Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

ГЕНЕРИРОВАНИЕ СЛУЧАЙНЫХ ЧИСЕЛ И ИХ АНАЛИЗ

Пояснительная записка

к курсовой работе по дисциплине

«Основы алгоритмизации и программирования»

**КП9К.939315.104.081ПЗ (посмотреть в СТП 2017 правильное оформление)**

Руководитель А.Н. Василькова (Кабариха, если он)

Учащийся гр. Е.Г. Книга

Минск 2024

**Содержание (сверить по листу задания и написать свое)**

[Введение 3](#_Toc72958838)

[1 Постановка задачи 4](#_Toc72958839)

[1.1 Описание предметной области 4](#_Toc72958840)

[1.2 Обзор существующих аналогов 5](#_Toc72958841)

[1.3 Функциональное назначение 9](#_Toc72958843)

[2 Проектирование задачи 10](#_Toc72958844)

[2.1 Алгоритм работы программы 10](#_Toc72958845)

[2.2 Выбор и обоснование инструментов разработки 12](#_Toc72958847)

[3. Программная реализация 14](#_Toc72958848)

[3.1 Описание алгоритмов решения задачи 14](#_Toc72958849)

[3.2 Проетирование интерфейса приложения](#_Toc72958850) 15

[4 Тестирование 17](#_Toc72958851)

[5 Руководство пользователя 20](#_Toc72958852)

[Заключение 23](#_Toc72958854)

[Список используемых источников 24](#_Toc72958855)

[ПРИЛОЖЕНИЕ А 25](#_Toc72958856)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

**КП9К.939315.104.081ПЗ (как на титульнике)**

Разраб.

Книга

Провер.

Василькова

Реценз.

Н. Контр.

Утверд.

Генерирование случайных чисел и их анализ

Пояснительная записка

Лит.

Листов

35

БГУИР

Введение

Курсовая работа «Генерирование случайных чисел и их анализ» предназначен для знакомства c различными алгоритмами генерации случайных чисел, в которой должны быть реализованы возможности генерации абсолютно случайных чисел, а также с изучением свойств этих чисел.

Генератор случайных чисел (ГСЧ, англ. Pseudorandom number generator, PRNG) - алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному).

При реализации курсового проекта «Генерирование случайных чисел и их анализ» были поставлены следующие задачи:

* изучение различных алгоритмов генерирования случайных чисел;
* изучение и анализирование данных сгенерированных чисел;
* изучение дополнительной и необходимой информации к анализу случайных чисел;
* наличие возможности выбрать диапазон для будущего генерирования чисел;
* возможность выбора любого из способов генерации чисел;
* реализация понятного и простого интерфейса.

При разработке курсового проекта «Генерирование случайных чисел и их анализ» были проанализированы аналоги: «AlpenBot», «Service-online.su», «Randstuff.ru», «Medstatistic.ru». На основе аналогов собирались функциональные возможности программного средства и создавался курсовой проект проекта «Генерирование случайных чисел и их анализ».

Курсовой проект «Генерирование случайных чисел и их анализ» ориентирован на новичков, желающих получить базовые знания о свойствах нескольких чисел, а также на обычных пользователей, желающих улучшить свои знания в анализе множества чисел.

Для работы над курсовым проектом «Программное средство обработки текстовых файлов» используется IDE (интегрированная среда разработки) Microsoft Visual Studio. Ее преимуществами являются ее популярность, следовательно, большое количество различной документации, также относительно удобная структура интерфейса, поддержка разных языков программирования, большое количество плагинов и дополнений, доступных для скачивания. Отдельно нужно отметить, что IDE позволяет удобно создать текстовый файл и начать работу с ним во время написания кода программы. В Microsoft Visual Studio заранее существует большое количество удобных и популярных библиотек, доступных для подключения в проекте.

С развитием технологий упрощается и визуализируется обучение для быстрого восприятия материала. Поэтому тема курсового проекта является актуальной и может быть использована самостоятельно и обучающем процессе.

# 1 Постановка задачи

Курсовая работа посвящена написанию программного средства «Генерирование случайных чисел и их анализ». Цель курсового проекта – рассмотрение множества способов генерирования чисел и упрощение изучения свойств чисел.

Функции и возможности программы заключаются в следующем:

* изучение способов генерации случайных чисел;
* возможность выбора любого из способов генерирования чисел;
* возможность получения подробного анализа чисел;
* возможность получения высокой степени точности вычислений;
* закрепление знаний с помощью дополнительных сведений об анализе чисел, предоставленных в курсовом проекте.

Основное назначение программного средства «Генерирование случайных чисел и их анализ» - приобретение новых знаний, совершенствование старых в предметной области, их тестирование.

## 1.1 Описание предметной области

Основная задача курсового проекта заключается в разработке программного средства «Генерирование случайных чисел и их анализ».

Предметная область – та часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы.

Учебные компьютерные модели (программы, предназначенные для обучения) позволяют наблюдать явления и процессы в динамике. В то же время такие программы предоставляют возможность увидеть на экране компьютера абстрактные объекты и явления, которые в реальном мире не существуют, а являются лишь моделями реальных объектов и явлений.   
Процесс представления предметной области задачи в виде совокупности объектов, обменивающихся сообщениями, называется объектной декомпозицией. В рамках объектной декомпозиции пытаются выделить основные содержательные элементы задачи, разбить их на типы (классы). Далее для каждого класса абстракций определяются его свойства (данные) и поведение (операции), а также взаимодействие этих классов абстракций друг с другом. В разрабатываемой системе главным объектом являются теоретическая информация, с ознакомлением которой происходит обучение пользователя. Обучаемый запускает программу, выбирает диапазон и любой из алгоритмов генерирования чисел, после чего делает анализ из генерируемого ряда чисел.

## 1.2 Обзор существующих аналогов

Аналог – это то, что представляет собою подобие чего–либо, соответствие какому либо предмету, явлению, понятию.

Аналогом называют объект того же назначения, близкий по совокупности существенных признаков.

В современном мире технологий, когда существует множество различных программ, практически каждая из них имеет один и более аналогов. Они похожи друг на друга, но в тоже время и имеют отличия. Даже если они незначительны, для многих пользователей они играют немаловажную роль. Каждый разработчик пытается внести что-то от себя, нечто эволюционное и новое, это может быть более удобный интерфейс или же добавление дополнительных функций, которых не было в других подобных программах. Главная цель автора - привлечение максимального количества пользователей к его программному средству. Это является непростой задачей, ведь нужно сделать так, чтобы "творение" понравилось большинству людей с совершенно разными предпочтениями и видами деятельности. Поэтому, мало просто написать программу, нужно изучить довольно много информации, проанализировать опыт уже имеющихся приложений, возможно, даже создать опросы среди потенциальных пользователей.

Приведем примеры нескольких самых известных программ генерации и анализа чисел.

«AlpenBot» – довольно известный портал с удобным и быстрым генерированием чисел для различных конкурсов, а также частичным его анализом. Он сочетает в себе 3 различных тарифа: базовый бесплатный (качает комментарии, проверит отметки друзей и выберет победителя), премиум и корпоративный, в котором доступны все возможности сервиса. Большинство пользователей предпочитают именно его, благодаря удобному интерфейсу и большому спектру возможностей, которые он может предоставить.

Преимуществами портала «AlpenBot» являются:

* большой выбор алгоритмов генерации.
* анализ активности пользователей.
* удобная панель инструментов на экране.
* многие элементы и функции являются автоматическими.
* можно одновременно задать несколько генераций случайных чисел в заданном диапазоне.
* возможность приблизить или отдалить страницы на экране.
* автоматика. Вы задаете параметры, а всю остальную работу система делает сама.
* возможность сохранять результаты на созданном аккаунте на сайте.

В свою очередь портал «AlpenBot» имеет ряд недостатков:

* периодически долгая загрузка и, возможно, медленная работа на слабых компьютерах.
* отсутствие возможности сохранять результаты для общего доступа.
* сложность поиска информации для рядового пользователя.
* отсутствие наглядности компонентов анализа
* отчасти сложный интерфейс.

На рисунке 1.1 изображен интерфейс портала «AlpenBot».

