бизнес-процессов Visual Paradigm.

При необходимости могут быть использованы и другие инструменты.

# **1 Аналитическое исследование**

## **1.1 Определение предметной области**

В ходе выполнения выпускной квалификационной работы необходимо разработать обучающую программу с веб-интерфейсом по внутренним форматам простых типов данных. Данная система, исходя из названия, обеспечивает перевод чисел во внутренние форматы простых типов данных разных языков программирования. В последствие систему можно адаптировать для других языков программирования, прочих типов данных.

## **1.2 Постановка целей и задач**

Главной целью выполнения выпускной квалификационной работы является создание обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных. Программа должна быть интуитивно понятно конечному пользователю, обладать всем возможным функционалом для успешной интеграции в обучающий процесс, иметь возможность коммуникации между пользователем и разработчиком, для дальнейшей реализации и удовлетворения потребностей пользователя, а также оперативного устранений багов.

Задачи, которые необходимо выполнить при реализации обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных:

* создание простого и приятного интерфейса;
* наличие руководства пользователя;
* реализация алгоритмов перевода числа в выбранный тип данных, в двоичную и шестнадцатеричную системы счисления;
* разработка формы обратной связи;
* наличие проверок на адекватность введенных данных.

## **1.3 Анализ бизнес-процессов**

Обучающая программа с веб-интерфейсом по внутренним форматам простых типов данных работает по следующему алгоритму [1]:

1. Происходит подключение к серверу разработчиков по информационно-коммуникационной сети Интернет.
2. Пользователь выбирает необходимую ему страницу из всех доступных на сайте.
3. Пользователь переходит на страницу перевода данных во внутренний формат языков C# или Pascal, выбирает режим работы «Обучение» или «Контроль».

При выборе режима «Обучения», тренажер начинает работу по следующему алгоритму:

1. Происходит проверка выбора (Был ли выбран тип данных).
2. Происходит проверка адекватности введенного числа, соответствует ли оно выбранному типу данных и находится ли оно в допустимом диапазоне.
3. Если при вводе пользователем числа не произошло ошибки, тренажер начинает перевод числа пользователя в выбранный тип данных.
4. После перевода в выбранный тип данных, тренажер преобразует значение из десятичной системы счисления в двоичную и шестнадцатеричную.
5. Полученные в ходе перевода значения тренажер выводит в поля вывода данных.

В случае появления ошибки на первом пункте, тренажер выводит ошибку.

Если первая проверка прошла успешно, то за ней следует вторая, в случае ошибке на втором пункте алгоритма, программа выведет ошибку.

Если все проверки были успешно пройдены, тренажер начинает переводить выбранное число во внутренний формат языка C# или Pascal.

Результат перевода числа выводится в заранее подготовленные для этого поля.

На страницах перевода чисел также присутствует режим «Контроль», который предназначен для проверки знаний пользователя. При выборе пользователем этого режима, тренажер действует согласно следующему алгоритму:

1. Пользователь нажимает кнопку «Сгенерировать» и в случае, если не выбран тип данных, программа выбрасывает ошибку.
2. Программа задает случайное число, в диапазоне, соответствующем выбранному типу данных и конвертирует это число в двоичное и шестнадцатеричное представление.
3. Пользователь вводит высчитанное им число в поле ввода.
4. Программа проверяет правильность введенного пользователем числа.

Если ошибка возникает на первом этапе алгоритма, то программа выводит ошибку. В случае ошибки пользователем, программа выдает исключение и в скобках выведет правильный результат.

Также функционал тренажера позволяет оставить свои пожелания или сообщит о найденном баге посредством отправки письма. Веб-тренажеру необходимо централизованное средство коммуникации, так как если оставить обычные контакты разработчика, то сообщение может затеряться и пожелание не будет реализовано. Если пользователь находится на странице «Обратной связи», то программа действует согласно следующему алгоритму:

1. Пользователь вводит свое имя, Email и сообщения для разработчика.
2. Программа проверяет заполнены ли все поля на форме.
3. Отправляет письмо на почту, прикрепленную к веб-тренажеру.

Ошибки могут возникнуть на втором этапе алгоритма отправки письма, если пользователь не заполнил какое-то из полей или заполнил его некорректно, то программа выдаст исключение.

## **1.4 Выбор среды и языка разработки**

Для реализации выпускной квалификационной работы был выбран язык программирования JavaScript. Выбор обусловлен требованиями технического задания.

Язык программирования JavaScript – современный объектно-ориентированный и типобезопасный язык программирования. JavaScript позволяет разработчикам создавать разные типы безопасных и надежных веб приложений.

Достоинства языка JavaSript:

* ни один современный браузер не обходится без поддержки JavaScript.
* с использованием написанных на JavaScript плагинов и скриптов справится даже не специалист;
* полезные функциональные настройки;
* постоянно совершенствующийся язык – сейчас разрабатывается бета-вариация проекта, JavaScript2;
* взаимодействие с приложением может осуществляется даже через текстовые редакторы – Microsoft Office и Open Office;
* перспектива использования языка в процессе обучения программированию и информатике.

В интернете распространяется огромное количество сред разработки (IDE) для разработки на языке JavaSript, число которых в дальнейшем будет только увеличиваться. Для выбора наиболее подходящей IDE необходимо сравнить их достоинства и недостатки с требованиями к IDE, описанными ранее [2].

Первой на рассмотрении будет среда разработки Visual Studio. Описание данной IDE гласит, что она является самой «правильной» средой разработки. С Visual Studio многие начинают знакомиться с языком программирования JavaScript и не расстаются с ней на протяжении всей карьеры программиста.

Среда разработки Visual Studio является:

* официальной – язык и среда разработки созданы в Microsoft. При использовании технологий UWP и WPF без Visual Studio не обойтись;
* бесплатной – версии «Community edition» для рядового пользователя вполне достаточно;
* функциональной – в Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки;
* мультиплатформенной – поддерживает платформы .NET. Visual Studio, имеет широкие возможности по разработке приложений под Windows, в том числе в .NET-сегменте;
* удобной в хранении – обладает облачными хранилищами;
* корпоративной – технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

В то же время Visual Studio обладает некоторыми недостатками:

* при переходе на платную версию могут теряться настройки и нарушаться работа корпоративного сервера;
* сложность – самостоятельно освоить Visual Studio будет непросто, так как. она обладает слишком большим количеством доступных функций.

Еще одной достойной средой разработки является Project Rider – среда разработки от JetBrains для работы с платформой .NET.

Project Rider обладает рядом достоинств:

* ReSharper – это плагин, изначально разработанный для повышения производительности Visual Studio. Теперь на его основе выпущена IDE;
* поддержка полного цикла – фирменная черта продуктов JetBrains, воплощенная и в Project Rider. С ним можно организовать весь цикл создания ПО: от идеи до поддержки;
* функциональность – Project Rider позволяет подключить MSBuild и XBuild, работать с CLI-проектами и организовать отладку приложений .NET and Mono;
* поддержка нескольких запущенных программ;
* кроссплатформенность;
* контроль версий – встроенный инструмент позволяет напрямую организовать работу с Git, Mercurial и TFS.

К недостаткам Project Red относят:

* молодость – часть функциональности еще в разработке, не все стартовые баги исправлены;
* стоимость – самая дешевая версия Project Rider обойдется в 139 долларов за первый год использования. Но есть триал-версия и специальные предложения для студентов и непрофильных организаций.

Третья IDE в списке на выбор среды разработки ВКР – Eclipse. Eclipse – одна из самых популярных мультиязычных сред, ориентированная на разработку Java-приложений, но полезна и для кодов на JavaScript.

Плюсы Eclipse:

* множество плагинов. У Eclipse едва ли не самое большое число надстроек – «на все случаи жизни»;
* активное сообщество, которое помогает быстрее освоить среду разработки и выпускает новые плагины;
* отличные компилятор и отладчик. Первый работает на порядок быстрее, чем у конкурентов, второй – показывает потоки, пересечения, позволяет гибко управлять ходом отладки;
* кастомизация – благодаря плагинам и настройкам можно полностью персонализировать Eclipse;
* это open-source проект, абсолютно бесплатный;
* высокая функциональность. Благодаря разработчикам-официалам и членам сообщества с помощью Eclipse можно провести любой JavaScript-продукт по полному циклу разработки.

Также Eclipse обладает некоторыми минусами:

* сложность – как и любой функциональный продукт, Eclipse может показаться новичку слишком сложным;
* нет гарантий надежности. Плагины создаются сообществом и за их качество отвечает только разработчик. Кроме того, сами создатели Eclipse с каждой новой версией плодят баги, не успевая порой исправлять старые.

Следующая IDE является «родственником» среды разработки Visual Studio. Она также разработана компанией Microsoft и обладает схожим названием – Visual Studio Code. Данное приложение является кроссплатформенным редактором кода, которым при помощи плагинов можно «подтянуть» к статусу IDE.

Среда разработки Visual Studio Code является:

* кроссплатформенной – работает на MacOS, Ubuntu и Windows;
* бесплатной;
* легковесной – потребуется совсем мало ресурсов, чтобы приступить к работе с минималистичным Visual Studio Code.

В то же время Visual Studio Code обладает некоторыми недостатками:

* низкая функциональность – несмотря на поддержку .NET-платформы, Visual Studio Code неудобен для сложных проектов;
* сомнительная надежность – многие надстройки имеют низкое качество сборки и не всегда выполняют даже основные функции.

MonoDevelop – еще одна из рассматриваемых IDE. Она является свободной средой разработки от Xamarin для создания приложений на множестве языков, в том числе на JavaScript.

Достоинства MonoDevelop:

* мультиплатформенность – поддержка Linux, Windows и Mac OS;
* кастомизация – на рабочем столе можно расположить функции и окна по своему усмотрению;
* Unity 3D. Полноценная поддержка популярной платформы для разработки игр;
* бесплатность.

Недостатком MonoDevelop является ограниченная функциональность. У MonoDevelop есть собственный отладчик и инструменты для работы с кодом, но в части поддержки разных платформ и проектов – это пока сырая IDE.

Наконец, рассмотрим IDE Code:Blocks – среду разработки, известную простой и удобной в настройке и использовании.

Достоинства Code:Blocks:

* бесплатность;
* простота;
* кроссплатформенность;
* возможность выбора компилятора;
* легковесность.

Минусы Code:Blocks:

* недостаточная функциональность – для создания комплексных приложений Code:Blocks категорически не подходит.

Нестабильность IDE – приходится сталкиваться с нелепыми ошибками в отладке и некорректной работой всей среды.

В результате выбора среды разработки:

* MonoDevelop и Code:Blocks не подходят, так как обладают недостаточным функционалом;
* Project Rider является платным продуктом;
* Visual Studio и Eclipse являются сложными в освоении.

Самой подходящей средой разработки для написания технической составляющей выпускной квалификационной работы является Visual Studio Сode. Данная среда разработки соответствует всем требованиям, изложенным в техническом задании и обладает всем набором необходимых плагинов, которые могут быть полезны при сетевой разработке на языке JavaScript/

## **1.5 Вывод по разделу**

В данном разделе было проведено аналитическое исследование, в котором была определена предметная область выпускной квалификационной работы. Были поставлены основные цели и задачи, которые необходимо решить по мере разработки программы. Также были проанализированы основные бизнес процессы, которые будут присутствовать в программе и как следствие расписаны подробные алгоритмы каждого из основных бизнес процессов. Завершающим этапа раздела аналитического исследования был выбор наиболее подходящего языка программирования, функционал которого позволил бы реализовать все потребности разработчика и пользователя, а также была выбрана наиболее подходящая среда разработки под выбранный язык программирования.

