СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc185347948)

[1 Постановка задачи 5](#_Toc185347949)

[1.1 Анализ предметной области 5](#_Toc185347950)

[2 Техническое задание на разработку ГОСТ 19.201-78 7](#_Toc185347951)

[2.1 Основания для разработки 7](#_Toc185347952)

[2.2 Назначение для разработки 7](#_Toc185347953)

[2.3 Требования к программе или программному изделию 7](#_Toc185347954)

[2.4 Паспорта модулей и метрики 9](#_Toc185347955)

[3.Проектирование диаграмм 17](#_Toc185347956)

[3.1 Поведенчиские диаграммы 17](#_Toc185347957)

# **ВВЕДЕНИЕ**

Учебная практика направлена на освоение процесса разработки программного обеспечения, включая создание документации, проектирование, реализацию, тестирование и анализ метрик ПО. Задачей является выработка требований к ПО, разработка алгоритма и модели программы, а также создание модулей функций на основе спецификаций. В рамках работы необходимо обеспечить высокое качество проектирования диаграмм, соответствующего функциональным требованиям.

Документация должна включать описание алгоритма работы программы, модель программы, листинг программы с экранными формами, протоколы тестирования и скриншоты результатов. Важными этапами работы будут являться анализ проектной и технической документации, выявление ошибок в системных компонентах и использование встроенных инструментов для тестирования. Также потребуется провести анализ кода на возможность его оптимизации и снятие метрик ПО для оценки качества программы.

Программа будет тестироваться с целью выявления ошибок и проверки её соответствия спецификациям. Документация, созданная в ходе практики, должна отражать весь процесс разработки — от анализа требований до получения окончательных результатов тестирования с метриками.

# **1 Постановка задачи**

## 1.1 Анализ предметной области

Программа предназначена для работы с массивами данных, позволяя пользователю вводить размерность массива, автоматически заполнять его значениями с использованием генератора случайных чисел (ГСЧ), а затем классифицировать элементы массива как чётные и нечётные с использованием классов. Программа также должна выводить графическое представление данных, включая первоначальный массив, массив чётных чисел и массив нечётных чисел.

Программа будет работать с массивами целых чисел и обеспечивать следующие основные функции.

Массив чисел: После ввода размерности массива пользователю будет предложено автоматическое заполнение массива случайными значениями.

Чётные и нечётные числа: Программа должна определить, какие элементы массива являются чётными, а какие нечётными, используя классы для фильтрации данных.

Графическое отображение: Программа должна выводить графики:

* график первоначального массива;
* график чётных чисел;
* график нечётных чисел.

Интерфейс пользователя:

* пользователь вводит размерность массива;
* программа автоматически генерирует массив случайных чисел;
* отображаются графики для каждого из этапов (первоначальный массив, чётные и нечётные числа).

Формы для ввода и вывода данных:

* пользователь может задать размерность массива,
* после генерации данных, программа классифицирует элементы массива и выводит на экран графики.

Функциональность системы:

* ввод размерности массива с помощью интерфейса;
* использование генератора случайных чисел для создания массива;
* определение чётных и нечётных чисел с использованием классов;
* построение графиков для первоначального массива, массива чётных и нечётных чисел.

Таким образом, программа предоставляет пользователю простой и интуитивно понятный интерфейс для работы с массивами, их фильтрации и визуализации результатов в виде графиков, что способствует лучшему пониманию распределения чисел в массиве.

# **2 Техническое задание на разработку ГОСТ 19.201-78**

## 2.1 Основания для разработки

Основанием для разработки является задание по учебной практике по Модулю ПМ.02 Осуществление интеграции программных модулей подписанное председателем цикловой комиссии А.Н. Юдаевыем 10.12.2024 г. и руководителем учебно-производственной практики и службы трудоустройства А.В. Суворовым 12.12.2024 г. Руководителем практики является Е.А. Мачнева.

## 2.2 Назначение для разработки

Назначение для разработки программы заключается в создании инструмента для работы с массивом данных, который будет автоматически генерировать элементы массива на основе заданной размерности. Программа должна анализировать каждый элемент массива, определяя, какие числа являются четными, а какие - нечетными, используя классы для их классификации. Визуализация результатов будет выполнена в виде графиков, которые поочередно отображают исходный массив, массив четных чисел и массив нечетных чисел.

Программа предназначена для удобного анализа и обработки данных, улучшая понимание распределения чисел в массиве и обеспечивая наглядное представление результатов через графическое отображение. Она может быть полезна для пользователей, которые занимаются анализом числовых данных и визуализацией, а также для изучающих работу с массивами и их обработку.