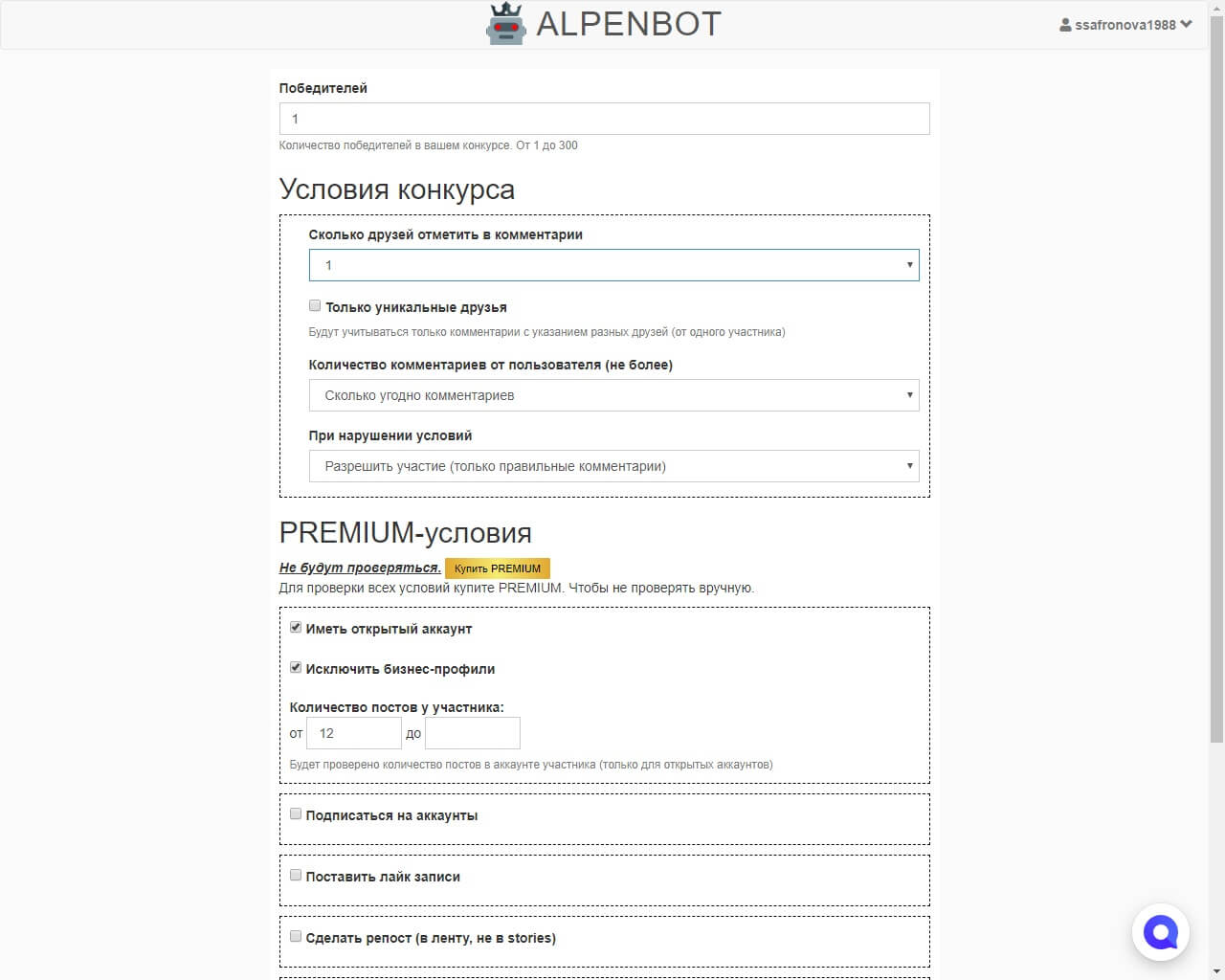


Рисунок 1.1 – Интерфейс портала «AlpenBot»

Рассмотрим еще один аналог – гораздо менее популярный у пользователей онлайн портал «Randstuff.ru», который является генератором чисел для конкурса на YouTube, в Инстаграм, Фейсбук и других соцсетях, а также анализом победителей. Он имеет большой набор генерации, превосходящий возможности большинства других аналогов, но не дотягивает по уровню до полноценного портала для генерации и анализа случайных чисел, например такого как «AlpenBot».

Онлайн портал «Randstuff.ru» имеет следующие преимущества:

* большой набор инструментов.
* генерация чисел в заданном интервале
* имеет простой интерфейс, напоминающий «AlpenBot».
* наличие большого количества возможностей.
* осуществление выбора победителя из списка.
* сохранение результатов в виде скриншота.

У онлайн портала «Randstuff.ru»имеются следующие недостатки:

* отсутствие некоторых полезных функций.
* невозможность открывать сразу несколько генераций.
* отсутствие анализа свойств чисел.
* отсутствие наличия выбора алгоритма генерирования чисел.

На рисунке 1.2 изображен онлайн портал «Randstuff.ru».

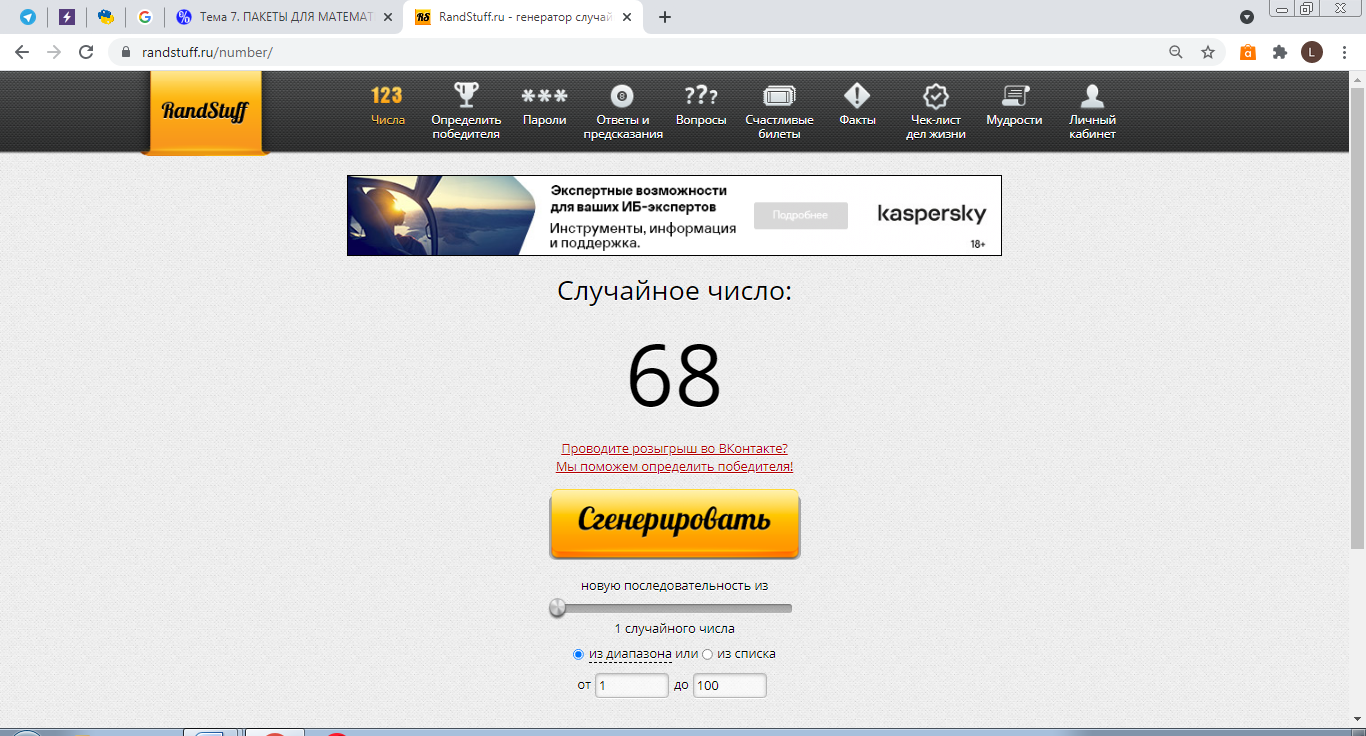


Рисунок 1.2 – Онлайн портал «Randstuff.ru»

Последним аналогом является довольно популярный онлайн портал «Randomus.ru». Зачастую он используется как генератор случайных чисел с возможностью их сортировки для различных лотерей и розыгрышей.

У онлайн портала «Randomus.ru» имеется ряд следующих достоинств:

* полностью бесплатный портал для генерации чисел.
* легкость в использовании.
* простой интерфейс.
* возможность настраивать варианты ответов.
* отсутствие изменить или подменить данные.
* наличие авторизации пользователя.
* возможность экспорта файла в различные форматы.
* возможность долговременного хранения результатов, у которого будет постоянная ссылка и готовая картинка.

Рассмотрим недостатки онлайн портала «Randomus.ru»:

* немного старомодный интерфейс.
* наличие рекламы.
* требовательность к системным ресурсам на некоторых компьютерах.
* итоги можно подводить только один раз.

На рисунке 1.3 изображен онлайн портал «Randomus.ru».

