УТВЕРЖДАЮ

М.К. Семёнов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“\_\_” \_\_\_\_\_\_\_2021г.

Система арифметических операций в массиве

наименование вида АС

СибГУ им. М.Ф. Решетнёва

наименование объекта автоматизации

САО массива

сокращенное наименование АС

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

На 10 листах

Действует с \_\_. \_\_ 2021г.

СОГЛАСОВАНО

М.К. Семёнов

“\_\_” \_\_\_\_\_\_\_ 2021г.

1. Общие сведения 3

1.1 Полное наименование системы и ее условное обозначение. 3

1.2 Наименование разработчика системы и реквизиты заказчика. 3

1.3. Основания для разработки ПО. 3

1.4. Плановые сроки начала и окончания работы по созданию системы: 3

1.5. Источник финансирования работ по созданию АС. 3

2. Назначение и цели создания системы 3

2.1 Назначение системы. 3

2.2 Цели создания системы. 4

3. Требования к системе 4

3.1. Требования к системе в целом. 4

3.1.1. Требования к структуре и функционированию системы 4

*3.1.2. Требования к средствам и способам связи для информационного обмена между компонентами системы.* 5

*3.1.3. Требования к характеристикам взаимосвязи создаваемой системы со смежными системами, требования к ее совместимости.* 5

*3.1.4. Требования по диагностированию системы.* 5

*3.1.5. Перспективы ПО, модернизация ПО.* 5

*3.1.6. Требования к надежности комплекса.* 5

*3.1.7 Требования к квалификации пользователя* 5

*3.1.8. Требования по стандартизации и унификации.* 6

3.2. Требования к задачам, выполняемым системой. 6

*3.2.1* *Перечень функций, подлежащих автоматизации:* 6

3.3. Требования к видам обеспечения. 6

*3.3.1. Требования к программному обеспечению.* 6

*3.3.2. Требования к техническому обеспечению.* 6

4. Тестирование программного обеспечения 7

4.1. Тестирование пределов программы: 7

4.2 Конфигурация рабочей машины 8

4.3. Тестирование работы программы при малом n: 8

4.4. Тестирование программы при максимальном n: 10

4.5 Результаты тестов и выводы 11

5. Техническое руководство 12

5.1. Создание массива размером n: 12

5.2. Выбор типа массива: 12

5.3. Генерация элементов массива с диапазоном [a; b]: 13

 Целочисленный массив: 13

 Дробный массив: 13

5.4. Выполнение арифметических операций над элементами массива: 13

 Целочисленный массив: 13

 Дробный массив: 14

# 1. Общие сведения

## 1.1 Полное наименование системы и ее условное обозначение.

Система арифметических операций в массиве

*Условное обозначение:* CAO массива

## 1.2 Наименование разработчика системы и реквизиты заказчика.

Заказчик – Семёнов Михаил Константинович

Разработчик – студент группы БПИ21-01 Иванов Владислав Григорьевич

## 1.3. Основания для разработки ПО.

Работа по созданию системы арифметических операций в массиве

## 1.4. Плановые сроки начала и окончания работы по созданию системы:

- начало работ по созданию системы – 1 декабря 2021

- окончание работ по созданию системы – .. декабря 2021

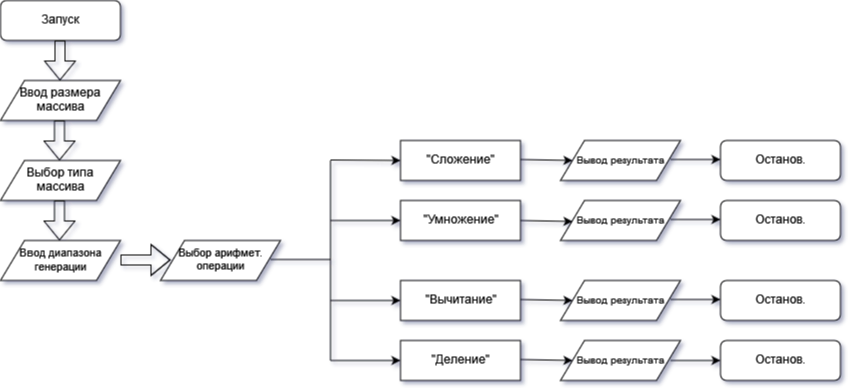
## 1.5. Источник финансирования работ по созданию АС.