## 2.3 Требования к программе или программному изделию

Требования к обеспечению надежного функционирования программы не предъявляются.

**2.3.1 Надежность и восстановление после сбоев**

Требования к обеспечению надежного функционирования программы не предъявляются. Время восстановления после нефатального сбоя операционной системы не должно превышать 10 минут при соблюдении условий эксплуатации технических и программных средств. Время восстановления после фатального сбоя зависит от времени, необходимого для устранения неисправностей технических средств и переустановки программного обеспечения.

**2.3.2 Отказы из-за некорректных действий оператора**

Отказы программы могут возникнуть из-за некорректных действий оператора при взаимодействии с операционной системой. Для минимизации таких рисков рекомендуется ограничить пользователю доступ к административным привилегиям.

**2.3.3 Условия эксплуатации**

Программа должна работать в климатических условиях, удовлетворяющих требованиям, предъявляемым к техническим средствам (ПК) в части эксплуатации.

**2.3.4 Требования к обслуживанию**

Программа не требует проведения каких-либо видов обслуживания.

**2.3.5 Требования к численности и квалификации персонала**

Минимальное количество персонала для эксплуатации программы должно составлять 2 человека: системный администратор и пользователь ПК – оператор. Системный администратор должен иметь высшее профильное образование. Его задачи включают поддержание работоспособности технических средств и системных программных средств, включая операционную систему.

**2.3.6 Требования к техническим средствам**

Программа должна работать на IBM-совместимом персональном компьютере с следующими характеристиками:

* процессор: Pentium 4 2.5 GHz (одноядерный) или выше;
* свободное место на жестком диске: от 1 Гб;
* оперативная память (ОЗУ): 4 Гб и выше;
* операционная система: Windows 10.

**2.3.7 Требования к информационной и программной совместимости**

Информационные структуры и методы решения: Требования к информационным структурам (записям) на входе и методам решения не предъявляются. Информационная структура должна включать отображение входных данных и результата на экране.

Исходные коды и языки программирования: Исходные коды программы должны быть реализованы на языке C#. В качестве интеграционной среды разработки программы должна быть использована среда Microsoft Visual Studio 2022 и приложения для построения диаграмм Draw.Io.

Защита информации и программ: Требования к защите информации и программ не предъявляются.

**2.3.8 Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем через графический интерфейс.

**2.3.9 Требования к программной докуменатции**

В состав сопровождающей документации должны входить:

* техническое задание на разработку (ГОСТ 19.201-78);
* руководство оператора (ГОСТ 19.505-79);
* программа, методика и результаты испытаний (ГОСТ 19.301-79).

## 2.4 Паспорта модулей и метрики

Разработка паспорта модуля и снятие метрик по заданию: Разработать программу для работы с массивом. Пользователь вводит M – размерность массива. ГСЧ заполняет его значениями. Определить, какие значения являются четными и нечетными (через классы). Поочередно вывести каждый график (первоначальный, четный, нечетный).

**2.4.1 Спецификации для паспорта модуля**

Метод GenerateArray генерирует массив случайных чисел с размерностью, введённой пользователем в текстовое поле. При успешной генерации появляется сообщение "Массив сгенерирован!".

Таблица 1 – Спецификация для функции GenerateArray

|  |  |  |
| --- | --- | --- |
| Имя | Функция | Параметры |
| GenerateArray | Генерация массива случайных чисел заданной размерности | Входные данные:  int size – размер массива для вывода графика. Выходные данные: Массив чисел, сгенерированный с помощью метода GenerateRandomArray. |

При нажатии на кнопку, метод ShowOriginalGraph вызывает метод DrawHistogram, передавая в него исходный массив, который был сгенерирован методом GenerateRandomArray. Гистограмма отображает все значения исходного массива.

Таблица 2 – Спецификация для функции ShowOriginalGraph

|  |  |  |
| --- | --- | --- |
| Имя | Функция | Параметры |
| ShowOriginalGraph | Отображение графика исходного массива | Входные данные:  Нет параметров (график строится на основе сгенерированного массива). Выходные данные: График, показывающий исходные значения массива. |

При нажатии на кнопку, метод ShowEvenGraph извлекает четные числа из массива с помощью метода GetEvenNumbers() и вызывает метод DrawHistogram для их графического отображения.

Таблица 3 – Спецификация для функции ShowEvenGraph

|  |  |  |
| --- | --- | --- |
| Имя | Функция | Параметры |
| ShowEvenGraph | Отображение графика четных чисел массива | Входные данные:  Нет параметров (график строится на основе сгенерированного массива). Выходные данные: График, показывающий только четные числа |

При нажатии на кнопку, метод ShowOddGraph извлекает нечетные числа из массива с помощью метода GetOddNumbers() и вызывает метод DrawHistogram для их графического отображения.