# **2 Проектирование информационной системы**

## **2.1 Обзор методологий моделирования бизнес-процессов**

Для реализации любого вида программного обеспечения необходимо провести моделирование бизнес-процессов, которые будут происходить в ходе его работы. Существует большое количество методологий, по которым производят моделирование процессов. Ниже приведено описание наиболее популярных из них [3].

*Методология SADT.*

Методология SADT, исходя из названия, описывает технологию структурного анализа и проектирования. В основе методологии лежит зависимость сущностей друг от друга.

SADT позволяет построить функциональную модель объектов предметной области, позволяющую проследить за взаимодействиями объекта с другими сущностями предметной области, а также за результатами, возникающими в ходе этого взаимодействия.

Любая методология моделирования-бизнес процессов обладает рядом характеризующих ее правил. Моделирование бизнес-процессов с помощью методологии SADT происходит с помощью представления модели в форме блочных графиков. Диаграмма состоит из блоков (взаимодействия объекта с окружающей его средой) и входных и выходных дуг (входы в процесс и выходы из него соответственно). Выходная дуга из одного блока может оказаться входной дугой другого блока, что описывает причинно-следственную связь между процессами.

Диаграмма SADT обладает строгими правилами оформления, которых следует придерживаться. В то же время эти правила не ограничивают работу системного аналитика. С помощью SADT можно в полной мере описать объект в его предметной области. Правила построения диаграмм в методологии SADT:

* следует использовать все преимущества декомпозиции и не представлять на одной диаграмме слишком большое количество блоков. Обычно используется около пяти блоков на диаграмме. В редких случаях приемлемым считается количество блоков близкое к десяти;
* необходимо придерживаться нумерации блоков, чтобы исключить недопонимания, связанные с порядком выполнения процессов. Особенно это правило касается диаграмм, в которых входные и/или выходные дуги пересекаются;
* использовать уникальные номера блоков, названий процессов и входных, и выходных переменных процесса;
* употреблять корректное наименование блоков и дуг. В блоках с помощью отглагольного существительного в настоящем времени описывается процесс выполнения какого-либо действия. Дуги являются сущностями, которые будут использоваться в процессе, поэтому их название должно быть записано существительным в единственном числе;
* различать входные переменные от управлений и механизмов. Входная переменная представляет собой задачу, которую необходимо выполнить в ходе процесса. Под управлением понимается инструмент выполнения процесса. Механизм – вспомогательный элемент для выполнения процесса;
* нельзя учитывать влияние административной модели на функциональную.

*Методология IDEF0.*

Неудивительным является то, что внешне диаграммы, построенные с помощью методологии IDEF0, несколько похожи на диаграммы, реализованные с помощью методологии SADT, ведь методология IDEF0 является подмножеством методологии SADT.

IDEF0 также, как и SADT, позволяет простроить функциональную модель описания бизнес-процессов предметной области. Различие заключается в том, что IDEF0 более детально останавливается на взаимодействии объектов между друг другом, нежели на последовательности действий отдельных объектов.

Огромное значение имеет местоположение блока функции на диаграмме – блоки располагаются по диагонали сверху вниз слева направо по наибольшему влиянию на моделирование описываемого процесса. В то же время IDEF0 придерживается строгого правила расположения стрелок для блоков: входы располагаются слева, управления – сверху, механизмы – снизу, а выходы – справа.

Моделирование происходит по принципу «черного ящика» – известно состояние объекта до какого-либо процесса и после него, но сам процесс скрыт от глаз читателя диаграммы. Более подробно процессы раскрываются в диаграммах декомпозиции.

На диаграмме IDEF0, в отличие от диаграммы потоков данных DFD, отображаются сигналы управления. Описанная методология используется для моделирования всех процессов.

*Диаграмма потоков данных DFD.*

DFD в дословном переводе с английского языка является диаграммой потоков данных. DFD акцентирует внимание на передаче данных между процессами, способами ее обработки и хранения. Диаграммы поток данных активно используются при разработке программных продуктов. Еще одной сферой использования данной нотации является анализ системы в рамках документооборота. Методология наглядно представляет:

* места хранения данных;
* процесс обмена документации;
* ошибки организации бизнес-процессов и прочее.

DFD состоит из:

* процесса – одно или несколько действий, в ходе которых происходит обработка данных. Название блока описания процесса начинается с глагола в инфинитиве;
* внешних сущностей – объектов, которые не входят в моделируемую систему, но привносят в нее новые данные, либо наоборот получают данные из этой системы;
* хранилища данных – любой способ хранения данных (например, базы данных или таблицы);
* потоков данных, которые отображается в виде стрелок, входящую и исходящую информацию из процессов на диаграмме.

DFD не описывает бизнес-процесс, так как не учитывает времени выполнения этого процесса, а также условий, при которых процесс может пойти по альтернативному «сценарию».

*Методология UML.*

Универсальный язык моделирования (UML) является одной из самых популярных методологий для графического описания бизнес-процесс. Большинство разработчиков программного обеспечения используют его при проектировании и сопровождении своих продуктов.

UML состоит из интегрированного набора диаграмм, разработанных для помощи разработчикам систем и программного обеспечении в определении, визуализации, конструировании и документировании результатов разработки программного обеспечения, а также для бизнес-моделирования.

С помощью методологии UML можно построить:

* диаграмму вариантов использования. Данная диаграмма описывает функционал бизнес-процесса, доступный различным объектам предметной области;
* диаграмму последовательности – описывает жизненный цикл объекта предметной области;
* диаграмму состояний – описывает изменение объектов системы при выполнении определенных условий;
* диаграмму классов – описывает все объекты бизнес-процесса, их атрибуты и взаимодействия друг с другом;
* диаграмму активности – моделирует переходы между действиями и показывает взаимодействия между объектами предметной области;
* диаграмму обзора взаимодействия – иллюстрирует как набор фрагментов взаимодействия может быть инициирован в различных сценариях;
* другие диаграммы.

*Методология ARIS.*

Методология ARIS представляет собой набор модулей, использующийся для рассмотрения бизнес-процессов сразу с нескольких точек зрения:

* организационной;
* функциональной;
* информационной;
* обрабатываемых данных;
* структуры бизнес-процессов;
* продуктов и услуг.

Нотация ARIS также, как и другие методологии, может быть использована при разработке и реализации информационных систем. Наиболее популярными в данной сфере являются следующие диаграммы ARIS:

* eEPC – позволяет детально рассмотреть последовательность действий бизнес-процесса, зависящих от произошедших событий;
* FAD – расширяет диаграмму eEPC, включая в нее представления входных и выходных данных функций бизнес-процесса;
* OC – иллюстрирует организационную структуру компании или бизнес-процесса.

Все диаграммы, которые можно построить в нотации ARIS распространяются в программном обеспечении ARIS Express, использующий также модуль составления отчетов для вывода краткой, но от того не менее полной информации, описанной в диаграмме.

## **2.2 Общие сведения о языке моделирования UML**

UML – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML–моделей возможна генерация кода.

В ходе анализа методологий моделирования бизнес-процессов и бизнес-процессов, происходящих у прототипа Telegram бота «помощника по изучению Английского языка», было принято решение использовать методологию объектного проектирования и анализа на языке UML для моделирования бизнес-процессов системы.

Для моделирования бизнес-процессов в UML необходимо построить следующие диаграммы:

* диаграмму вариантов использования системы;
* диаграмму последовательности онлайн тренажера;
* диаграмму состояний объекта «Число»;
* диаграмму активностей онлайн тренажера;
* диаграмму обзора взаимодействия онлайн тренажера;
* диаграмму классов предметной области.

## **2.3 Диаграмма вариантов использования**

В первую очередь построена диаграмма вариантов использования (Use Case Diagram), так как именно она является исходным концептуальным представлением системы в процессе ее проектирования и разработки.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества актеров, взаимодействующих с системой с помощью, так называемых вариантов использования. Полученная диаграмма вариантов использования представлена на рисунке 2.1.



Рисунок 2.1 – Диаграмма вариантов использования системы

## **2.4 Диаграмма последовательности**

Затем создана диаграмма последовательности (Sequence Diagram) - диаграмма, которая служит для представления взаимодействия элементов модели в форме последовательности сообщений и соответствующих событий на линиях жизни объектов. Данные диаграммы фокусируются на времени и визуально показывают порядок взаимодействия, используя вертикальную ось диаграммы для представления времени, когда отправляются сообщения и ответ на эти сообщения. Полученная диаграмма последовательности представлена на рисунке 2.2.

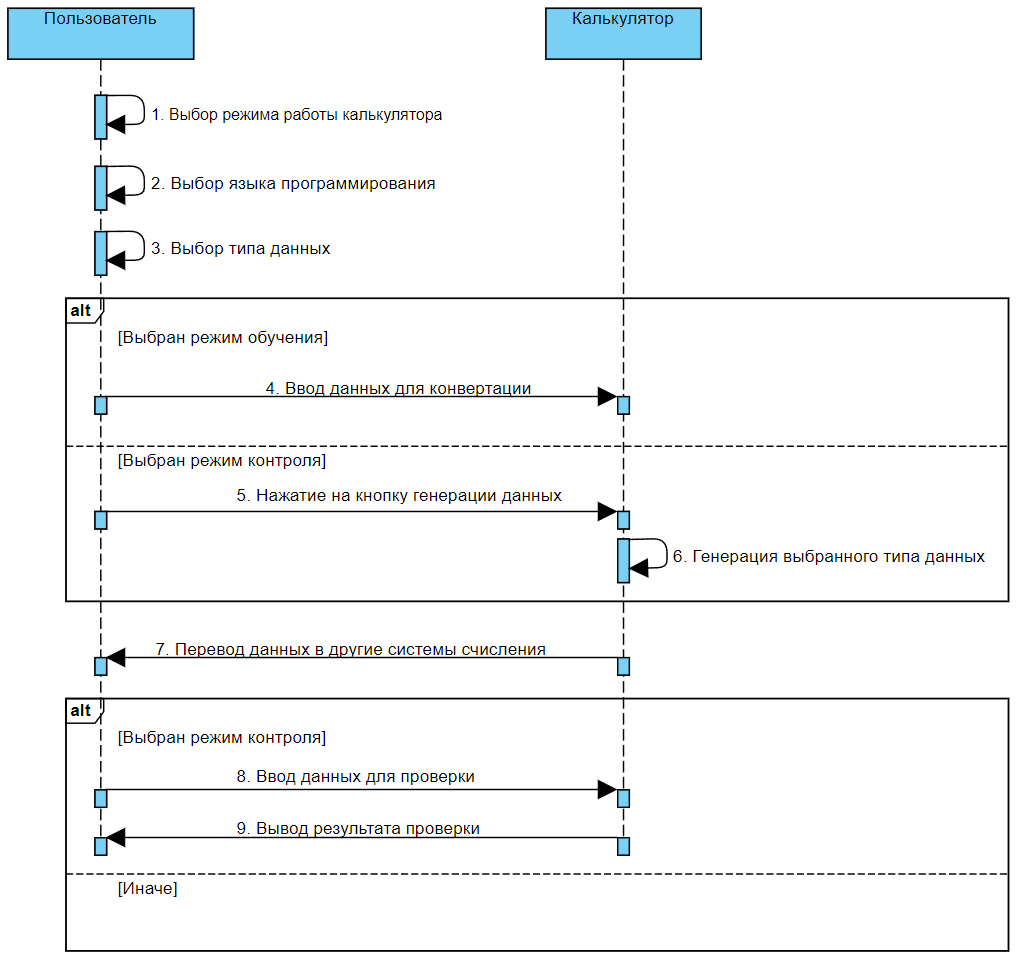


Рисунок 2.2 – Диаграмма последовательности процесса работы онлайн тренажера

## **2.5 Диаграмма активностей процесса**

Далее построена диаграмма активности (Activity diagram), которая используется для построения более «продвинутых» блок-схем, в этом случае часто применяются конструкции принятия решения, а также разделения и слияния потоков управления. Полученная диаграмма активностей представлена на рисунке 2.3.