Отсутствует.

# 2. Назначение и цели создания системы

## 2.1 Назначение системы.

САО массива предназначена для генерации массивов, размером заданных переменной N, с диапазоном генерации [a; b] и для проведения таких арифметических операций, как сложение, вычитание, умножение, деление. Программа должна выводить время выполнения операций.



UML диаграмма для пользователя

## 2.2 Цели создания системы.

Целью создания системы является:

Задание, данное преподавателем дисциплины “Основы программной инженерии”.

# 3. Требования к системе

## 3.1. Требования к системе в целом.

### 3.1.1. Требования к структуре и функционированию системы

САО массива должна включать в себя следующий функционал:

* создание массива с размером n
* выбор типа массива (целочисленный, дробный)
* генерация элементов массива с диапазоном [a;b]
* выполнение арифметических операций над элементами массива

1. Создание массива с размером n:
   * Переменная n задаётся пользователем с клавиатуры;
2. Выбор типа массива (целочисленный, дробный)
   * Пользователю предлагается выбрать тип элементов массива;
   * В соответствии с выбранным типом, создаётся определённый массив на n элементов;
3. Генерация элементов массива с диапазоном [a; b]:
   * Переменные a и b задаются пользователем с клавиатуры;
4. Выполнение арифметических операций над элементами массива:
   * Пользователь может выбрать 4 типа операций;
   * Арифметические операции должны быть выполнены над всеми элементами массива в соответствии с выбранной операцией
   * Программа подсчитывает время выполнения операции

### *3.1.2. Требования к средствам и способам связи для информационного обмена между компонентами системы.*

Требования к средствам и способам связи отсутствуют.

### *3.1.3. Требования к характеристикам взаимосвязи создаваемой системы со смежными системами, требования к ее совместимости.*

Требования к характеристикам взаимосвязи создаваемой системы отсутствуют.

### *3.1.4. Требования по диагностированию системы.*

Проверка программного и аппаратного обеспечения проводится по мере необходимости.

### *3.1.5. Перспективы ПО, модернизация ПО.*

САО массива может быть выполнена в другой среде разработки с целью создания понятного графического интерфейса для пользователя.

### *3.1.6. Требования к надежности комплекса.*

Необходимо, чтобы система обладала устойчивостью к отказам оборудования и программных систем, а также электропитания. Для надежной работы комплекса необходимы высоконадежные аппаратные и программные системы. Требования надежности должны быть регламентированы для следующих аварийных ситуаций:

* выход из строя аппаратных средств системы;
* отсутствие электроэнергии;
* выход из строя программных средств системы;
* неверные действия персонала компании;
* пожар, взрыв и т.п.

Методы оценки и контроля показателей надежности на разных стадиях создания системы должны отвечать следующим особенностям:

* + многофункциональность;
  + разнообразные законы распределения среднего времени безотказной работы и восстановления.

### *3.1.7 Требования к квалификации пользователя*

Квалификация пользователя программы:

Пользователь программы должен владеть навыками работы с операционной системой Microsoft Windows 7/8/8.1/10/11.

### *3.1.8. Требования по стандартизации и унификации.*

Массив и результаты вычислений содержатся исключительно в оперативной памяти, то есть не сохраняются.

## 3.2. Требования к задачам, выполняемым системой.

### *Перечень функций, подлежащих автоматизации:*

1. *Создание массива размером n:*

Массив создаётся путём выделения динамической памяти компьютера. Данный способ позволяет пользователю самому задать необходимый ему размера массива.

1. *Выбор типа массива:*

Пользователь может выбрать тип создаваемого массива и его элементов. Для каждого типа создаётся свой уникальный массив со своими уникальными функциями.

1. *Генерация элементов массива с диапазоном [a; b]:*

Пользователь сам выбирает границы диапазона генерации элементов массива путём ввода переменных a и b. Стоит отметить, что числа a и b включены в диапазон генерации.