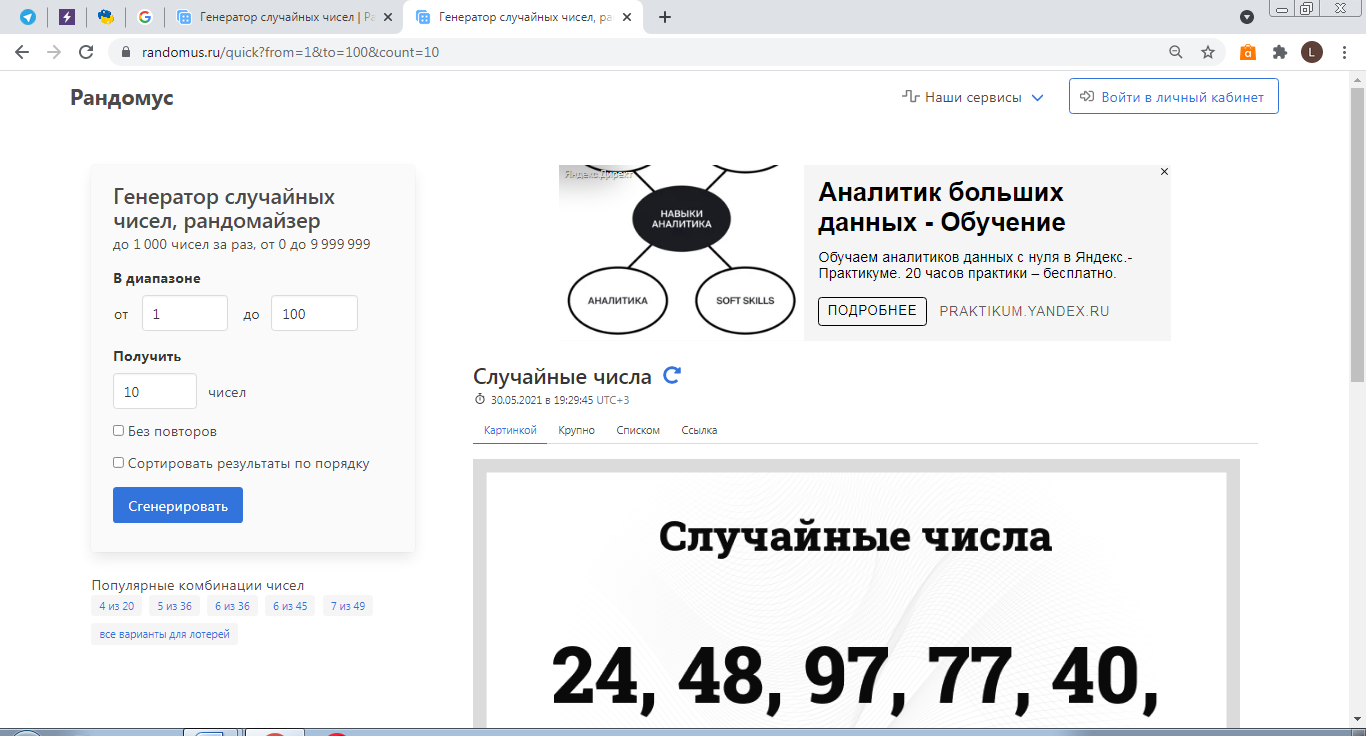


Рисунок 1.3 – Интерфейс онлайн портала «Randomus.ru»

В результате изучения аналогов курсового проекта «Генерирование случайных чисел и их анализ» был сформирован набор функций и особенностей работы, которые должны присутствовать в разрабатываемой программе.

## 1.3 Функциональное назначение

Курсовой проект «Генерирование случайных чисел и их анализ» предназначен для знакомства c различными алгоритмами генерации случайных чисел, в которой должны быть реализованы возможности генерации абсолютно случайных чисел, а также с изучением свойств этих чисел.

Компьютерная версия курсового проекта «Генерирование случайных чисел и их анализ» предоставляет все ее возможности и удобный интерфейс, позволяющий автоматизировать параллельные процессы программного средства. Приложение предназначено для оказания помощи при работе со случайными числами, возможно, приобретении новых знаний и совершенствовании старых в предметной области.

Основные задачи курсового проекта «Генерирование случайных чисел и их анализ»:

* предоставление максимально удобных условий работы со случайными числами.
* предоставление важного и полезного анализа.
* максимально возможная интерактивность и качественное графическое оформление.
* стабильное совершенствование и закрепление результатов в работе с анализом случайных чисел.

Основные функции и возможности программы заключаются в следующем:

* пользователь может самостоятельно указать любой диапазон для генерирования случайных чисел;
* пользователь может выбрать один или несколько алгоритмов генерирования чисел, которые в последствии будут показаны на экране;
* пользователю предоставляется возможность закрывать программу, когда ему будет это удобно;
* информационная функция – программа позволяет наглядно рассмотреть и изучить подробный анализ чисел с их примерами;
* обучающая функция – программа способствует изучению различных способов генерирования чисел, впоследствии чего рассматривать и изучать подробный анализ данного множества чисел;
* возможность «сбрасывать» данные, для реализации нового анализа чисел.
* пользователь имеет возможность из окна инструкции анализу чисел возвращаться в главное меню.

# 2 Проектирование задачи

## 2.1 Алгоритм работы программы

Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результа. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Независимые инструкции могут выполняться в произвольном порядке, параллельно, если это позволяют используемые исполнители.

Требуется разработать курсовой проект «Генерирование случайных чисел и их анализ». Этот программный продукт предназначен для приобретения базовых знаний генерации чисел различными способами и проведение анализа между этими числами. Программное средство может быть использовано различными категориями пользователей.

Алгоритм работы программы заключается в следующем.

Если дважды кликнуть по исполняемому файлу «Kurs.exe» левой кнопкой мыши, откроется окно программы. В окне расположены следующие компоненты:

## «textBox20» для ввода диапазона начального числа;

## «textBox21» для ввода диапазона конечного числа;

## «button1» для выбора генерирования случайных чисел первым и вторым алгоритмом;

## «button4» для выбора генерирования случайных чисел третьим алгоритмом;

## «button3» для очистки введенных данных и результата;

## «textBox8» для вывода сгенерированных чисел первым алгоритмом;

## «textBox1» для вывода сгенерированных чисел вторым алгоритмом;

## «textBox25» для вывода сгенерированных чисел третьим алгоритмом;

## «textBox2»,«textBox3»,«textBox6»,«textBox7» для вывода результата проверки на чётность чисел, сгенерированных первым алгоритмом;

## «textBox9»,«textBox22»,«textBox23»,«textBox24» для вывода результата проверки на чётность чисел, сгенерированных вторым алгоритмом;

## «textBox26»,«textBox27»,«textBox28»,«textBox29» для вывода результата проверки на чётность чисел, сгенерированных третьим алгоритмом;

## «textBox4»,«textBox5» для вывода результата нахождения минимального и максимального чисел, сгенерированных первым алгоритмом;

## «textBox30»,«textBox31» для вывода результата нахождения минимального и максимального чисел, сгенерированных вторым алгоритмом;

## «textBox32»,«textBox33» для вывода результата нахождения минимального и максимального чисел, сгенерированных третьим алгоритмом;

## «textBox10»,«textBox11» для вывода результата нахождения простых и составных чисел, сгенерированных первым алгоритмом;

## «textBox34»,«textBox35» для вывода результата нахождения простых и составных чисел, сгенерированных вторым алгоритмом;

## «textBox36»,«textBox37» для вывода результата нахождения простых и составных чисел, сгенерированных третьим алгоритмом;

## «textBox48»,«textBox47» для вывода результата нахождения положительных и отрицательных чисел, сгенерированных первым алгоритмом;

## «textBox19»,«textBox40» для вывода результата нахождения положительных и отрицательных чисел, сгенерированных вторым алгоритмом;

## «textBox43»,«textBox46» для вывода результата нахождения положительных и отрицательных чисел, сгенерированных третьим алгоритмом;

## «textBox16» для вывода результата нахождения повторяющихся чисел, сгенерированных первым алгоритмом;

## «textBox38» для вывода результата нахождения повторяющихся чисел, сгенерированных вторым алгоритмом;

## «textBox39» для вывода результата нахождения повторяющихся чисел, сгенерированных третьим алгоритмом;

## «textBox17»,«textBox18» для вывода результата нахождения однозначных и двузначных чисел, сгенерированных первым алгоритмом;

## «textBox41»,«textBox42» для вывода результата нахождения однозначных и двузначных чисел, сгенерированных вторым алгоритмом;

## «textBox44»,«textBox45» для вывода результата нахождения однозначных и двузначных чисел, сгенерированных третьим алгоритмом.

При запуске программы откроется основное окно с генерированием и анализом чисел, которое является главной и единственной рабочей областью. Далее пользователь может начать выбирать диапазон для генерирования случайных чисел. Числа в диапазоне могут быть совершенно любыми, как положительными, так и отрицательными. Главная особенность, что числа должна были целыми. Далее пользователь имеет возможность выбрать один из предложенных алгоритмов генерирования чисел. В зависимости от выбранного алгоритма, выводится ряд случайно сгенерированных чисел. Если пользователь вводит числовое значение не корректно, выводятся ошибки в окне программы. Если какое-то поле в диапазоне не заполнено, в окне программы выводится ошибка и программа завершает свою работу. Параллельно с получением ряда случайно сгенерированных чисел, мы получаем подробный их анализ. В процессе деятельности пользователь может в любой момент сохранить и сбросить свои изменения. Параллельно с математическими вычислениями пользователь может просмотреть и изучить инструкцию к анализу множества чисел.

Основной принцип работы программы показан в блок-схеме (приложение А).

## 2.2 Выбор и обоснование инструментов разработки

Для реализации поставленной задачи была выбрана среда разработки Microsoft Visual Studio 2019, так как она наиболее подходит для создания курсового проекта «Генерирование случайных чисел и их анализ», а именно из-за привычного и понятного интерфейса, большого набора функций и инструментов.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Продукты Visual Studio позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и веб-службы.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб–редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно–ориентированных языках программирования или инструментов для прочих аспектов процесса разработки программного обеспечения.