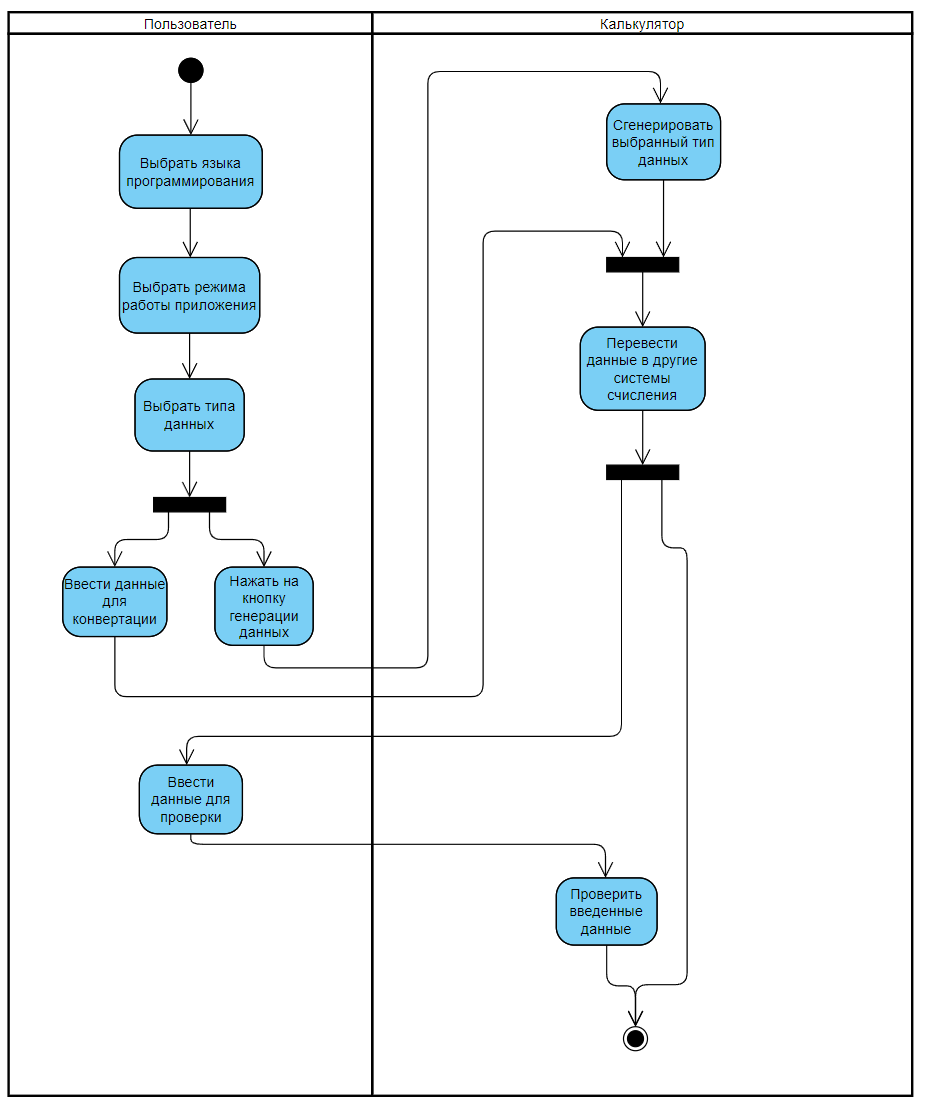


Рисунок 2.3 – Диаграмма активностей процесса работы онлайн тренажера

## **2.6 Диаграмма состояний объекта число**

Диаграмма состояний покажет нам все возможные состояния, в которых может находиться объект, а также процесс смены состояний в результате внешнего влияния. Диаграмма представлена на рисунке 2.4.

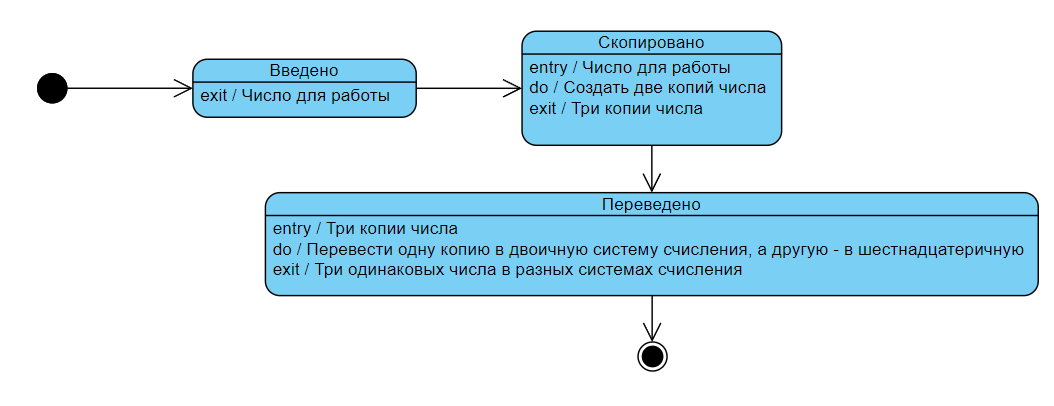


Рисунок 2.4 – Диаграмма состояний объекта «Число»

## **2.7 Диаграмма обзора взаимодействия**

Диаграмма обзора взаимодействия – сочетание диаграммы деятельности и диаграммы последовательности. Диаграмма изображена на рисунке 2.5.

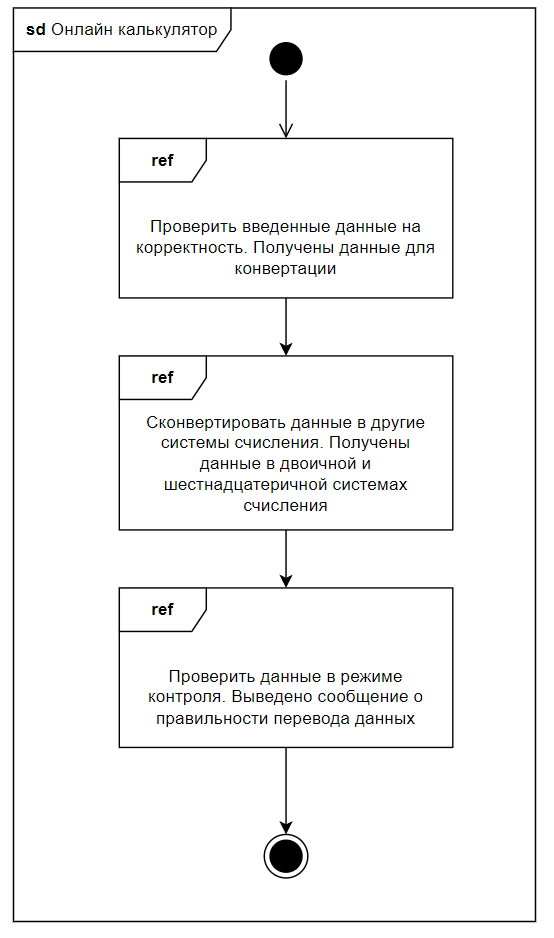


Рисунок 2.5 – Диаграмма обзора взаимодействия онлайн тренажера

## **2.8 Диаграмма классов предметной области**

Диаграмма классов (англ. Class diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Диаграмма изображена на рисунке 2.6.

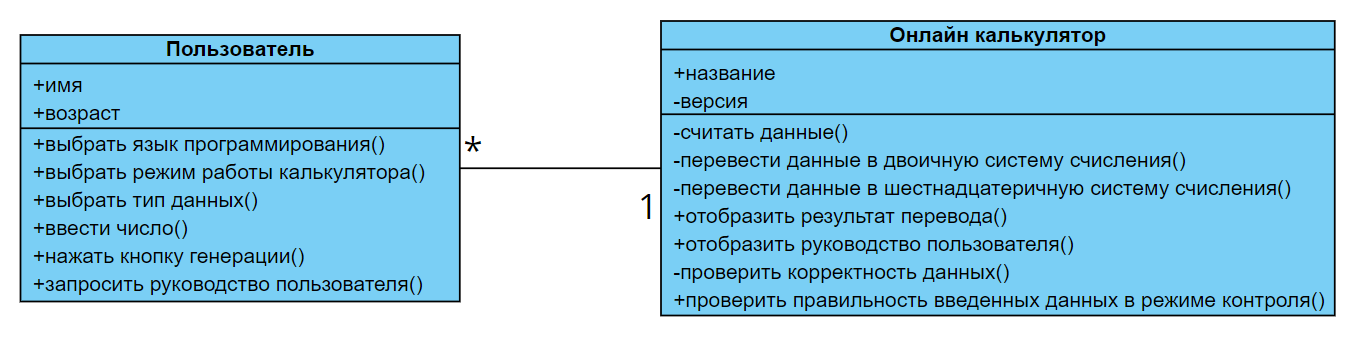


Рисунок 2.6 – Диаграмма классов предметной области

## **2.9 Описание структуры базы данных**

Проанализировав техническое задание к обучающей веб программе по внутренним форматам простых типов данных, можно сделать вывод, что данная система не нуждается в базе данных.

Данная система работает с числами и введенными пользователям, данный элемент программы не нуждается в хранении в БД. Однако в программе присутствует форма обратной связи, которая отвечает за работу с обращениями пользователей, но данный элемент тоже не нуждается в хранении, так как все обращения уходят на привязанную к сайту почту.

При моделировании бизнес-процессов было принято решение разработать обучающую программу с веб-интерфейсом по внутренним форматам простых типов данных в виде набора программных блоков, где каждый блок отвечает за свою страницу. Данные блоки могут быть реализованы в качестве вызываемых файлов на html странице.

## **2.10 Вывод по разделу**

В ходе моделирования бизнес-процессов предметной области были приведены:

* наименование и область применения разрабатываемой системы;
* обзор методологий моделирования бизнес-процессов;
* анализ бизнес-процессов и выбор методологии моделирования;
* моделирование бизнес-процессов.

Для удобства проектирования информационной системы было принято решение разделить систему на несколько функциональных блоков, которые будут вызываться на каждой странице самостоятельно:

* блок проверки и перевода чисел из десятичного формата во внутренние форматы языка C#;
* блок проверки и перевода чисел из десятичного формата во внутренние форматы языка C#;
* блок отправки сообщений.

Все три блока находятся в вызываемых файлах сайта, который в свою очередь будет располагаться на хостинге. Для возможности взаимодействовать с программой у пользователя должен быть Интернет на компьютере, а у админа доступ к панели администратора сайта.

Было принято решение минимизировать взаимодействие пользователей веб тренажера с вызываемыми ошибками системы во избежание некорректной работы, поэтому все процессы необходимо автоматизировать. Пользователь лишь будет вводить число, которое необходимо перевести и получать уведомления о результате перевода. Более подробная информацию об ошибках можно будет посмотреть в консоли браузера.

# **3 Реализация веб-тренажера**

## **3.1 Выбор и обоснование типа информационной системы**

При моделировании бизнес-процессов было принято решение разработать обучающую программу с веб-интерфейсом по внутренним форматам простых типов данных в виде набора программных блоков, где каждый блок отвечает за свою страницу. Данные блоки могут быть реализованы в качестве вызываемых файлов на html странице [4].

## **3.2 Система управления базами данных**

Проанализировав бизнес процессы к обучающей веб программе по внутренним форматам простых типов данных, можно сделать вывод, что данная система не нуждается в базе данных.

Данная система работает с числами и введенными пользователям, данный элемент программы не нуждается в хранении в БД. Однако в программе присутствует форма обратной связи, которая отвечает за работу с обращениями пользователей, но данный элемент тоже не нуждается в хранении, та как. все обращения уходят на привязанную к сайту почту.