Таблица 4 - Спецификация для функции ShowOddNumbers

|  |  |  |
| --- | --- | --- |
| Имя | Функция | Параметры |
| ShowOriginalGraph | Отображение графика  нечетных чисел массива | Входные данные:  Нет параметров (график строится на основе сгенерированного массива). Выходные данные: График, , показывающий только нечетные числа |

**2.4.2 Блок-схемы для паспорта модуля**

В данной блок-схеме (рисунок 1) мы видим работу кнопки «Генерировать массив», при вводе целого числа мы попадаем в функцию GenerateRandomArray() в которой происходит генерация и заполнение массива, по итогу выводится сообщение «Массив сгенерирован», если пользователь вводит не целое число, он увидит сообщение «Пожалуйста, введите целое число»

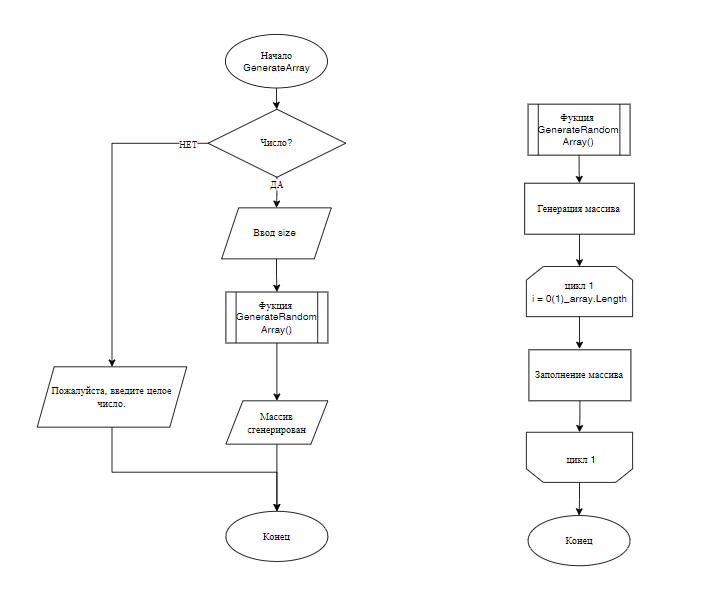


Рисунок 1 – Блок-схема для кнопки «Генерировать массив»

В блок-схеме (рисунок 2) мы видим работу кнопки «Показать исходный график», при нажатии на кнопку пользователю выводится исходный график. Кнопка работает при помощи классов GetArray() и DrawHistogram, функция GetArray() используется для генерации и заполнения массива, DrawHistogram нужна для того что бы отрисовывать графики.

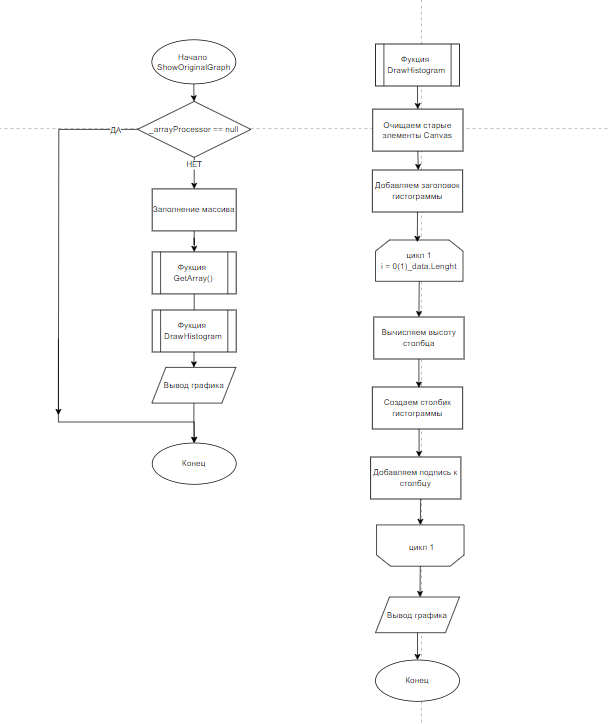


Рисунок 2 – Блок-схема для кнопки «Показать исходный график»

В блок-схеме (рисунок 3) мы видим работу кнопки «Показать четные числа», при нажатии на кнопку пользователю выводится график четных чисел. Кнопка работает при помощи классов GetArray() (рисунок 2) и DrawHistogram (рисунок 2).