На рисунке 2.1 изображен Microsoft Visual Studio 2019.

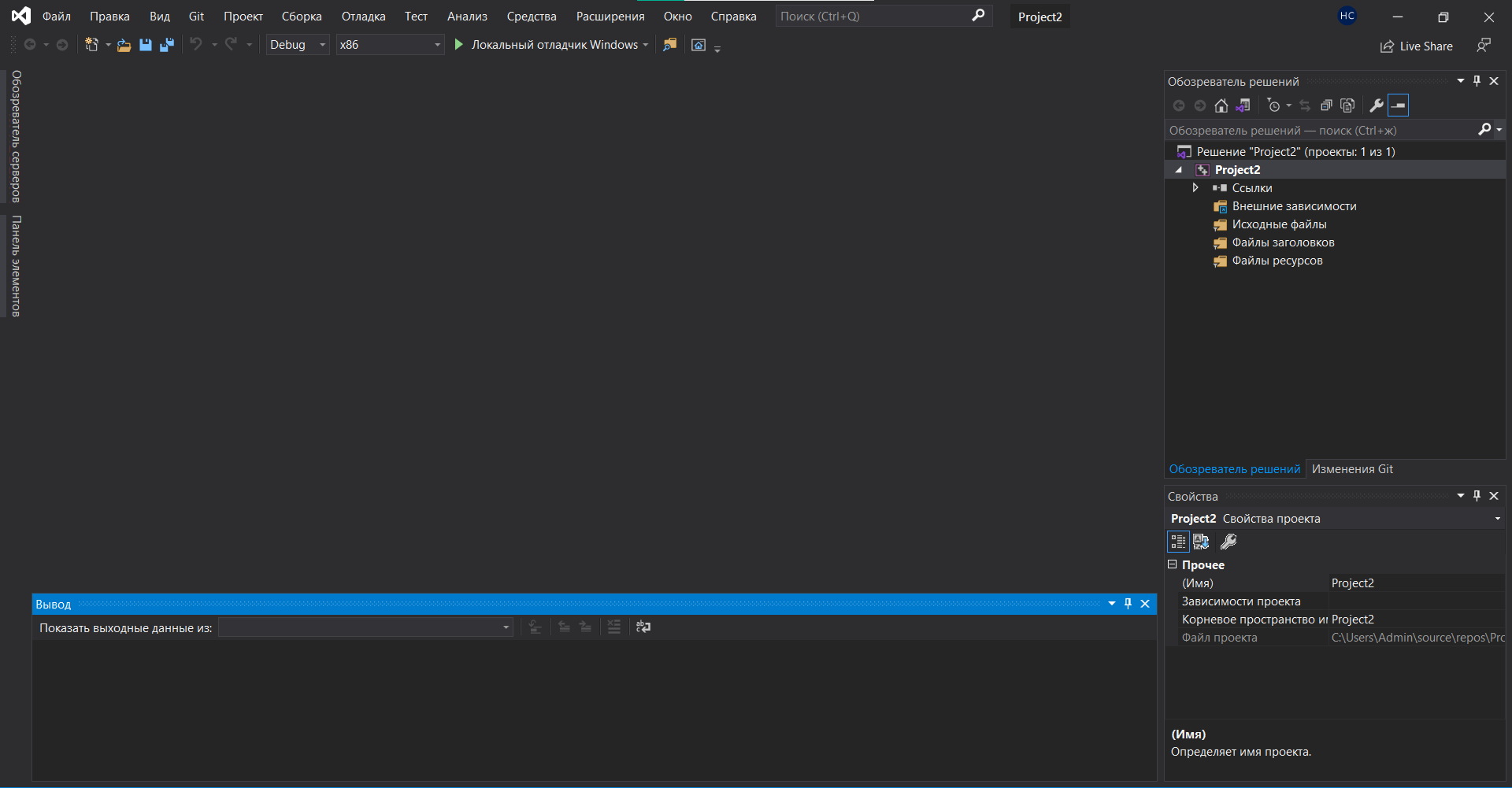


Рисунок 2.1 – Microsoft Visual Studio 2019

Курсовой проект «Генерирование случайных чисел и их анализ» разработан на языке программирования C++. Этот язык использовался как основной и единственный в процессе разработки. Он является уже хорошо знакомым языком, имеющий много преимуществ, например скорость и функциональность. Язык программирования С++ является весьма мощным языком, он спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Также стоит отметить возможность создания графического интерфейса, а именно наличие технологии Windows Forms.

Технология Windows Forms проста в освоении, имеет много готовых классов и объектов как общего, так и специального назначения. Интегрированная среда разработки Microsoft Visual Studio 2019 отлично подходит для создания и работы с графическими элементами. Компоненты формы можно располагать на ней в интерактивном режиме, просто перетаскивая их из списка.

Наличие компонента Windows Forms стало одним из ключевых принципов выбора именно IDE Visual Studio 2019. Кроме этого, у среды разработки достаточно удобно реализован интерфейс вкладок, предназначенных для настройки Visual Studio 2019. Удобной функцией может стать возможность дополнительной установки компонентов Visual Studio прошлых версий.

# 3 Программная реализация

# 3.1 Описание алгоритмов решения задачи

Алгоритм – набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результа. В старой трактовке вместо слова «порядок» использовалось слово «последовательность», но по мере развития параллельности в работе компьютеров слово «последовательность» стали заменять более общим словом «порядок». Независимые инструкции могут выполняться в произвольном порядке, параллельно, если это позволяют используемые исполнители.

Программа начинаеися с ввода диапазона начального и конечного числовых значений. Далее пользователь выбирает любой из трёх предложенных алгоритмов генериррования чисел. После чего получает ряд сгенерированных чисел параллельно с их подробным анализом.

Алгоритм проверки чисел на чётность:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Далее идёт простая математическая проверка на чётность. Если число делится на 2 без остатка, оно является чётным. В противном случае число будет являться нечётным.

В ячейки будет выводиться количество чётных и нечётных чисел,а также будут выводиться сами эти числа.

Алгоритм нахождения максимального и минимального чисел:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Вводятся определённые переменные, отвечающие за минимальное и максимальное число. Будет производиться прохождение по всему массиву чисел. Если максимальный элемент меньше элемента массива, то оно присваивается этому лементу массива. Аналогично будет с поиском минимального элемента.

В ячейки выводятся сами минимальные и максимальные числа.

Алгоритм поиска простых и составных чисел:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Если данный элемент массива делится только на единицу и на само себя, то оно является простым. Во всех остальных случаях число будет составным.

В ячейки будет выводиться количество простых и составных чисел,а также будут выводиться сами эти числа.

Алгоритм поиска положительных и отрицательных чисел:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Если элемент массива больше или равен нулю, он будет являиться положительным. В противном случае, число будет отрицательным.

В ячейки будет выводиться количество положительных и отрицательных чисел,а также будут выводиться сами эти числа.

Алгоритм поиска повторяющихся чисел:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Далее идёт прохождение по уже заполненным элементам массива, где сравнивается каждый с введенным значением.

Таким образом, если находится совпадение, то в ячейку выводятся повторяющиеся числа.

Алгоритм проверки однозначных и двузначных чисел:

Ряд сгенерированных чисел записывается в массив, где каждое число это элемент массива. Если элемент попадает в диапазон однозначных чисел, а это от -9 до 9 включительно, то он является однозначным. Если же элемент попадает в диапазон двузначных чисел от -99 до 99 включительно, но при этом не включая диапазон однозначных чисел, то он является двузначным.

В ячейки будут выводиться однозначные и двузначные числа.

# 3.2 Проектирование интерфейса приложения

Рассмотрим сам термин интерфейс. Интерфейс – это совокупность средств и правил, обеспечивающих взаимодействие устройств вычислительной системы и программ, а также взаимодействие с человеком. Основные задачи, для решения которых он предназначен:

* ввод и отображение информации (звук, изображение);
* управление отдельными приложениями;
* обмен данными с другими устройствами;
* взаимодействие с операционной системой.

Пользовательский интерфейс объединяет в себе все элементы и компоненты программы, которые способны оказать влияние на взаимодействие пользователя с программным обеспечением.

Интерфейс должен быть простым и интуитивно понятным рядовому пользователю. Все надписи должны быть чётко видны, элементы не должны перекрывать друг друга.

Приложение было решено разрабатывать в ярких и светлых тонах. Макет интерфейса главного окна представлен ниже (см. рисунок 3.2).

На форме располагаются следующие функциональные элементы:

* TextBox;
* Label;
* Button.

textBox4

textBox5

textBox31

textBox30

textBox33

textBox32

label26

label25

label22

textBox8

button1

textBox20

textBox21

textBox1

button4

textBox25

label4

label15

label9

label7

textBox16

textBox6

textBox6

textBox10

textBox34

textBox11

textBox35

textBox36

textBox37

textBox22

textBox38

textBox9

textBox24

textBox27

textBox26

textBox9

textBox39

textBox22

label49

label8

label8

textBox18

textBox17

textBox43

textBox46

textBox19

textBox40

textBox47

textBox48

textBox41

textBox42

textBox44

textBox45

Рисунок 3.2 – Макет главного окна программы

# 4 Тестирование

Тестирование – процесс проверки программного обеспечения с целью выявления и предотвращения дефектов, оценки соответствия требованиям заказчика.