## **3.3 Хостинг**

Все три блока находятся в вызываемых файлах сайта, который в свою очередь будет располагаться на хостинге. Наилучшим вариантом хранения является аренда одного из хостингов, так как сайт не хранит конфиденциальную информацию, программа не нуждается в собственных серверах. Выбор пал на сайт reg.ru, так как это один из самых популярных хостингов в интернете и данный ресурс отвечает за качество предоставляемых услуг, плюсом также является бесплатный 10 дневный пробный период.

## **3.4 Модель данных (ER модель)**

Несмотря на то, что система обучающая программа с веб-интерфейсом по внутренним форматам простых типов данных не нуждается в базе данных, необходимо разработать инструмент хранения информации о том, какие пожелания или баги были обнаружены конечным пользователем, для дальнейшего устранения или реализации.

Онлайн тренажер переводит различные типы данных двух языков программирования (C# и Pascal) в двоичную и шестнадцатеричную систему счисления, а также контролирует обратный перевод чисел пользователем.

Алгоритм работы онлайн тренажера.

Пользователь выбирает режим язык программирования, типы данных которого будут переводиться в другие системы счисления.

Пользователь выбирает тип данных, который будет переводиться в другие системы счисления.

Если выбран режим обучения, пользователь вводит данные в зависимости от типа и нажимает кнопку перевода, а иначе пользователь нажимает кнопку генерации данных, а калькулятор генерирует случайное корректное значение данных, которое не отображается пользователю.

Калькулятор переводит введенные или сгенерированные данные в двоичную и шестнадцатеричную системы счисления.

Результаты перевода отображаются пользователю.

Если был выбран режим контроля:

* пользователю предлагается самому перевести данные из других систем в исходную;
* если данные введены верно, пользователю сообщается о правильности его ответа;
* если данные введены неверно, пользователю сообщается правильный ответ.

Исходя из описания работы онлайн тренажера, можно выделить следующие объекты:

* пользователь;
* калькулятор;
* данные.

На основе выделенных объектов можно построить ER-диаграмму, представленную на рисунке 3.1.

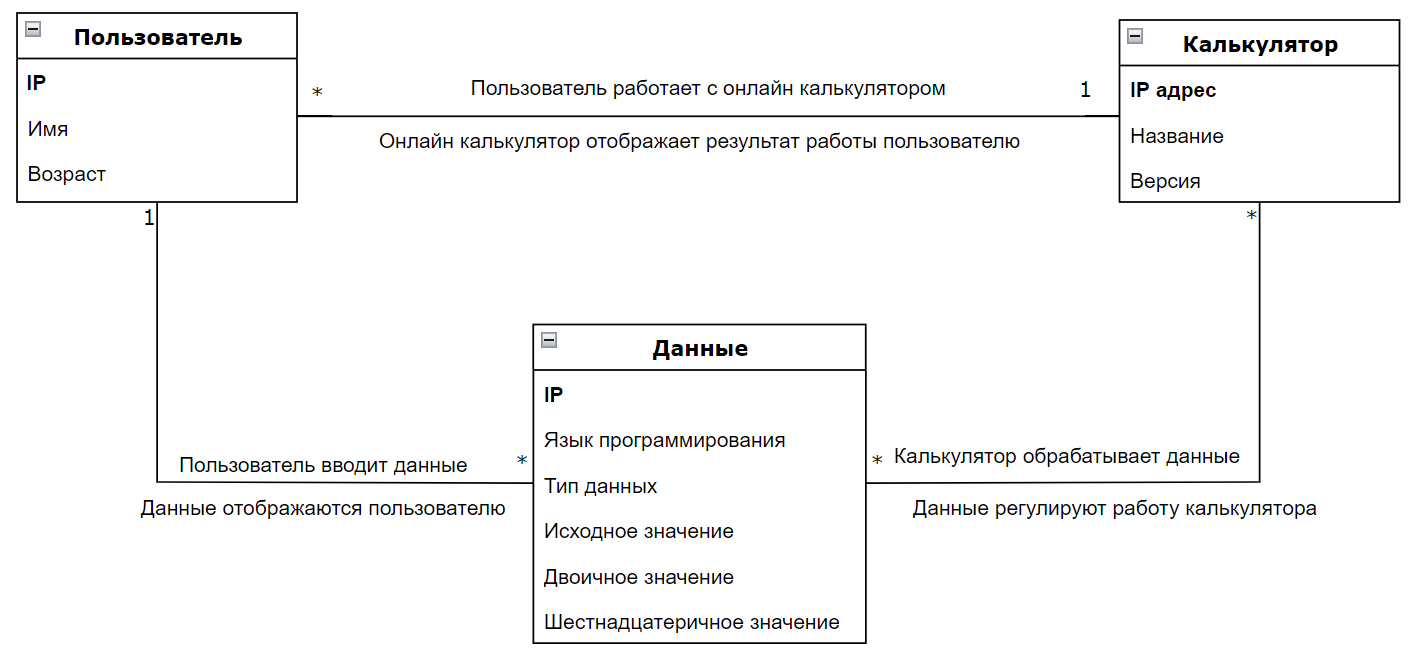


Рисунок 3.1 – ER-модель онлайн тренажера

Как отдельный пункт можно вынести работу письма обратной связи, форма обратной связи на прямую не связана с работой тренажера, однако косвенный функционал затронуть стоит.

* пользователь вводит свое имя и email;
* пользователь пишет само письмо с пожеланиями или найденными ошибками;
* если на форме не заполнено хотя бы одно из полей, то форма выдает ошибку заполнения и не дает отправить письмо, пока все поля не будут заполнены. Не заполненные поля подсвечиваются красным цветом;
* если все данные введены корректно, пользователь нажимает кнопку отправить и письмо уходит на почту конвертера.

Исходя из описания работы онлайн тренажера, можно выделить следующие объекты:

* пользователь;
* форма обратной связи;
* данные.

На основе выделенных объектов можно построить ER-диаграмму, представленную на рисунке 3.2.

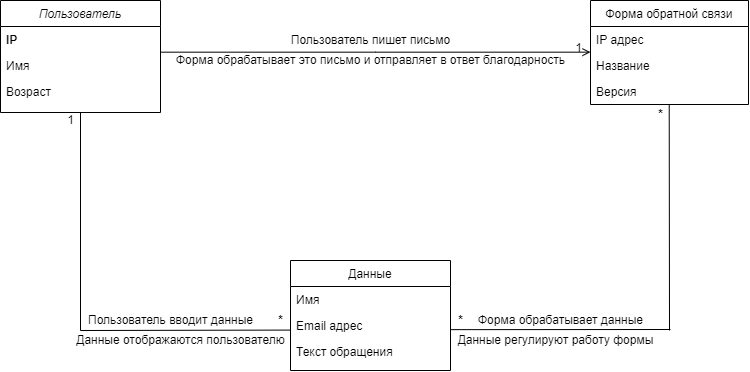


Рисунок 3.2 – ER-модель формы обратной связи

## **3.5 Проектирование алгоритмов перевода**

Алгоритм основной программы представлен на рисунке 3.3 в виде блок– схемы. Подробности реализации и назначение методов описано ниже.

Для перевода целых знаковых чисел используется метод IntConvert, а для целых неотрицательных [5] - UIntConvert. Для записи результата используется строка GetTwoC, а число записывается в переменную Number. Переменная kl-логическая переменная для остановки перевода, BCount – количество бит, выделенных для перевода числа в выбранный пользователем тип данных [6].

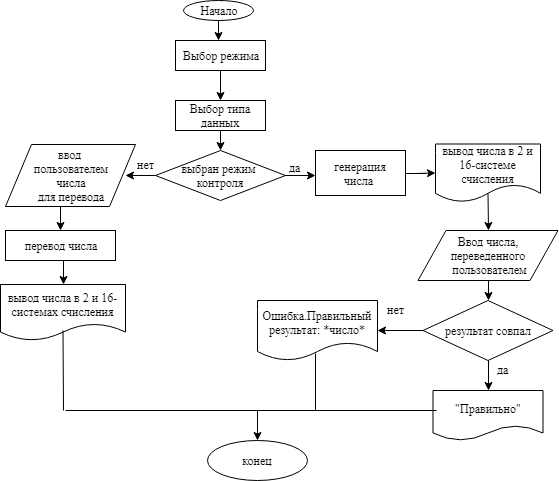


Рисунок 3.3 – Блок-схема основной программы

ConvertReal-метод перевода чисел в тип данных double, float. Str-переменная для записи числа во внутреннем представлении,S1-строковая переменная для записи целой части [7],S2- строковая переменная для записи дробной части, а- целая часть числа,b-дробная часть числа,Order-позиция начала дробной части/конца целой части, k1-переменная для хранения остатка, Е-для записи экспоненты, eConst-количество битов, выделенных для записи экспоненты[8], fConst-количество битов, выделенных для записи нормализованной мантиссы, Const-количество бит для записи числа (размер типа данных в битах). Также в методе вызывается метод TranslationFraction, который переводит дробную часть вещественного числа [9].

Алгоритм перевода числа типа extendet представлен на рисунке 3.4.

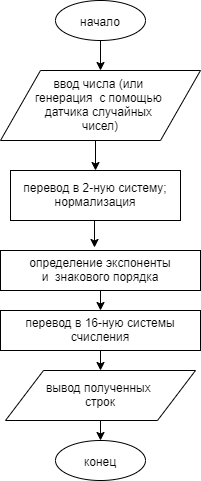


Рисунок 3.4 – Блок-схема перевода чисел extendet

На рисунке 3.5 представлены алгоритмы перевода символов и строк с помощью кодировочных таблиц и внутреннего метода Encoding [10].

Для получения внутреннего представления чисел типа данных currency используем алгоритм, представленный на рисунке 3.6.

Так как в процессе написания выпускной квалификационной работы используется язык программирования JavaScript, то используем указатели для получения значения байтов [11], в которых хранятся числа, символы или строки. Используем этот же принцип для аналогичных типов данных языка Pascal. Данный алгоритм представлен на рисунке 3.7.

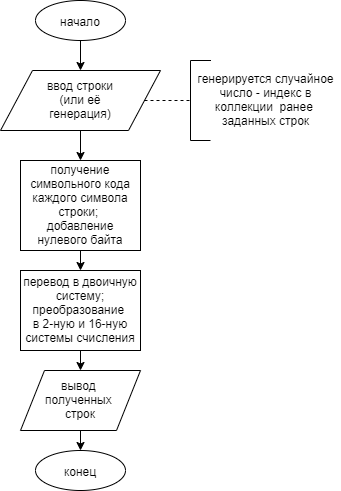
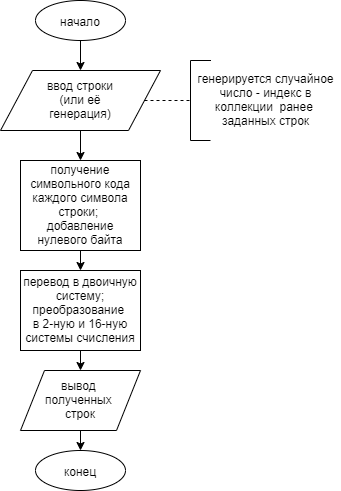


Рисунок 3.5 – Блок-схема перевода строк и символов



Рисунок 3.6 – Блок-схема перевода currency

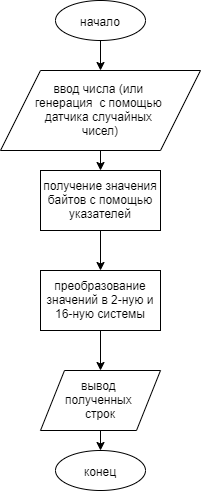


Рисунок 3.7 – Блок-схема перевода с помощью указателей

## **3.6 Графический интерфейс веб-тренажера**

Опираясь на техническое задание, сделаем вывод, что пользователь обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных может взаимодействовать с системой. Вся работа выполняется на трех страницах веб тренажера. Для работы системы достаточно вывода лишь нескольких сообщений на странице перевода или форме обратной связи. Основная активность со стороны пользователя это ввод чисел в специализированное поле. Если же это форма обратной связи, то ввод текста пожелания или найденного бага в подготовленные поле. На рисунке 3.8 изображен элемент header с интерактивными кнопками при нажатии на которые можно перейти на другую страницу.