1. *Выполнение арифметических операций над элементами массива:*

Пользователю предлагается выбор из четырёх арифметических операций. Каждая операция выполняется при помощи циклов, которые затрагивают каждый элемент массива.

## 3.3. Требования к видам обеспечения.

### *3.3.1. Требования к программному обеспечению.*

Для работы САО массива необходимо следующее ПО:

1. На рабочей машине должны быть установлены:

* Операционная система: Microsoft Windows 7/8/8.1/10/11.

### *3.3.2. Требования к техническому обеспечению.*

* процессор Intel Core i3-i5 или AMD Ryzen 3;
* 4GB и более оперативной памяти (для создания больших массивов необходимо 8GB и более);
* 80 GB – жесткий диск;
* Видеокарта с 2GB видеопамяти;
* Клавиатура;
* Манипулятор типа «мышь».

# 4. Тестирование программного обеспечения

## 4.1. Тестирование пределов программы:

Программа предназначена для создания массива размером n, где n – число, которое задаёт пользователь. Путём тестов было выяснено, что максимальное значение n для 8GB – равно 89999999 (значение может превышать максимум в зависимости от объёма оперативной памяти). То есть в программе можно создать массив максимум на 89999999 элементов.

Значения a и b, отвечающие за диапазон массива имеют свои пределы равные пределу типа long long (для целочисленных массивов) и типа long double (для дробных массивов).

4.2 Конфигурация рабочей машины

Все тесты проходили на персональном компьютере с данной конфигурацией:

|  |  |
| --- | --- |
| Процессор | Intel® Core™ i5-10210U CPU @ 1.60GHz |
| Оперативная память | 8,00 ГБ |
| Тип системы | 64-разрядная операционная система, процессор x64 |
| Видеоадаптер | Inter® UHD Graphics |

## 4.3. Тестирование работы программы при малом n:

Результаты работы программы:

1. **Сложение:**

|  |  |
| --- | --- |
| Количество элементов массива: 50 | Количество элементов массива: 50 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: 271 | Результат вычислений: 290.405 |
| Минимальное значение: 1 | Минимальное значение: 1.1231 |
| Максимальное значение: 10 | Максимальное значение: 9.54201 |
| Время выполнения операции: 6мс. | Время выполнения операции: 8мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный**. |

1. **Умножение:**

|  |  |
| --- | --- |
| Количество элементов массива: 50 | Количество элементов массива: 50 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: 2.84605e+3 | Результат вычислений: 1.43883e+34 |
| Минимальное значение: 1 | Минимальное значение: 1.223 |
| Максимальное значение: 10 | Максимальное значение: 9.78774 |
| Время выполнения операции: 6мс. | Время выполнения операции: 6мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный.** |

1. **Вычитание**

|  |  |
| --- | --- |
| Количество элементов массива: 50 | Количество элементов массива: 50 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: -283 | Результат вычислений: -272.985 |
| Минимальное значение: 1 | Минимальное значение: 1.11638 |
| Максимальное значение: 10 | Максимальное значение: 9.85864 |
| Время выполнения операции: 4мс. | Время выполнения операции: 7мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный.** |

1. **Деление**

|  |  |
| --- | --- |
| Количество элементов массива: 50 | Количество элементов массива: 50 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: 4.3803e30 | Результат вычислений: 4.38286e-35 |
| Минимальное значение: 1 | Минимальное значение: 1.18755 |
| Максимальное значение: 10 | Максимальное значение: 9.8428 |
| Время выполнения операции: 5мс. | Время выполнения операции: 8мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный.** |

4.4. Тестирование программы при максимальном n:

1. **Сложение:**

|  |  |
| --- | --- |
| Количество элементов массива: 89999999 | Количество элементов массива: 89999999 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: 4.94967e+08 | Результат вычислений: 4.95002e+08 |
| Минимальное значение: 1 | Минимальное значение: 1. |
| Максимальное значение: 10 | Максимальное значение: 9.9 |
| Время выполнения операции: 1415мс. | Время выполнения операции: 948мс. |
| Время создания массива: 11288мс. | Время создания массива: 9079мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный** |