Баг (дефект) – ошибка в коде, которая может привести к тому, что программа будет работать не так, как ожидается.

В широком смысле, тестирование – это одна из техник контроля качества (Quality Control), которая включает планирование, составление тестов, непосредственно выполнение тестирования и анализ полученных результатов.

Некоторые ошибки проявляются после первого же запуска программы на выполнение. Они заметны «невооруженным глазом», и для их обнаружения не надо прибегать ни к каким специальным средствам. Некоторые ошибки проявляются в случайные моменты работы программы. С такими ошибками справиться труднее всего. Если вы не можете зафиксировать условия возникновения ошибки, вы не сможете понять причину и устранить ее. Иногда, чтобы добиться устойчивого проявления ошибки, перед каждым стартом программы приходится перезагружать среду разработки.

Всевозможные ошибки обычно подразделяют на три группы:

– синтаксические ошибки;

– ошибки времени выполнения программы;

– смысловые (логические) ошибки;

– рассмотрим каждую из выделенных групп по порядку.

Синтаксические ошибки являются самыми простыми, они устраняются уже на этапе компиляции. Причина их проста и однообразна – неправильная запись служебных слов, операторов.

Намного больше неприятностей доставляют ошибки времени выполнения. Они дают о себе знать прекращением выполнения программы. Чаще всего ошибка времени выполнения является признаком смысловой ошибки.

Смысловые (логические) ошибки – самые сложные и трудноуловимые. Они проявляются в том, что программа выполняет не то, что от нее предполагалось. Последствия смысловых ошибок могут быть самыми разными: это может быть что-то безобидное, например, неправильное содержимое окна, невыполнение или неверное выполнение команд пользователя, неправильное содержимое выходной информации и прочее, а может быть и достаточно серьезными – программа может досрочно завершиться с ошибкой времени выполнения и многое другое.

При исправлении ошибки самое главное – это не внести в программу новых ошибок.

Тестирование программного обеспечения имеет следующие уровни:

* компонентное – поиск дефектов и проверка функциональности (модулей, программ, объектов, классов), которые можно протестировать изолированно;
* интеграционное – проверяет взаимодействие между компонентами, взаимодействие различных частей системы (ОС, файловая система, аппаратное обеспечение);
* системное – тестирование поведения тестового объекта как целостной системы или продукта;
* приемочное, в свою очередь, подразделяется на пользовательское, эксплуатационное, контрактное и правовое, альфа и бета. Заключается в проверке работоспособности системы, частей системы или отдельных нефункциональных характеристик системы.

Также различают следующие типы тестирования:

* функциональное – проверяет, что система делает. Проводится тестирование внешнего поведения ПО (black–box testing);
* нефункциональное – проверяет, как система работает. Сюда входят нагрузочное тестирование, тестирование производительности, стресс–тестирование, тестирование удобства использования (в большинстве случаев black–box testing);
* структурное – тестирование, основанное на анализе внутренней структуры компонента или системы (white–box testing).

Тестирование изменений включает в себя подтверждение и регрессионное тестирование. Подтверждение – проверка того, что дефект действительно исправлен. Регрессионное тестирование – проверка, что ранее работающая корректно функциональность все еще работает.

После того, как ошибка зафиксирована, необходимо найти в исходном тексте программы то место, в котором она возникает. Другими словами, нужно локализировать ошибку. Первое, что приходит в голову, – протестировать программу по шагам с помощью отладчика, выполнить трассировку. Однако сделать это далеко не всегда возможно, особенно в больших и сложных программах. При исправлении ошибки самое главное – не внести в программу новых ошибок.

Тестирование программы проходило в ходе и после ее разработки.

Отладка программы осуществлялась встроенными средствами Microsoft Visual Studio 2019.

По результатам тестирования были выявлены некоторые ошибки в работе программы. В дальнейшем они были исправлены и все имеющиеся функции протестированы снова.

Таблица 4.1 – Тестирование программного средства

|  |  |  |
| --- | --- | --- |
| Тестируемая операция | Требуемый  результат | Полученный  результат |
| Запуск программы | Появление главного рабочего окна программы со всеми его компонентами | После нажатия на значок программы (открытия приложения) появилось окно программы со всеми компонентами |
| Ввод неверных (не числовых) символов в поля для диапазона числовых значений | Сообщение об ошибке, остутствие неверно введенных символов | Программа вылетает, появление сведений об ошибке |
| Нажатие на одну из кнопок генерирования чисел и их анализа | Появление сгенерированных чисел параллельно с заполненными полями для анализа случайно сгенерированных чисел | Появление в поле сгеенрированного множества чисел, а также появление в ячейках анализа данных чисел |
| Нажатие на одну из кнопок генерирования и анализа чисел с незаполненными полями для диапазона чисел | Появление надписи об ошибке в окнах результата | Программа вылетает, появление сведений об оишбке |
| Нажатие на кнопку «Сброс» | Очистка всех полей | Очистка всех полей |
| Нажатие на главное меню «Меню» | Появление возможных дополнительных функций для программы | Появление возможных дополнительных функций для программы |
| Нажатие на кнопку «Подробнее об анализе» | Появление нового окна с инструкцией к анализу множества чисел для дополнительного изучения информации | Повяление нового окна с инстукцией к анализу множества чисел. |
| Завершение программы | Закрытие формы программы | Программа завершила свою работу |

# 5 Руководство пользователя

# Для запуска программы необходимо дважды нажать на файл с названием «Kurs.exe». После этого на экране появится главное окно (см. рисунок 5.1).

# 

Рисунок 5.1 – Главное окно программы «Kurs»

Далее предлагается ввести два n-разрядных числа в поля. Это и будет диапазон для последующего генерирования чисел (см. рисунок 5.2).

# 

Рисунок 5.2 – Поля для ввода диапазона чисел

При попытке ввести некорректный символ, программа будет вылетать и сообщать об ошибке (см. рисунок 5.3).

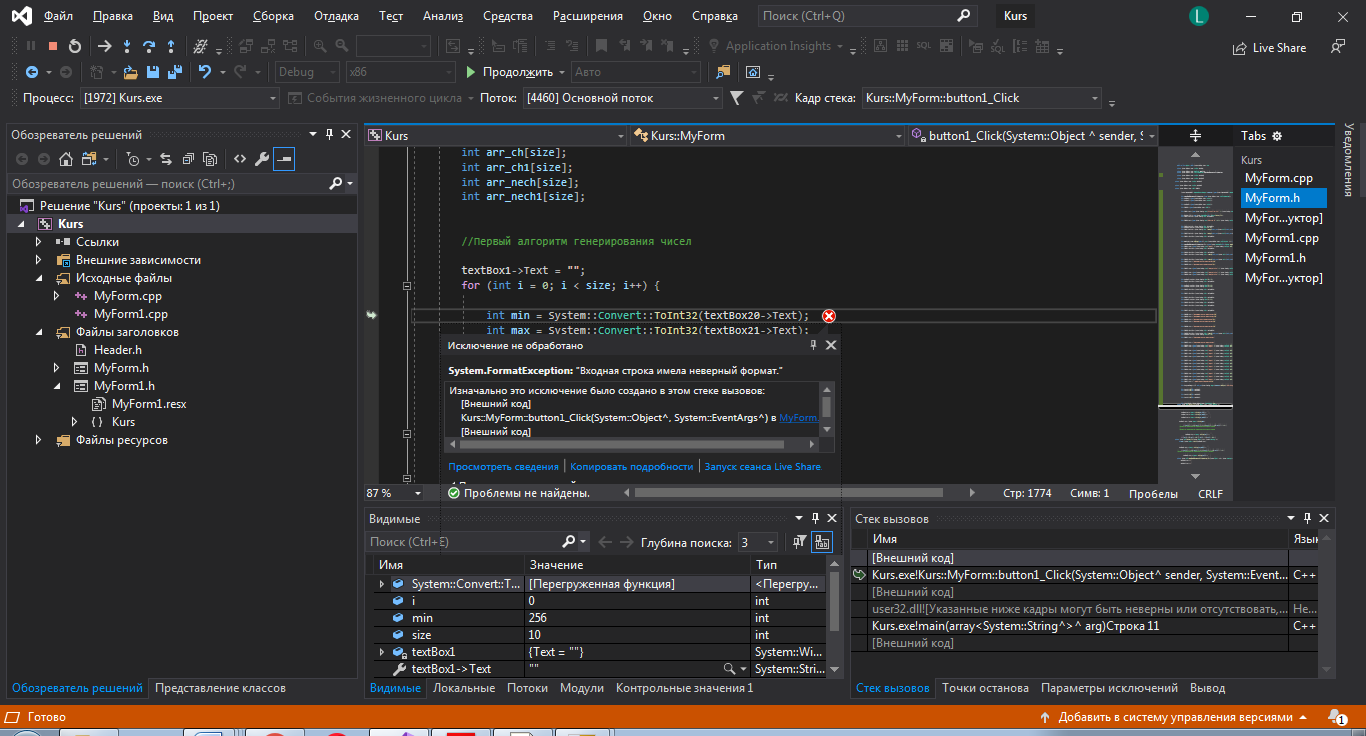


Рисунок 5.3 – Сведения об ошибке при некорректном вводе

Следующим шагом пользователь может выбрать любой из предложенных алгоритмов генерирования чисел и получить ряд данного множества чисел в ячейках (см. рисунок 5.4).