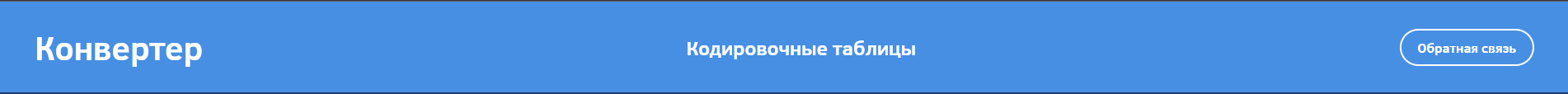


Рисунок 3.8 – Элемент header с интерактивными кнопками

Также на каждой странице присутствуют кнопки перехода на другие страницы сайт. На рисунке 3.9 изображена главная страница сайта, которая встречает пользователя при первом посещении веб страницы. На ней присутствуют еще две кнопки Pascal и C# при нажатии на которые, пользователя перебрасывает на страницы перевод данных [12].

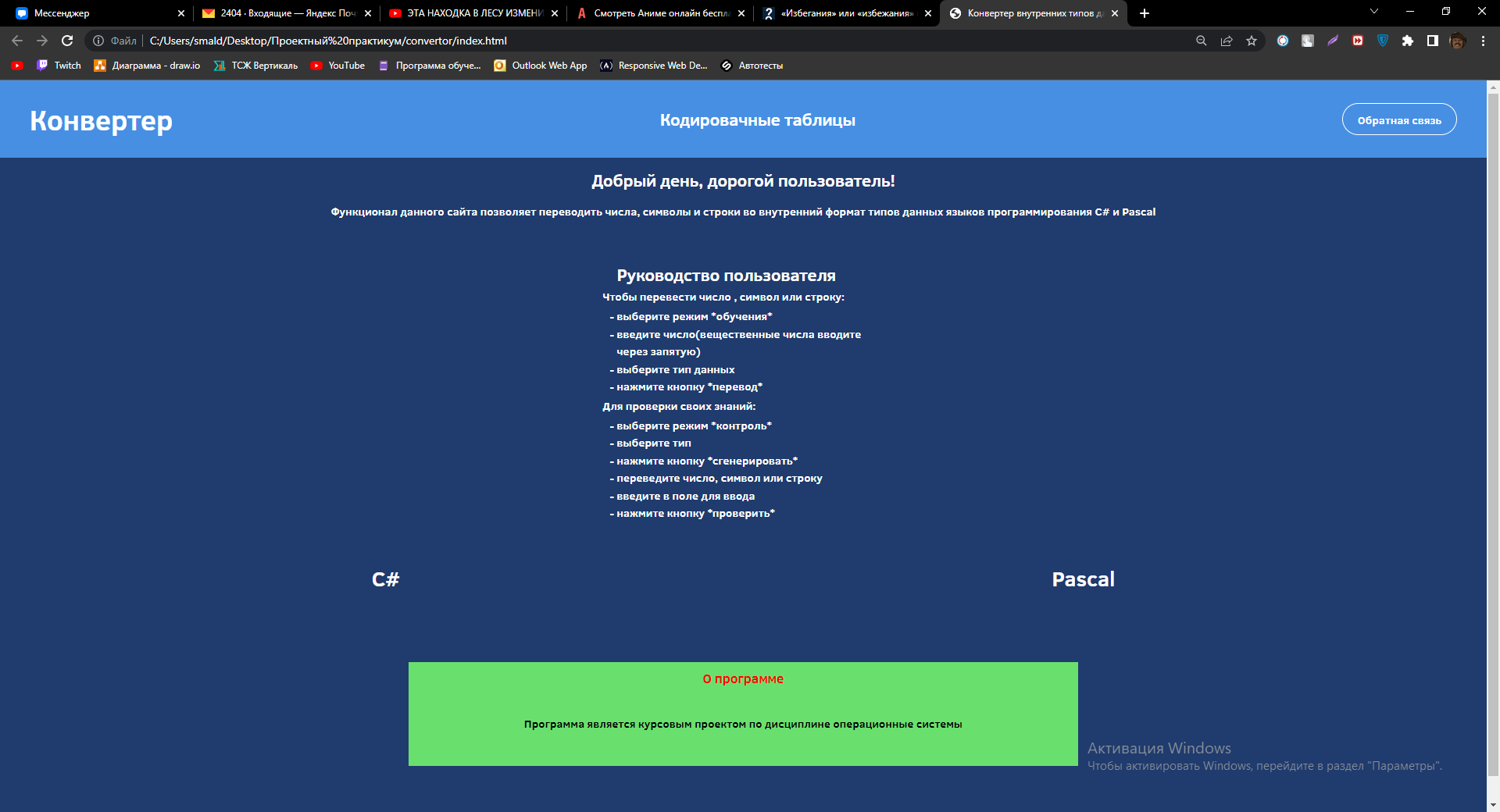


Рисунок 3.9 – Главная страница сайта

На рисунках 3.10 -3.13 представлены страницы с конвертацией типов данных, формы обратной связи и ознакомительной страницы с кодировочными таблицами, необходимыми для успешного перевода чисел. Все эти страницы обладают кликабельными элементами, за исключением страницы с кодировочными таблицами, она служит ознакомительной страницей и не обладает интерактивными элементами.