1. **Умножение:**

|  |  |
| --- | --- |
| Количество элементов массива: 89999999 | Количество элементов массива: 89999999 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: inf (ошибка) | Результат вычислений: inf (ошибка) |
| Минимальное значение: 1 | Минимальное значение: 1.1 |
| Максимальное значение: 10 | Максимальное значение: 9.9 |
| Время выполнения операции: 1515мс. | Время выполнения операции: 800мс. |
| Время создания массива: 10477мс. | Время создания массива: 9760мс. |
| Тип данных: **Целочисленный.** | Тип данных: **Дробный** |

1. **Вычитание:**

|  |  |
| --- | --- |
| Количество элементов массива: 89999999 | Количество элементов массива: 89999999 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: -4.94969e+08 | Результат вычислений: -4.94952e+08 |
| Минимальное значение: 1 | Минимальное значение: 1.1 |
| Максимальное значение: 10 | Максимальное значение: 9.9 |
| Время выполнения операции: 1532мс. | Время выполнения операции: 800мс. |
| Время создания массива: 10512мс. | Время создания массива: 7896мс. |
| Тип данных: **Целочисленный** | Тип данных: **Дробный.** |

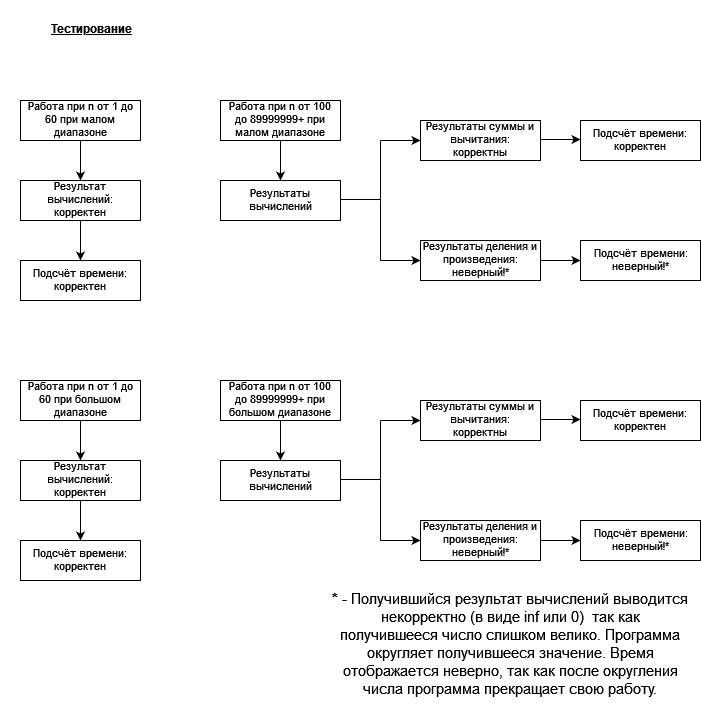
1. **Деление:**

|  |  |
| --- | --- |
| Количество элементов массива: 89999999 | Количество элементов массива: 89999999 |
| Диапазон генерации от 1 до 10 | Диапазон генерации от 1.1 до 9.9 |
| Результат вычислений: 0 (ошибка) | Результат вычислений: 0 (ошибка) |
| Минимальное значение: 1 | Минимальное значение: 1.1 |
| Максимальное значение: 10 | Максимальное значение: 9. |
| Время выполнения операции: 962мс | Время выполнения операции: 893мс. |
| Время создания массива: 8144мс. | Время создания массива: 8202мс. |
| Тип данных: **Целочисленный** | Тип данных: **Дробный.** |

4.5 Результаты тестов и выводы

В ходе тестов было выяснено, что при значениях переменой n превышающих значение равное 100 и при большом диапазоне генерации [a; b] программа может выдавать неверные результаты при умножении или делении. Так же время выполнения операций тоже не верно. При сложении и вычитании предел n увеличивается до 100000000 при условии, что диапазон генерации небольшой.

Время создания массивов и выполнения арифметических операций напрямую зависит от объёма доступной оперативной памяти.