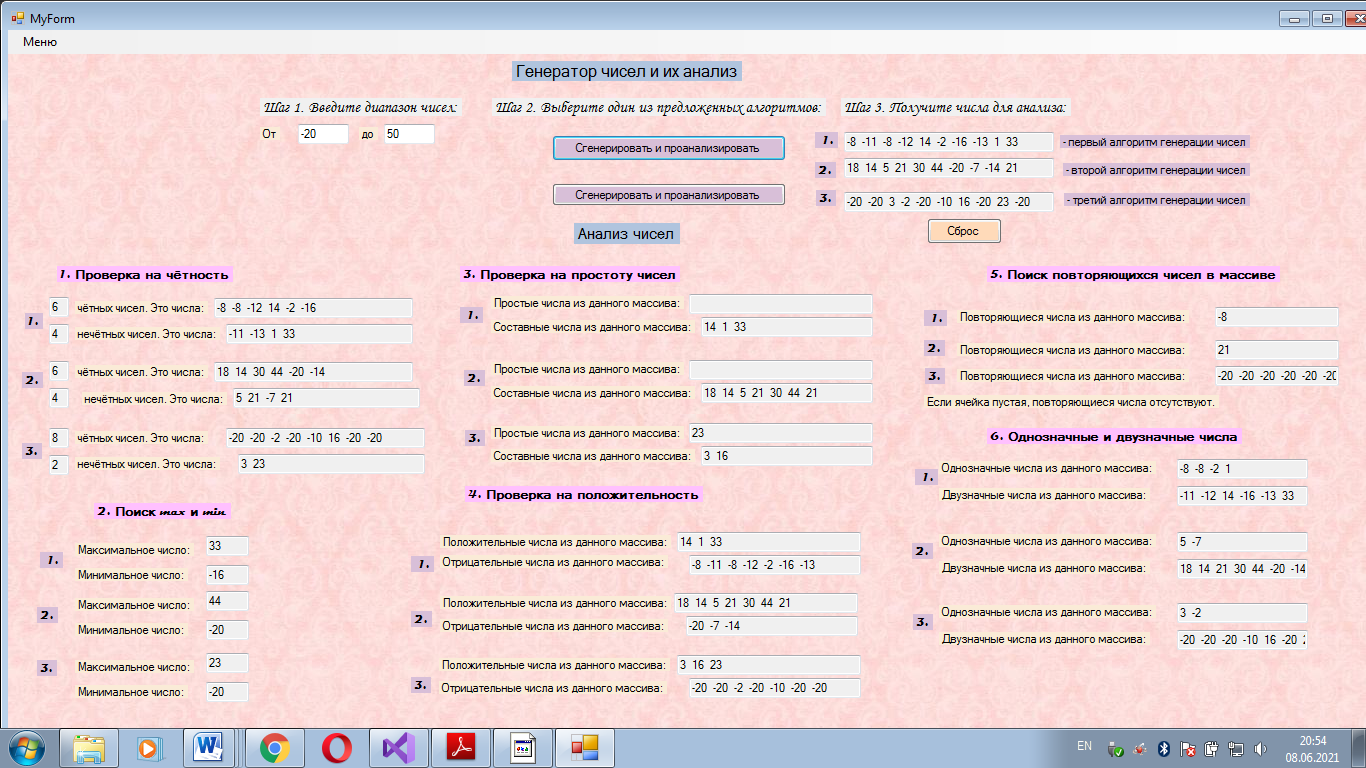


Рисунок 5.4 – Получения ряда чисел различными алгоритмами генерации

Параллельно с генерированием чисел, пользователь получает подробный анализ ряда чисел. Анализ включает в себя проверки чисел на чётность, поиск максимальных и минимальных чисел из данного множества, нахождение простых и составных чисел, определение положительных и отрицательных чисел, поиск повторяющихся чисел в массиве, а также возможность нахождения однозначных и двузначных чисел (см. рисунок 5.5).

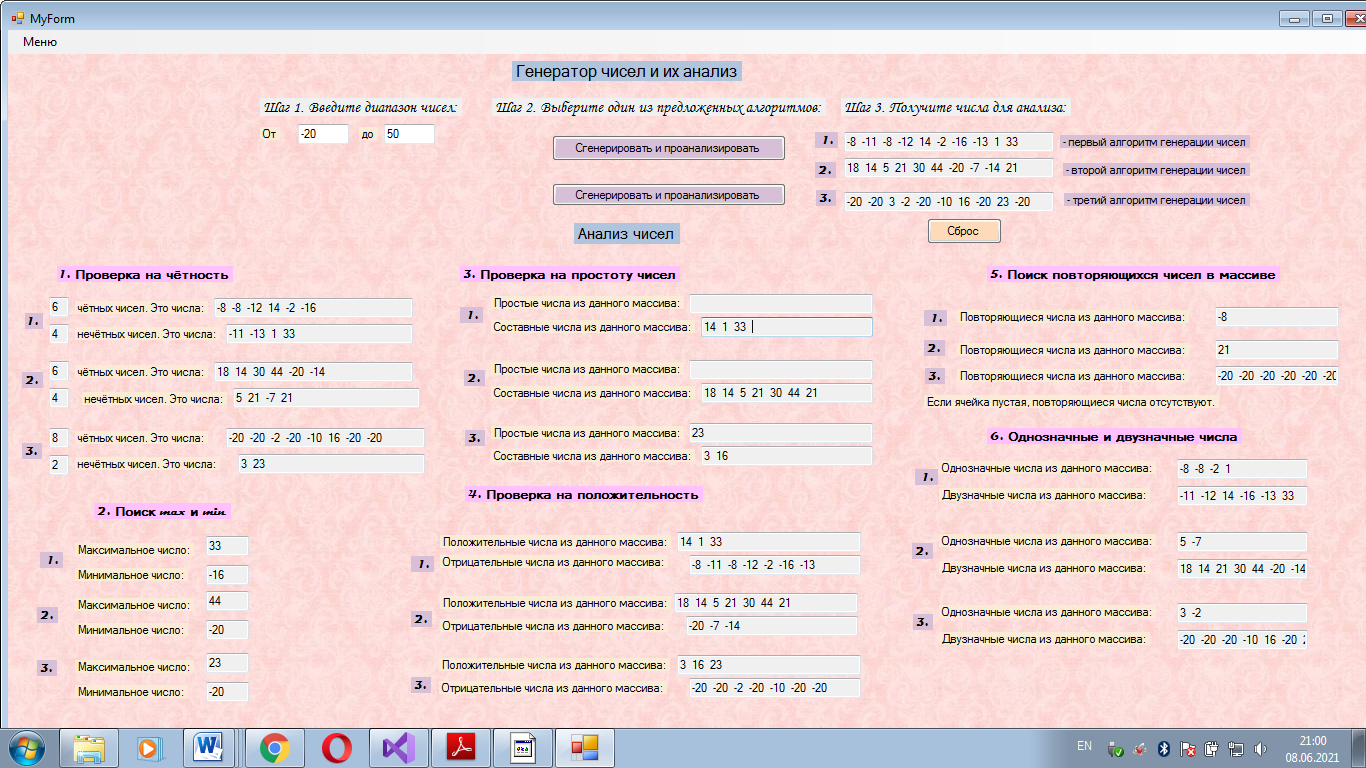


Рисунок 5.5 – Интерфейс анализа сгенерированного множества чисел

При желании пользователь может открыть инструкцию «Подробнее об анализе», которая располагается в главном «Меню» (см. рисунок 5.6).