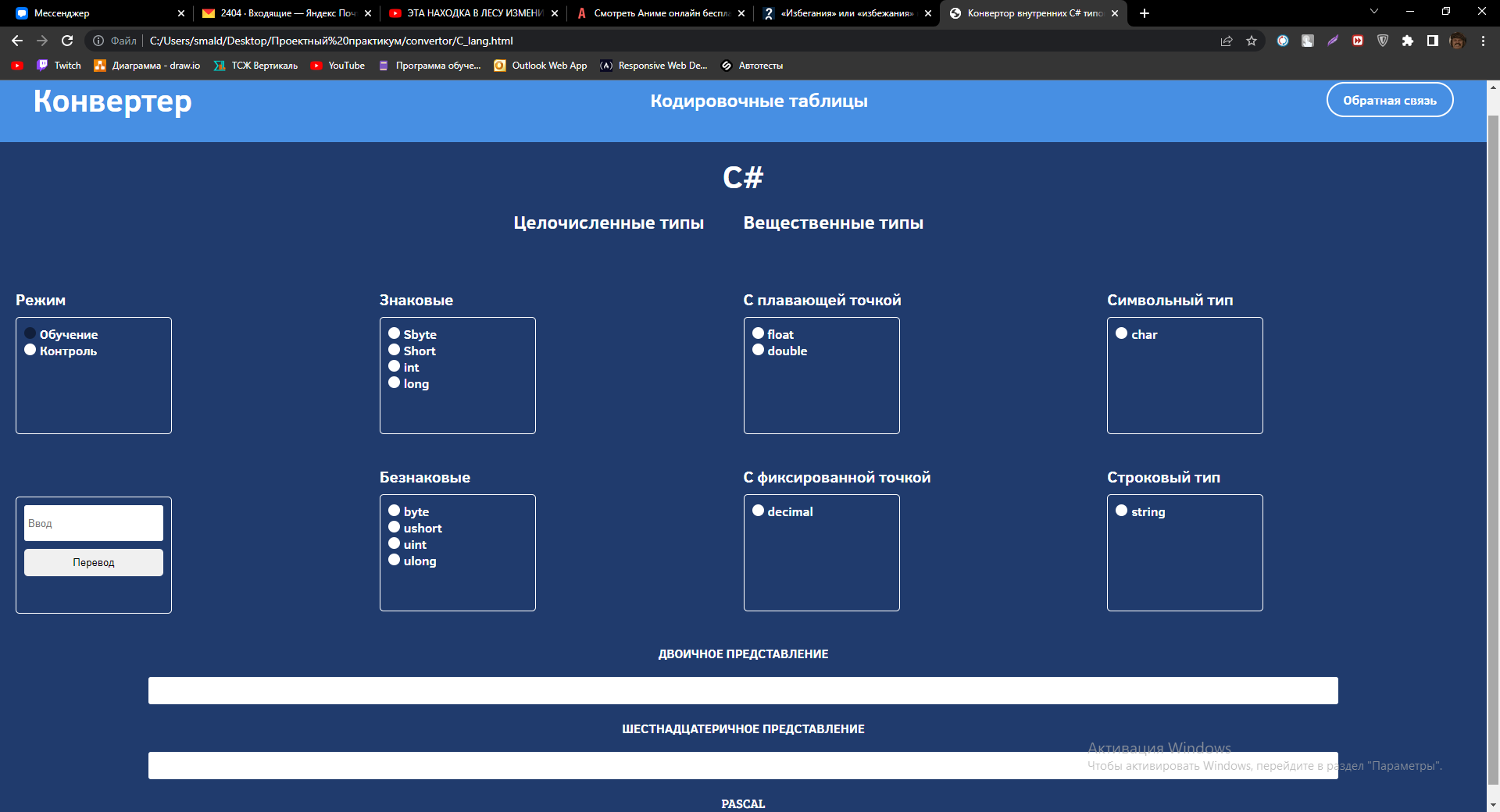


Рисунок 3.10 – Страница перевода чисел во внутренние форматы языка C#

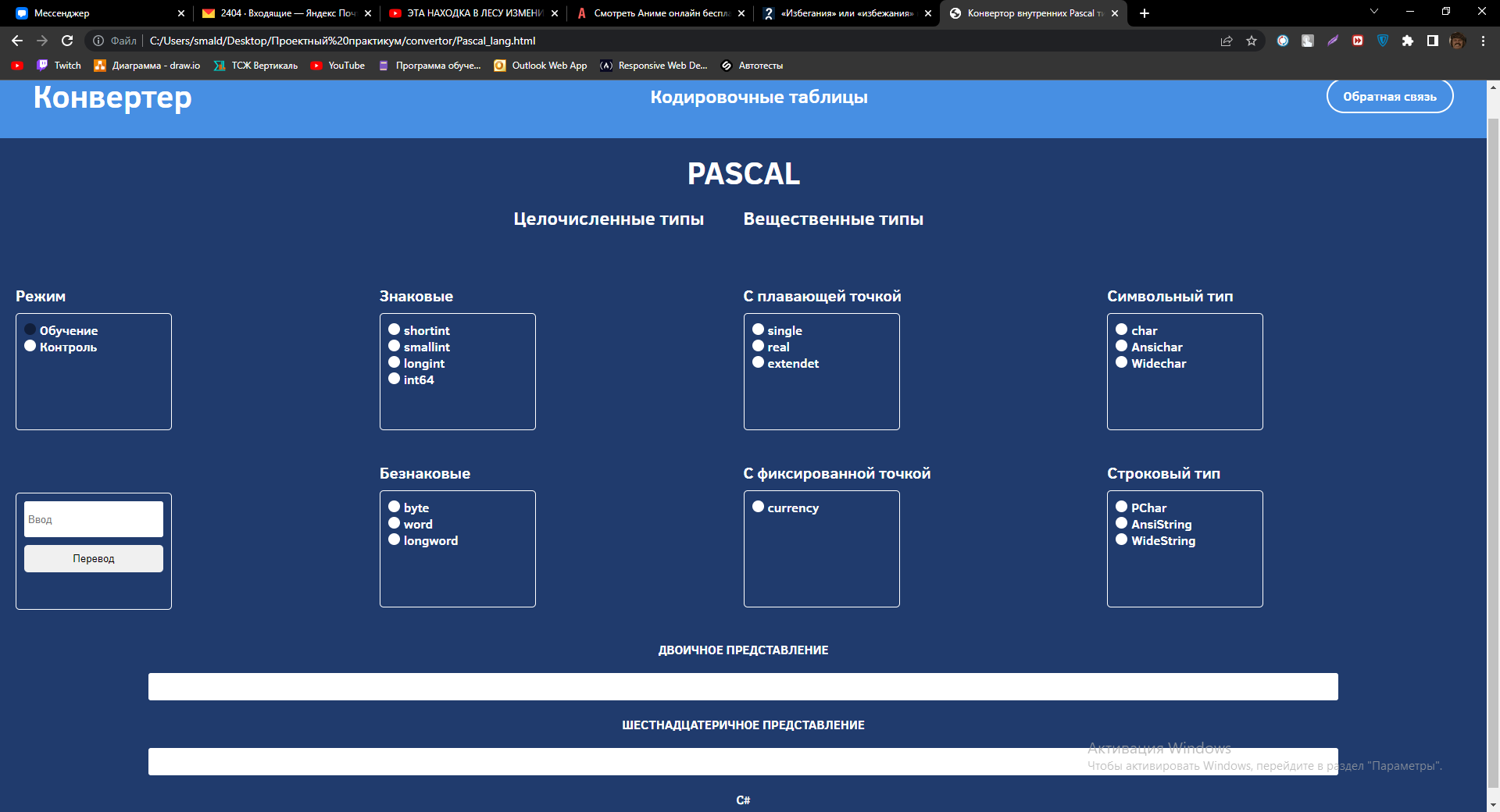


Рисунок 3.11 – Страница перевода чисел во внутренние форматы языка Pascal

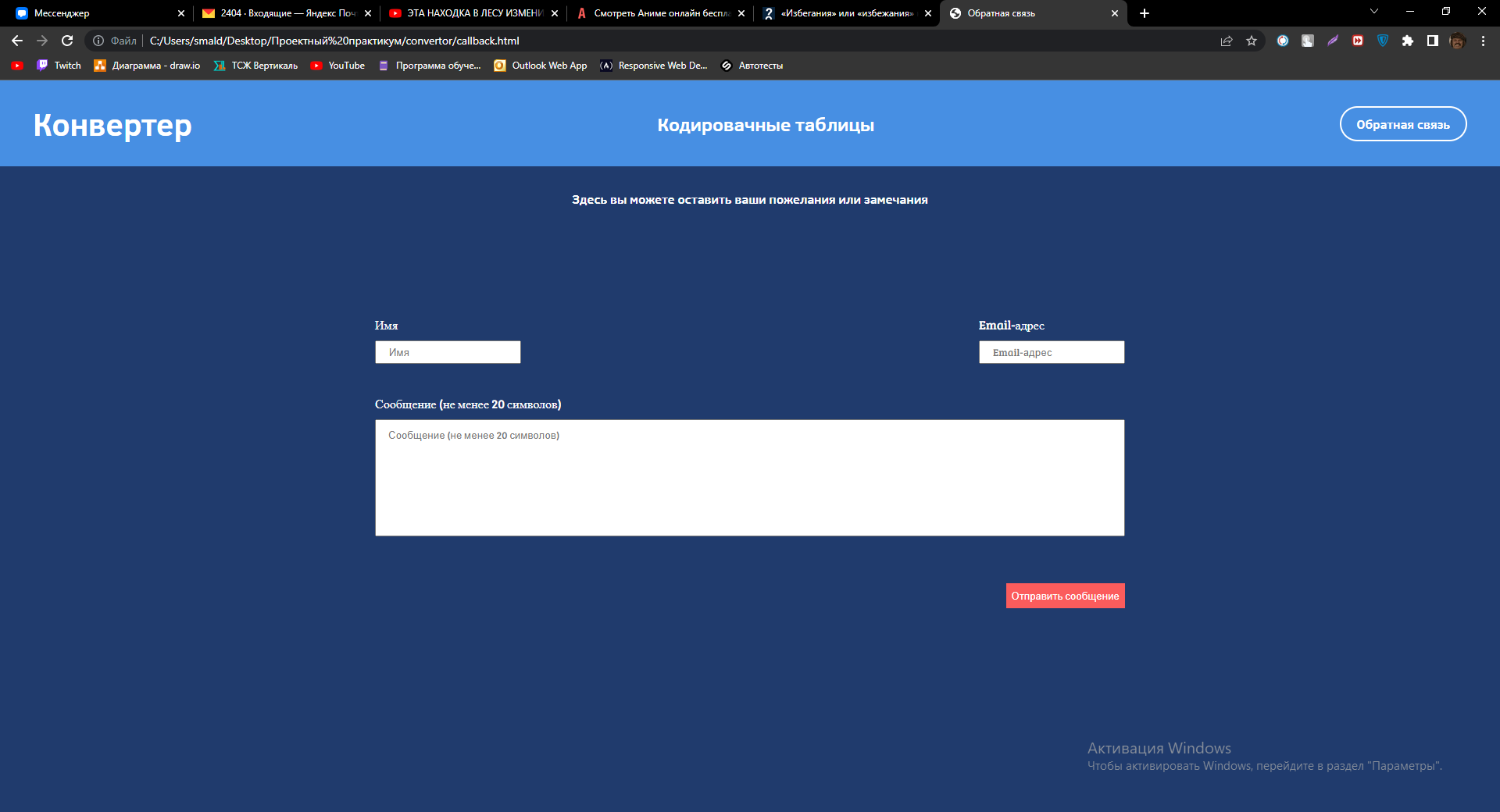


Рисунок 2.12 – Страница формы обратной связи



Рисунок 3.13 – Страница с кодировочными таблицами

Для удобной работы пользователя на сайте был добавлены такие элементы как «text» «radiobutton» «button». Соответственно это поле ввода, кнопки выбора типа данных и кнопки генерации и проверки чисел. Все эти элементы присутствуют на страницах перевода данных и форме обратной связи.

При вводе некорректного числа появляется сообщение об ошибке, пример представлен на рисунке 3.14

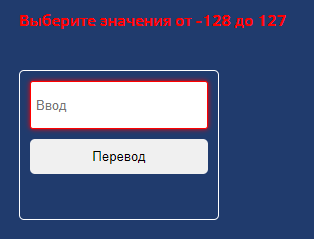


Рисунок 3.14 – Вывод сообщения об ошибке

## **3.7 Средства и методы тестирования**

На рисунках 3.15 и 3.16 изображено окно программы при корректном вводе данных.