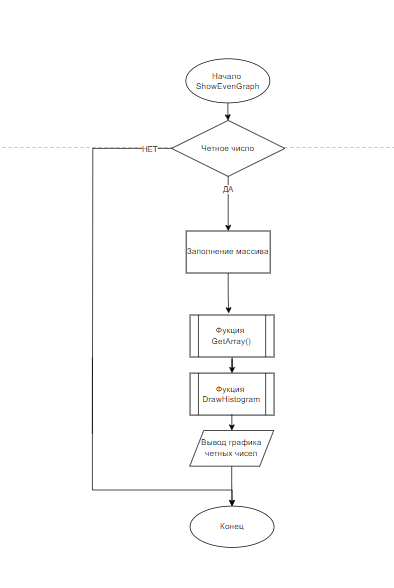


Рисунок 3 – Блок-схема для кнопки «Показать четные числа»

В блок-схеме (рисунок 4) мы видим работу кнопки «Показать нечетные числа», при нажатии на кнопку пользователю выводится график четных чисел. Кнопка работает при помощи классов GetArray() (рисунок 2) и DrawHistogram (рисунок 2).

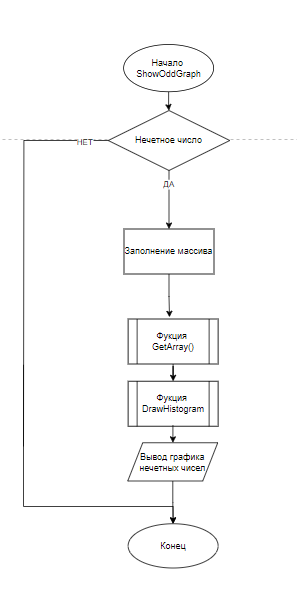


Рисунок 4 – Блок-схема для кнопки «Показать четные числа»

**2.4.3 Метрики**

1. Индекс удобства поддержки: 76

* этот индекс показывает, насколько легко поддерживать и сопровождать код:
* значение 76 находится в хорошем диапазоне. Значения выше 20 считаются приемлемыми;
* 76–100: отлично, код легко поддерживать;
* 50–75: хорошо, но есть место для улучшений;
* <50: код сложно поддерживать; рекомендуется рефакторинг.

Заключение: Мой код имеет хороший индекс поддерживаемости.

1. Сложность организации циклов: 28

* измеряет количество возможных путей выполнения программы (ветвлений, условий if, else, циклов и т.д.);

Значение 28 указывает на умеренную сложность.

* 1–10: простая логика;
* 11–20: умеренно сложная логика;
* >20: сложный код; рекомендуется декомпозировать большие методы.

Заключение: Сложность немного выше нормы. Возможно, стоит разделить код на более мелкие методы для упрощения.

1. Глубина наследования: 9

* показывает глубину иерархии наследования. Чем глубже иерархия, тем сложнее понять и поддерживать код.

Глубина 9 — достаточно высокая.

* 0–3: нормально, низкая сложность;
* 4–6: средняя сложность;
* >6: высокая сложность; код становится менее гибким.

Заключение: Высокая глубина наследования может привести к проблемам с поддержкой и расширяемостью кода.

1. Взаимозависимость классов: 9

* измеряет, насколько класс зависит от других классов и библиотек;
* 0–10: низкая связность (хорошо);
* 11–20: умеренная связность;
* >20: высокая связность; код сильно зависит от внешних классов.

Заключение: Значение 9 указывает на низкую связность, что является хорошим показателем.

1. Строки исходного кода: 36

* общее количество строк в исходном коде (без учета пустых строк и комментариев).

Заключение: Код небольшой по объему, что упрощает его чтение и поддержку.

1. Строки исполняемого кода: 45

* количество строк, которые действительно выполняются при запуске программы.

Заключение: Код содержит 45 исполняемых строк, что соответствует небольшому объему и линейности программы.

1. Общее заключение:

* индекс удобства поддержки: хороший (76);
* сложность кода: умеренная (28) — рекомендуется декомпозировать сложные методы;
* глубина наследования: высокая (9) — стоит упростить иерархию классов;
* связность классов: низкая (9) — хороший показатель;
* размер кода: компактный и небольшой.

# **3.Проектирование диаграмм**

## 3.1 Поведенчиские диаграммы

Основное внимание здесь уделяется динамическим аспектам системы программного обеспечения или процесса. Эти диаграммы показывают функциональные возможности системы и демонстрируют, что должно происходить в моделируемой системе. Говоря проще, поведенческие диаграммы описывают поведение объектов или субъектов. В этом типе нет строгого упора на сухую разработку ПО. При разработке поведенческой диаграммы всегда можно дать описание бизнеспроцессу.

На диаграмме (рисунок 5) мы видим диаграмму прецедентов,