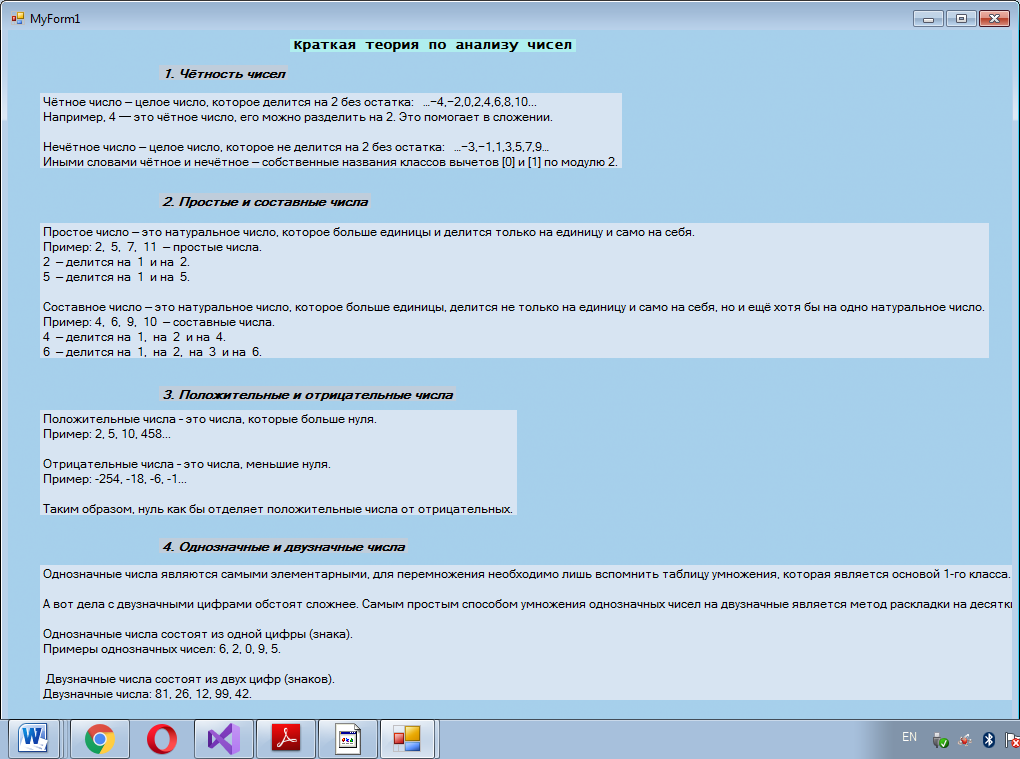


Рисунок 5.6 – Инструкция к анализу чисел

Для очистки всех полей необходимо нажать на кнопку «Сброс».

Для закрытия программы необходимо нажать на крест в правом верхнем углу или открыть «Меню» и нажать на «Закрыть».

# Заключение

В ходе выполнения курсового проекта была достигнута цель проектирования – разработан курсовой проект «Генерирование случайных чисел и их анализ» с графической реализацией. Все компоненты в ходе реализации были обёрнуты в класс языка C++ для более удобной работы над структурой.

Затем, с помощью Windows Forms, была реализована графическая оболочка для визуализации структуры данных и более удобного ввода значений в поля, предназначенные для этого. После нажатий кнопок будут происходить все необходимые действия, предусмотренные программой.

В пояснительной записке выполнены анализ задачи, проектирование и реализация программных средств, которые являются необходимыми этапами разработки программного обеспечения.

Курсовой проект «Генерирование случайных чисел и их анализ» помогает закрепить полученные знания на протяжения всего курса обучения, улучшить навыки решения поставленных задач, дает возможность научиться поиску необходимой информации для облегчения решения поставленной задачи.

Важным достижением при разработке курсового проекта «Генерирование случайных чисел и их анализ» стала успешная работа всех запланированных функций, особенно удобность интерфейсов дополнительных окон. Вместе с курсовым проектом «Генерирование случайных чисел и их анализ» идет встроенная в программу подробная инструкция, описывающая назначение и принципы использования соответствующих функций.

Таким образом, в результате разработки курсового проекта «Генерирование случайных чисел и их анализ» были достигнуты все поставленные цели и реализованы предусмотренные задачи. Удалось создать программу с удобным графическим интерфейсом, сочетающимся со всеми ее возможностями. Для удобства работы использовались автоматизированные и стабильно работающие функции. Также в ходе разработки было изучено много новой информации, компонентов и свойств Visual Studio 2019 и языка программирования C++, использовавшихся при работе над курсовым проектом «Генерирование случайных чисел и их анализ».

# Список использованных источников

Литература

1. Албахари, Джозеф C++ 3.0. Справочник / Джозеф Албахари , Бен Албахари. М.: БХВПетербург, 2012. 944 c.

2. Медведьев В.И. Особенности объектно-ориентированного программирования на C++/CLI, C# и Java. / В.И. Медведьев – 2-е изд. – Казань: РИЦ «Школа», 2010. -444 c.

3. Лафоре Роберт Объектно-ориентированное программирование в C++ / Роберт Латофе. – Питер: Санкт-Петербург. 2019. – 928 с.

4. Доусон Майкл Изучаем C++ через программирование игр / Майкл Доунсон. – Питер: Санкт-Петербург. 2020. – 352 с.

5. Вагнер, Билл С++Эффективное программирование / Билл Вагнер. М.: ЛОРИ, 2013. 320 c.

Интернет-ресурсы

6. Интернет ресурс DrawIo [электронный ресурс] Режим доступа: <https://app.diagrams.net/#G1pmLYmOM6D9bOF9xbtPqSsZu72U3mEjAk>. – 29.05.2021

7. Интернет ресурс Argon online [электронный ресурс] Режим доступа: <http://argon.pro/windows/other/ssavers>. – 17.04.2021

8. Интернет ресурс SMMplanner [электронный ресурс] Режим доступа: <https://smmplanner.com/blog/top-randomaizierov-dlia-konkursov-v-sotssietiakh>. – 19.05.2021

# **ПРИЛОЖЕНИЕ А**

(обязательное)

**Код программы**

**MyForm.cpp:**

#include "MyForm.h"

#include "string"

using namespace System;

using namespace System::Windows::Forms;

[STAThread]

void main(cli::array<String^>^ arg) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Kurs::MyForm form; //WinFormsTest - имя вашего проекта

Application::Run(% form);

}

**MyForm.h:**

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

const int size = 10;

int arr[size];

int arr1[size];

int arr\_ch[size];

int arr\_ch1[size];

int arr\_nech[size];

int arr\_nech1[size];

//Первый алгоритм генерирования чисел

textBox1->Text = "";

for (int i = 0; i < size; i++) {

int min = System::Convert::ToInt32(textBox20->Text);

int max = System::Convert::ToInt32(textBox21->Text);

static unsigned seed = 1; // зерно не должно быть 0

seed = (seed \* 73129 + 95121) % 100000;

int schet = seed % 123;

int y = min;

while (schet > 0) {

schet--;

y++;

if (y == max) {

y = min;

}

}

textBox1->Text += Convert::ToString(y) + " ";

arr1[i] = y;

}

//Второй алгоритм генерирования чисел

textBox8->Text = "";

Random^ rnd = gcnew Random();

for (int i = 0; i < size; i++) {

int min = System::Convert::ToInt32(textBox20->Text);

int max = System::Convert::ToInt32(textBox21->Text);

int x = rnd->Next(min, max);

textBox8->Text += Convert::ToString(x) + " ";

arr[i] = x;

}

//Проверка на чётность для первого алгоритма

int sch = 0;

for (int i = 0; i < size; i++) {

if (arr[i] % 2 == 0) {

arr\_ch[sch] = arr[i];

sch++;

}

}

textBox2->Text += Convert::ToString(sch);

for (int i = 0; i < sch; i++) {

textBox6->Text += Convert::ToString(arr\_ch[i]) + " ";

}

sch = 0;

for (int i = 0; i < size; i++) {

if (arr[i] % 2 != 0) {

arr\_nech[sch] = arr[i];

sch++;

}

}

textBox3->Text += Convert::ToString(sch);

for (int i = 0; i < sch; i++) {

textBox7->Text += Convert::ToString(arr\_nech[i]) + " ";

}

//Проверка на чётность для второго алгоритма

int tsch = 0;

for (int i = 0; i < size; i++) {

if (arr1[i] % 2 == 0) {

arr\_ch1[tsch] = arr1[i];

tsch++;

}

}

textBox24->Text += Convert::ToString(tsch);

for (int i = 0; i < tsch; i++) {

textBox22->Text += Convert::ToString(arr\_ch1[i]) + " ";

}

tsch = 0;

for (int i = 0; i < size; i++) {

if (arr1[i] % 2 != 0) {

arr\_nech1[tsch] = arr1[i];

tsch++;

}

}

textBox23->Text += Convert::ToString(tsch);

for (int i = 0; i < tsch; i++) {

textBox9->Text += Convert::ToString(arr\_nech1[i]) + " ";

}

//Поиск максимум и минимум для первого алгоритма

int min = arr[0];

int max = arr[0];

for (int i = 1;i < size;i++) {

if (max < arr[i]) {

max = arr[i];

}

}

textBox4->Text = System::Convert::ToString(max);

for (int i = 1;i < size;i++) {

if (min > arr[i]) {

min = arr[i];

}

}

textBox5->Text = System::Convert::ToString(min);

//Поиск максимум и минимум для второго алгоритма

int mini = arr1[0];

int maxi = arr1[0];

for (int i = 1;i < size;i++) {

if (maxi < arr1[i]) {

maxi = arr1[i];

}

}

textBox31->Text = System::Convert::ToString(maxi);

for (int i = 1;i < size;i++) {

if (mini > arr1[i]) {

mini = arr1[i];

}

}

textBox30->Text = System::Convert::ToString(mini);

//Проверка на простоту для первого алгоритма

for (int j = 0;j < size;j++) {

bool check = true;

if (arr[j] > 0) {

for (int i = 2;i < arr[j];i++) {

if (arr[j] % i == 0) {

check = false;