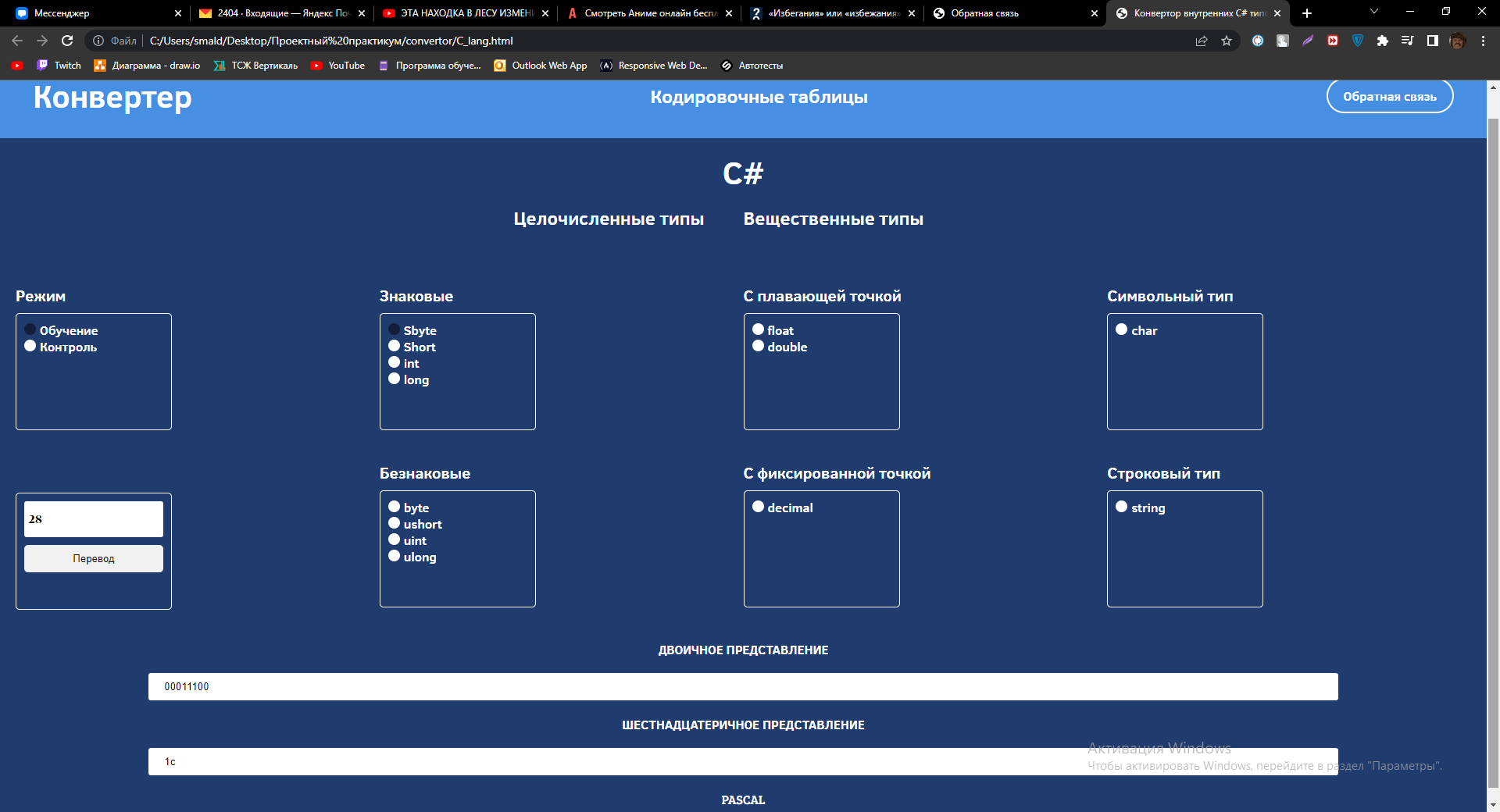


Рисунок 3.15 – Окно программы при корректном вводе данных

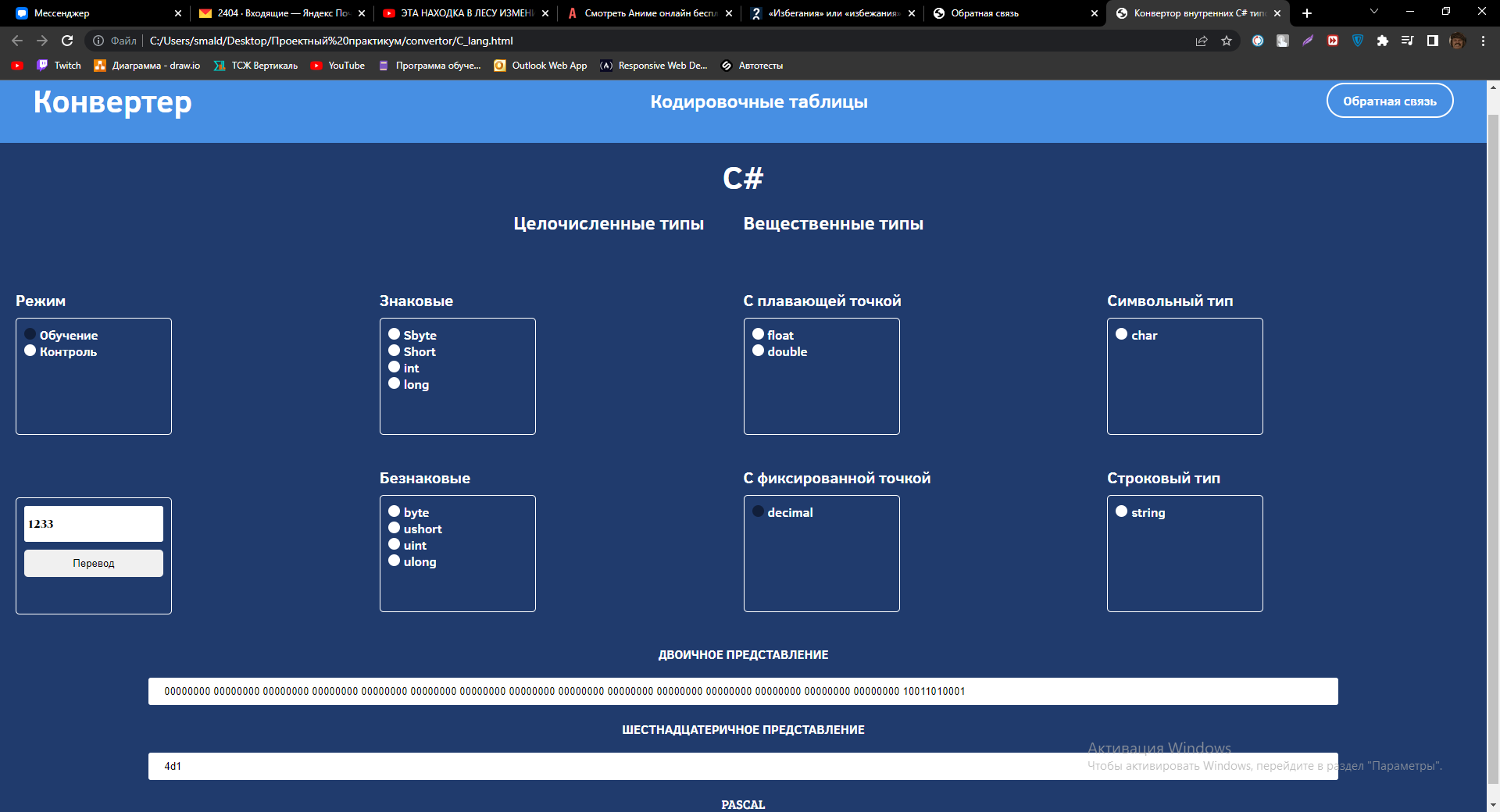


Рисунок 3.16 – Окно программы при корректном вводе данных

Если пользователь вводит число, выходящее за диапазон значений типа данных, или вводит не число, то программа выводит сообщение об ошибке. Примеры представлены на рисунках 3.17 и 3.18.

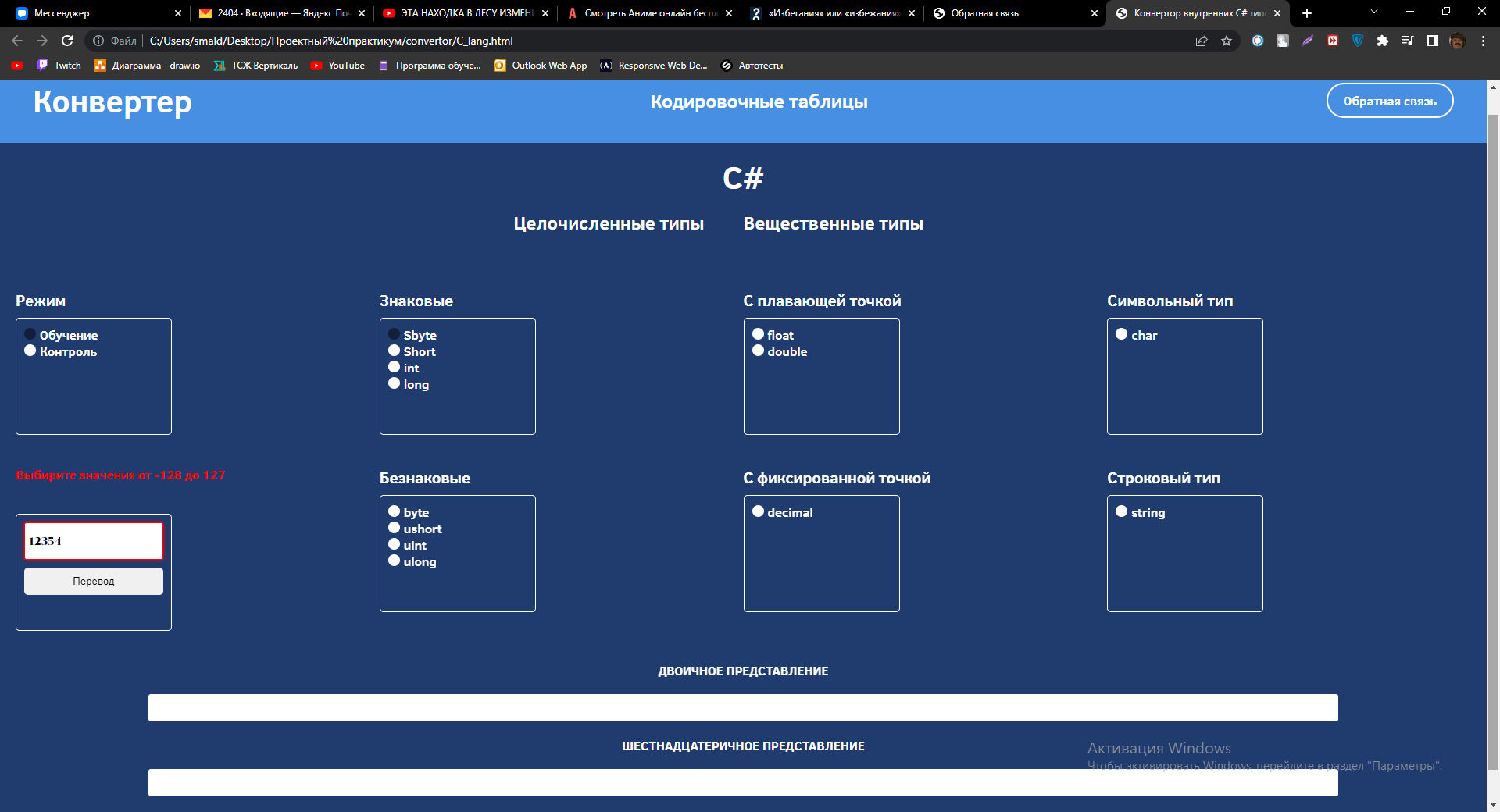


Рисунок 3.17 – Окно программы при некорректном вводе данных

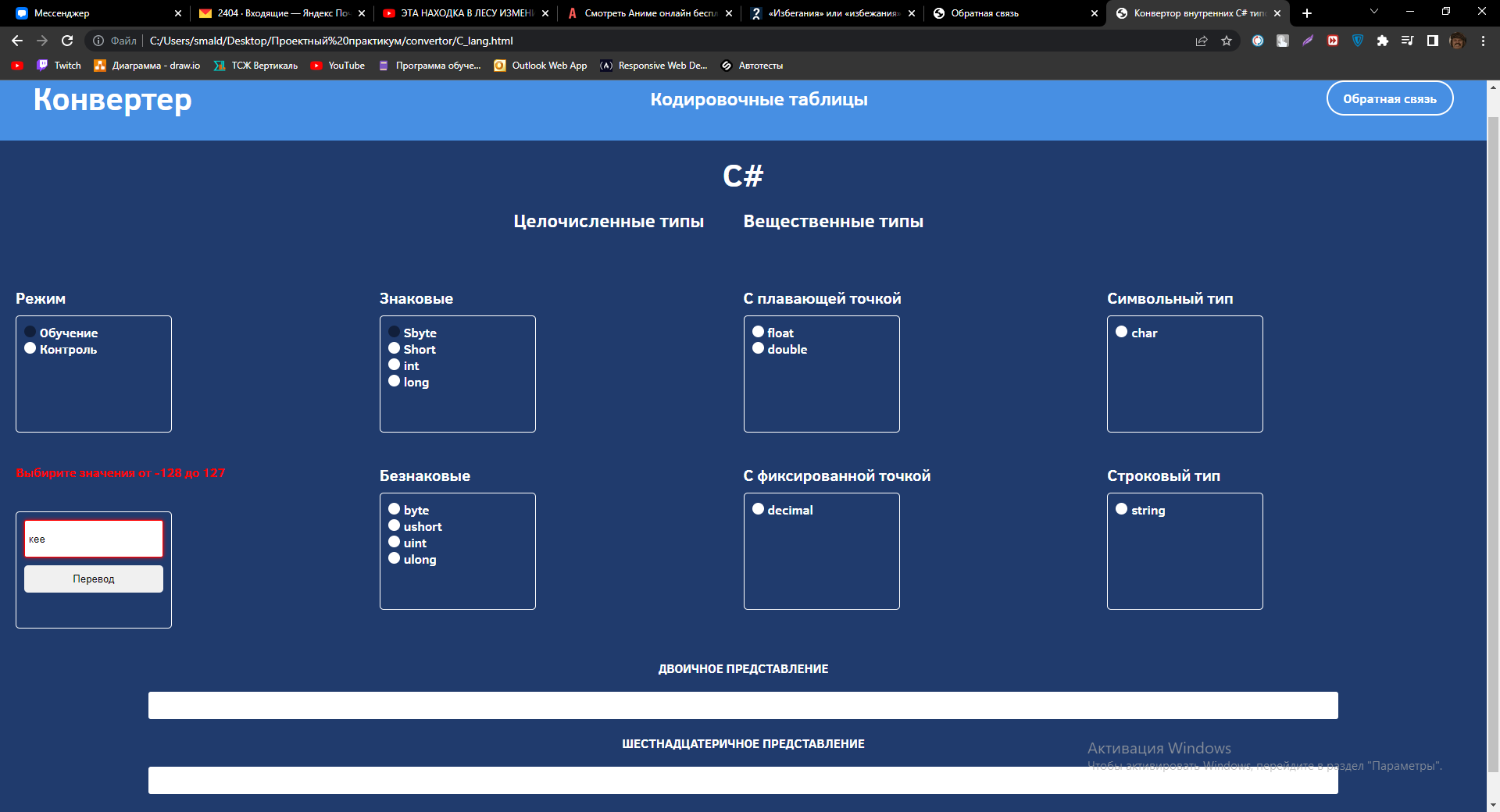


Рисунок 3.18 – Окно программы при некорректном вводе данных

Если пользователь введет строку вместо символа при выбранном символьном типе, то программа выведет сообщение об ошибке, рисунок 3.19.