UML диаграмма тестирования ПО

**5. Техническое руководство**

5.1. Создание массива размером n:

За размер массива отвечает переменная n, имеющая тип unsigned long long. Данный тип данных, позволяет пользователю задавать очень большие размеры для массива.

Переменная n задаётся пользователем с клавиатуры.

5.2. Выбор типа массива:

В зависимости от типа данных создаётся определённый массив. Пользователю предлагается выбор из целочисленного типа данных и дробного типа данных.

Оба массива задаются динамически:

unsigned long long\* array = new unsigned long long[n]; (для целочисленных)

long double\* array2 = new long double[n]; (для дробных)

Далее при помощи оператора switch(), пользователю предлагается выбор. Если пользователь ввёл ‘1’, то выполняется работа с целочисленным массивом, если ‘2’, то идёт работа с дробным массивом.

5.3. Генерация элементов массива с диапазоном [a; b]:

* Целочисленный массив:

За границы диапазона генерации отвечают переменные a и b. Значения для данных переменных вводится пользователем с клавиатуры.

Массив заполняется случайным числами при помощи данного цикла:

for (long long j = 0; j < n; j++)

{

array[j] = a + rand() % ((b + 1) - a);

}

Для работы случайной генерации необходимы следующие директивы:

#include <time.h>

srand(time(0));

* Дробный массив:

За границы диапазона генерации отвечают переменные с и d. Значения для данных переменных вводится пользователем с клавиатуры.

Массив заполняется случайным числами при помощи данного цикла:

for (long j = 0; j < n; j++)

{

array2[j] = c + static\_cast <float> (rand()) / (static\_cast <float> (RAND\_MAX / (d - c)));

}

Для работы случайной генерации необходимы следующие директивы:

#include <time.h>

srand(time(0));

5.4. Выполнение арифметических операций над элементами массива:

* Целочисленный массив:

Пользователю предлагается выбор из 4 арифметических операций (сложение, умножение, вычитание, деление). Выбор реализован при помощи оператора switch().

* + Если пользователь выбрал ‘1’:
    - Создаётся нулевая переменная slozh
    - При помощи цикла подсчитывается сумма всех элементов:

for (long long z = 0; z < n; z++)

{

slozh = slozh + array[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << slozh;

* + Если пользователь выбрал ‘2’:
    - Создаётся переменная со значение первого элемента массива proiz
    - При помощи цикла подсчитывается произведение всех элементов:

for (long long z = 1; z < n; z++)

{

proiz \*= array[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << proiz;

* + Если пользователь выбрал ‘3’:
    - Создаётся переменная со значение первого элемента массива vychit
    - При помощи цикла подсчитывается разность всех элементов:

for (long long z = 1; z < n; z++)

{

vychit -= array[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << vychit;

* + Если пользователь выбрал ‘4’:
    - Создаётся переменная со значение первого элемента массива del
    - При помощи цикла все элементы делятся друг на друга:

for (long long z = 0; z < n; z++)

{

del /= array[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << del;

* Дробный массив:
  + Если пользователь выбрал ‘1’:
    - Создаётся нулевая переменная slozh2
    - При помощи цикла подсчитывается сумма всех элементов:

for (long long z = 0; z < n; z++)

{

slozh2 = slozh2 + array2[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << slozh2;

* + Если пользователь выбрал ‘2’:
    - Создаётся переменная со значение первого элемента массива proiz2
    - При помощи цикла подсчитывается произведение всех элементов:

for (long long z = 1; z < n; z++)

{

proiz2 \*= array2[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << proiz2;

* + Если пользователь выбрал ‘3’:
    - Создаётся переменная со значение первого элемента массива vychit2
    - При помощи цикла подсчитывается разность всех элементов:

for (long long z = 1; z < n; z++)

{

vychit2 -= array2[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << vychit2;

* + Если пользователь выбрал ‘4’:
    - Создаётся переменная со значение первого элемента массива del2
    - При помощи цикла все элементы делятся друг на друга:

for (long long z = 0; z < n; z++)

{

del2 /= array2[z];

}

* + - Результат выводится на экран:

cout << "\n\n\nРезультат вычислений: " << del2;

5.5 UML диаграммы для разработчика