}

}

if (check) {

textBox10->Text += Convert::ToString(arr[j]) + " ";

}

else textBox11->Text += Convert::ToString(arr[j]) + " ";

}

}

//Проверка на простоту для второго алгоритма

for (int j = 0;j < size;j++) {

bool check = true;

if (arr1[j] > 0) {

for (int i = 2;i < arr1[j];i++) {

if (arr1[j] % i == 0) {

check = false;

}

}

if (check) {

textBox34->Text += Convert::ToString(arr1[j]) + " ";

}

else textBox35->Text += Convert::ToString(arr1[j]) + " ";

}

}

//Проверка на положительное или отрицательное число первого алгоритма

int sch1 = 0;

for (int i = 0; i < size; i++) {

if (arr[i] >= 0) {

arr\_ch[sch1] = arr[i];

sch1++;

}

}

textBox12->Text += Convert::ToString(sch1);

for (int i = 0; i < sch1; i++) {

textBox48->Text += Convert::ToString(arr\_ch[i]) + " ";

}

sch1 = 0;

for (int i = 0; i < size; i++) {

if (arr[i] < 0) {

arr\_nech[sch1] = arr[i];

sch1++;

}

}

textBox14->Text += Convert::ToString(sch1);

for (int i = 0; i < sch1; i++) {

textBox47->Text += Convert::ToString(arr\_nech[i]) + " ";

}

//Проверка на положительное или отрицательное число второго алгоритма

int tsch1 = 0;

for (int i = 0; i < size; i++) {

if (arr1[i] >= 0) {

arr\_ch1[tsch1] = arr1[i];

tsch1++;

}

}

textBox12->Text += Convert::ToString(tsch1);

for (int i = 0; i < tsch1; i++) {

textBox40->Text += Convert::ToString(arr\_ch1[i]) + " ";

}

tsch1 = 0;

for (int i = 0; i < size; i++) {

if (arr1[i] < 0) {

arr\_nech1[tsch1] = arr1[i];

tsch1++;

}

}

textBox14->Text += Convert::ToString(tsch1);

for (int i = 0; i < tsch1; i++) {

textBox19->Text += Convert::ToString(arr\_nech1[i]) + " ";

}

//Проверка на повторение чисел первого алгоритма

for (int i = 0;i < size;i++)

{

for (int j = i + 1;j < size; j++)

{

if (arr[i] == arr[j])

textBox16->Text += Convert::ToString(arr[i]) + " ";

}

}

//Проверка на повторение чисел второго алгоритма

for (int i = 0;i < size;i++)

{

for (int j = i + 1;j < size; j++)

{

if (arr1[i] == arr1[j])

textBox38->Text += Convert::ToString(arr1[i]) + " ";

}

}

//Проверка чисел на количество знаков первого алгоритма

for (int i = 0; i < size; i++) {

if (arr[i] > -10 && arr[i] < 10) {

textBox17->Text += Convert::ToString(arr[i]) + " ";

}

}

for (int i = 0; i < size; i++) {

if ((arr[i]<-9 && arr[i] >-100) || (arr[i] > 9 && arr[i] < 100)) {

textBox18->Text += Convert::ToString(arr[i]) + " ";

}

}

//Проверка чисел на количество знаков второго алгоритма

for (int i = 0; i < size; i++) {

if (arr1[i] > -10 && arr1[i] < 10) {

textBox42->Text += Convert::ToString(arr1[i]) + " ";

}

}

for (int i = 0; i < size; i++) {

if ((arr1[i]<-9 && arr1[i] >-100) || (arr1[i] > 9 && arr1[i] < 100)) {

textBox41->Text += Convert::ToString(arr1[i]) + " ";

}

}

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = "";

const int size = 10;

int arr2[size];

int arr\_ch2[size];

int arr\_nech2[size];

// Третий алгоритм генерирования чисел

srand(time(0));

mt19937 engine;

engine.seed(time(nullptr));

for (int i = 0; i < size; i++) {

int min = System::Convert::ToInt32(textBox20->Text);

int max = System::Convert::ToInt32(textBox21->Text);

int schet = engine();

int z = min;

while (schet > 0) {

schet--;

z++;

if (z == max) {

z = min;

}

}

textBox25->Text += Convert::ToString(z) + " ";

arr2[i] = z;

}

//Проверка на чётность для третьего алгоритма

int trsch = 0;

for (int i = 0; i < size; i++) {

if (arr2[i] % 2 == 0) {

arr\_ch2[trsch] = arr2[i];

trsch++;

}

}

textBox29->Text += Convert::ToString(trsch);

for (int i = 0; i < trsch; i++) {

textBox27->Text += Convert::ToString(arr\_ch2[i]) + " ";

}

trsch = 0;

for (int i = 0; i < size; i++) {

if (arr2[i] % 2 != 0) {

arr\_nech2[trsch] = arr2[i];

trsch++;

}

}

textBox28->Text += Convert::ToString(trsch);

for (int i = 0; i < trsch; i++) {

textBox26->Text += Convert::ToString(arr\_nech2[i]) + " ";

}

//Поиск максимум и минимум для третьего алгоритма

int minim = arr2[0];

int maxim = arr2[0];

for (int i = 1;i < size;i++) {

if (maxim < arr2[i]) {

maxim = arr2[i];

}

}

textBox33->Text = System::Convert::ToString(maxim);

for (int i = 1;i < size;i++) {

if (minim > arr2[i]) {

minim = arr2[i];

}

}

textBox32->Text = System::Convert::ToString(minim);

//Проверка на простоту для третьего алгоритма

for (int j = 0;j < size;j++) {

bool check = true;

if (arr2[j] > 0) {

for (int i = 2;i < arr2[j];i++) {

if (arr2[j] % i == 0) {

check = false;

}

}

if (check) {

textBox36->Text += Convert::ToString(arr2[j]) + " ";

}

else textBox37->Text += Convert::ToString(arr2[j]) + " ";

}

}

//Проверка на положительное или отрицательное число третьего алгоритма

int trsch2 = 0;

for (int i = 0; i < size; i++) {

if (arr2[i] >= 0) {

arr\_ch2[trsch2] = arr2[i];

trsch2++;

}

}

textBox12->Text += Convert::ToString(trsch2);

for (int i = 0; i < trsch2; i++) {

textBox46->Text += Convert::ToString(arr\_ch2[i]) + " ";

}

trsch2 = 0;

for (int i = 0; i < size; i++) {

if (arr2[i] < 0) {

arr\_nech2[trsch2] = arr2[i];

trsch2++;

}

}

textBox14->Text += Convert::ToString(trsch2);

for (int i = 0; i < trsch2; i++) {

textBox43->Text += Convert::ToString(arr\_nech2[i]) + " ";

}

//Проверка на повторение чисел третьего алгоритма

for (int i = 0;i < size;i++)

{

for (int j = i + 1;j < size; j++)

{

if (arr2[i] == arr2[j])

textBox39->Text += Convert::ToString(arr2[i]) + " ";

}

}

//Проверка чисел на количество знаков третьего алгоритма

for (int i = 0; i < size; i++) {

if (arr2[i] > -10 && arr2[i] < 10) {

textBox45->Text += Convert::ToString(arr2[i]) + " ";

}

}

for (int i = 0; i < size; i++) {

if ((arr2[i]<-9 && arr2[i] >-100) || (arr2[i] > 9 && arr2[i] < 100)) {

textBox44->Text += Convert::ToString(arr2[i]) + " ";

}

}

}

private: System::Void закрытьToolStripMenuItem1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

private: System::Void подробнееОбАнализеToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

MyForm1^ M1 = gcnew MyForm1(); //создаём новый экземпляр

M1->Show();//открываем вторую форму

}

private: System::Void button3\_Click\_1(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = L"";

textBox2->Text = L"";

textBox3->Text = L"";

textBox4->Text = L"";

textBox5->Text = L"";

textBox6->Text = L"";

textBox7->Text = L"";

textBox8->Text = L"";

textBox9->Text = L"";

textBox10->Text = L"";

textBox11->Text = L"";

textBox12->Text = L"";

textBox13->Text = L"";

textBox14->Text = L"";

textBox15->Text = L"";

textBox16->Text = L"";

textBox17->Text = L"";

textBox18->Text = L"";

textBox19->Text = L"";

textBox22->Text = L"";

textBox23->Text = L"";

textBox24->Text = L"";

textBox25->Text = L"";

textBox26->Text = L"";

textBox27->Text = L"";

textBox28->Text = L"";

textBox29->Text = L"";

textBox30->Text = L"";

textBox31->Text = L"";

textBox32->Text = L"";

textBox33->Text = L"";

textBox34->Text = L"";

textBox35->Text = L"";

textBox36->Text = L"";

textBox37->Text = L"";

textBox38->Text = L"";

textBox39->Text = L"";

textBox40->Text = L"";

textBox41->Text = L"";

textBox42->Text = L"";

textBox43->Text = L"";

textBox44->Text = L"";

textBox45->Text = L"";

textBox46->Text = L"";

textBox47->Text = L"";

textBox48->Text = L"";

}

};

}