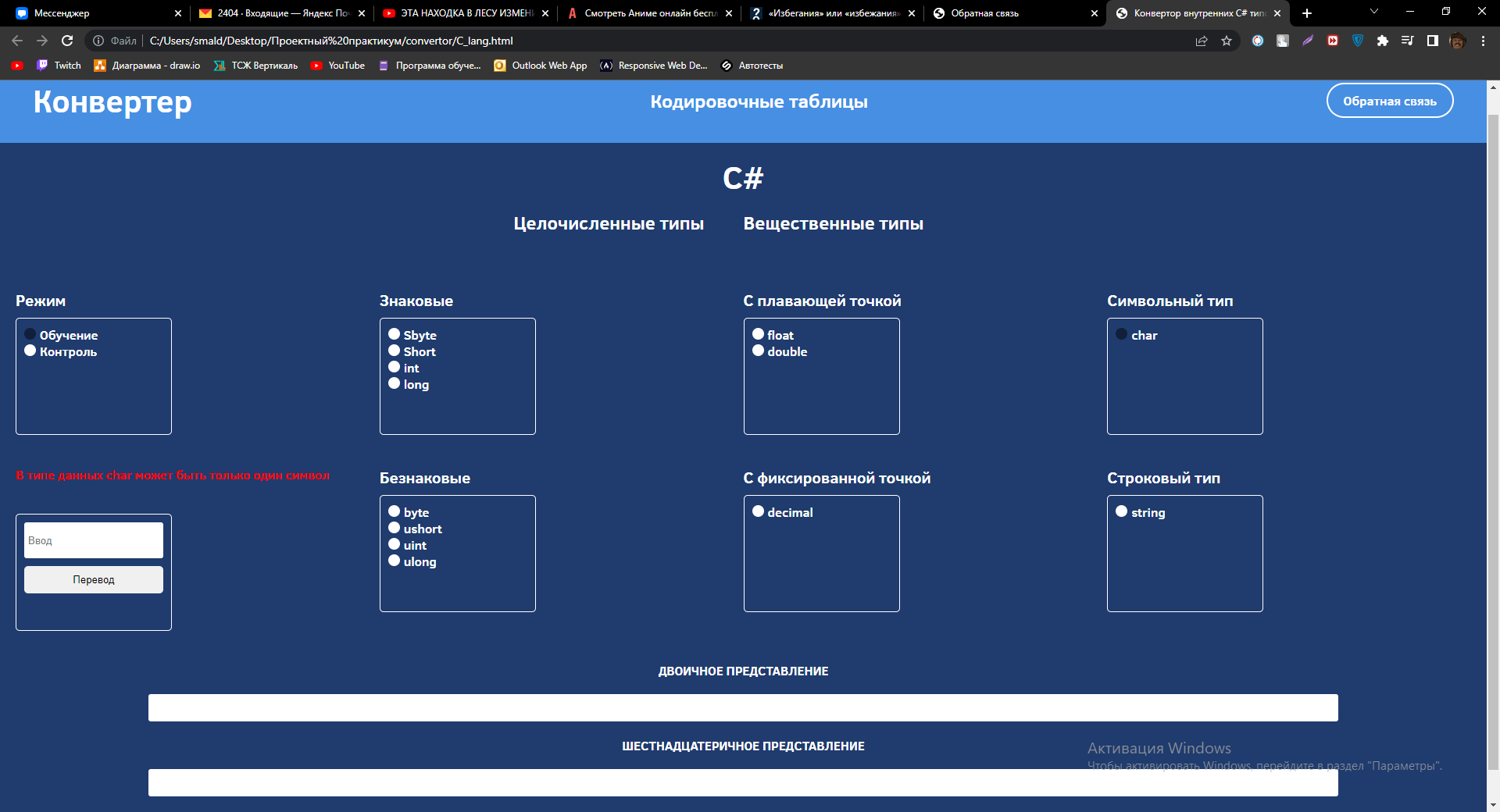


Рисунок 3.19 – Результат работы программы при неккоректном вводе

## **3.8 Развертывание, установка, настройка информационной системы**

Обучающая программа с веб-интерфейсом по внутренним форматам простых типов данных состоит исключительно из сетевой части, от пользователя требуется только пк с доступом к сети интернет. Пк должен обладать минимальными системными требованиями и стабильным интернет соединением. Также для успешного обучения и перевода необходимо ознакомиться с таблицами переводов. Данные таблицы находятся на сайте на странице «Кодировочные таблицы» [13].

## **3.9 Описание функциональных особенностей информационной системы**

Разработанная система позволяет:

* проверять корректность введенных пользователем чисел в соответствии с выбранным типом данных;
* переводить введенное пользователем число во внутренние типы данных языков C# и Pascal;
* переводить преобразованные числа из внутренних форматов в двоичную и шестнадцатеричную системы счисления;
* работать в режиме генерации чисел и проверки знания пользователя;
* отправлять письма с пожеланиями пользователей на внутреннюю почту веб приложения.

## **3.10 Руководство администратора**

Данные о прикрепленной к сайту почте хранятся в файле конфигураторе, данную почту можно изменить на любую другую при необходимости. Также в этом же файле хранится шаблон письма пользователя, он тоже может быть изменен при необходимости. Информация об ошибках хранится в консоль браузера, которая в свою очередь сохраняется в логах на хостинге [14].

Файл конфигурации ссылается на благодарственный ответ, данный файл хранится в одной папке с файлом конфигурации и при желании может быть изменен. Все алгоритмы хранятся в папке JS, если понадобится отладка алгоритмов, то все они хранятся там

## **3.11 Руководство пользователя**

Разработанный тренажер полностью автоматизирован и требует вмешательства пользователя только по одной причине – ввести число для перевода или ввести число для проверки знаний, либо отправить обращение на имя разработчика.

От пользователя только требуется выбрать режим взаимодействия, это либо «Обучения», либо контроль. В зависимости от выбранного режима порядок действия отличается. Также присутствует возможность отправки письма на имя разработчика с пожеланиями, либо отчете о найденном баге.

При выборе режима «Обучения», тренажер начинает работу по следующему алгоритму:

1. Происходит проверка выбора (Был ли выбран тип данных).
2. Происходит проверка адекватности введенного числа, соответствует ли оно выбранному типу данных и находится ли оно в допустимом диапазоне.
3. Если при вводе пользователем числа не произошло ошибки, тренажер начинает перевод числа пользователя в выбранный тип данных.
4. После перевода в выбранный тип данных, тренажер преобразует значение из десятичной системы счисления в двоичную и шестнадцатеричную.
5. Полученные в ходе перевода значения тренажер выводит в поля вывода данных.

При выборе пользователем режима «Контроль», тренажер действует согласно следующему алгоритму:

1. Пользователь нажимает кнопку «Сгенерировать» и в случае, если не выбран тип данных, программа выбрасывает ошибку.
2. Программа задает случайное число, в диапазоне, соответствующем выбранному типу данных.
3. Программа переводит сгенерированное числа в двоичную и шестнадцатеричную систему счисления, и выводит полученный результат в специализированные поля.
4. Пользователь вводит высчитанное им число в поле ввода.

Если же пользователь решил отправить письмо на имя разработчика, то программа действует в соответствии с этим алгоритмом:

1. Пользователь вводит свое имя, Email и сообщения для разработчика.
2. Программа проверяет заполнены ли все поля на форме.
3. Отправляет письмо на почту, прикрепленную к веб-тренажеру.

## **3.12 Условия обеспечения работоспособности информационной системы**

Для корректного функционирования онлайн тренажера требуется:

* наличие подключения по сети Интернет;
* наличие современного браузера;
* процессор с частотой не ниже 1ГГц;
* объем ОЗУ не менее 512 Мб;
* наличие сетевого адаптера.

## **3.13 Результаты реализации веб тренажера**

В ходе реализации обучающей программы с веб-интерфейсом по внутренним форматам простых типов данных были описаны:

* алгоритмы перевода данных;
* графический интерфейс программы;
* порядок развертывания, установки и настройки системы;
* функциональные особенности системы;
* руководства администратора и пользователя системы;
* условия обеспечения работоспособности системы.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения выпускной квалификационной работы были выполнены этапы, описанные ниже.

Аналитическое исследование:

* наименование и область применения системы;
* постановка задач;
* анализ бизнес-процессов;
* использовать JavaScript в качестве основного языка разработки системы моделирование бизнес-процессов.

Проектирование информационной системы:

* обзор методологий моделирования;
* моделирование бизнес процессов;
* не использовать системы управления базами данных;
* использовать общедоступные хостинги для выкладки сайта;
* хранить данные обращений пользователей на почте, привязанной к сайту;
* выполнять проверку работоспособности системы вручную.

Реализация веб-тренажера:

* описание функциональных особенностей системы;
* выкладка сайта на общедоступный хостинг;
* разработка ER-моделей;
* спроектированы алгоритмы перевода и графический интерфейс;
* средства и методы тестирования;
* выполнять проверку работоспособности системы вручную;
* развертывание разработанной системы;
* руководства администратора и пользователя системы;
* условия обеспечения работоспособности системы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Виды бизнес процессов и их составляющие [Электронный ресурс]. – URL: https://rzbpm.ru/knowledge/pravilnyj-biznes-process-eto-osnova-otlichnyx-rezultatov.html (дата обращения 14.05.2022).
2. Сравнение интегрированных сред разработки [Электронный ресурс]. – URL: https://timeweb.com/ru/community/articles/5-luchshih-ide-1 (дата обращения 14.05.2022).
3. Главные методологии моделирования бизнес процессов: какие бывают и как выбрать; [Электронный ресурс]. – URL: https://checkroi.ru/blog/metodologii-modelirovaniya-biznes-processov (дата обращения 18.05.2022).
4. AutoUpdater.NET: Репозиторий пользователя ravbpatel на сайте GitHub; [Электронный ресурс]. – URL: github.com/ravibpatel/AutoUpdater.NET (дата обращения 18.05.2022).
5. Кобылянский В. Г., Агуленко Н. И. Хранение данных в ЭВМ. Методические указания к практическим занятиям и лабораторным работам. – Новосибирск, 2019. // Режим доступа: http://moodle3.stu.ru (дата обращения 25.05.2022 г.).
6. Кобылянский В. Г. Архитектура вычислительных систем: Учебное пособие. - Новосибирск: Изд-во СГУПС, 2012. – 105 с.
7. Методические указания к выполнению РГР / В.Г. Кобылянский. (дата обращения 25.05.2022 г.).
8. Новосибирск: [Электронный ресурс] — URL: http://moodle3.stu.ru/pluginfile.php/91029/mod\_resource/content/5/МУ%20для%20РГР.doc (дата обращения: 27.05.2022).
9. Элементы управления WPF [Электронный ресурс] — URL: <https://professorweb.ru/my/WPF/UI_WPF/level6/UI_WPF_index.php>(дата обращения: 28.05.2022).
10. Методы [Электронный ресурс] — URL: https://metanit.com/sharp/tutorial/2.8.php (дата обращения: 30.05.2022).
11. [Javascript: использование ссылок на переменную (или указателей)](https://ru.stackoverflow.com/questions/804918/javascript-%d0%b8%d1%81%d0%bf%d0%be%d0%bb%d1%8c%d0%b7%d0%be%d0%b2%d0%b0%d0%bd%d0%b8%d0%b5-%d1%81%d1%81%d1%8b%d0%bb%d0%be%d0%ba-%d0%bd%d0%b0-%d0%bf%d0%b5%d1%80%d0%b5%d0%bc%d0%b5%d0%bd%d0%bd%d1%83%d1%8e-%d0%b8%d0%bb%d0%b8-%d1%83%d0%ba%d0%b0%d0%b7%d0%b0%d1%82%d0%b5%d0%bb%d0%b5%d0%b9); [Электронный ресурс]. – URL: github.com/NetSparkleUpdater/NetSparkle (дата обращения 08.06.2022).
12. AutoUpdaterEasy: Репозиторий пользователя cecon на сайте GitHub; [Электронный ресурс]. – URL: https://ru.stackoverflow.com/questions/804918/javascript (дата обращения 09.04.2022).
13. Zl.AutoUpgrade.Core: Репозиторий пользователя zenglo на сайте GitHub; [Электронный ресурс]. – URL: github.com/zenglo/AutoUpgrade (дата обращения 09.06.2022).
14. Introducing AppLife Update: руководство разработчиков; [Электронный ресурс]. – URL: www.kineticjump.com/update/default.aspx (дата обращения 12.06.2022).