УТВЕРЖДАЮ

М.К. Семёнов

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

“\_\_” \_\_\_\_\_\_\_2022г.

Аппарат различного научного исследования массива

наименование вида АС

СибГУ им. М.Ф. Решетнёва

наименование объекта автоматизации

АРНИ массива

сокращенное наименование АС

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

На 16 листах

Действует с \_\_. \_\_ 2022г.

СОГЛАСОВАНО

М.К. Семёнов

“\_\_” \_\_\_\_\_\_\_ 2022г.

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 предельных значений программы: 7

4.2. Конфигурация рабочей машины 8

4.3. Исследование работы программы при различных параметрах (size, min, max, prec): 8

4.4. Результаты исследования и выводы 10

**5. Техническое руководство** 10

5.1. Измерение времени выполнения операций 10

5.2. Реализация взаимодействий пользователя и программы (UI): 11

5.3. Создание массива заданного размера, заданного типа и заданного диапазона: 11

5.4. Выполнение арифметических операций над элементами массива 12

5.5 UML диаграммы для разработчика 15

# 1. Общие сведения

## 1.1 Полное наименование системы и ее условное обозначение.

Аппарат различного научного исследования массива

*Условное обозначение:* АРНИ массива

## 1.2 Наименование разработчика системы и реквизиты заказчика.

Заказчик – Семёнов Михаил Константинович

Разработчик – студент группы БПИ23-01 Мордован Артём Андреевич

## 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 – 12 декабря 2023

- окончание работ по созданию системы – 12 декабря 2023

## 1.5. Источник финансирования работ по созданию АС.

Отсутствует.

# 2. Назначение и цели создания системы

## 2.1 Назначение системы.

АРНИ массива предназначена для генерации массивов, размером size, с диапазоном генерации [min; max) и для проведения таких арифметических операций, как сложение, вычитание, умножение, деление. Программа должна выводить время выполнения операций.

да

нет

запуск

-Ввод размера массива;

- Ввод диапазона;

- Выбор типа (int/float/double)

Нажатие кнопки “Сгенерировать”

Выполнение необходимых операций с массивом, посредством нажатия кнопков со знаком действия

Просмотр результатов

требуется создать новый массив?

остановка

UML диаграмма для пользователя

## 2.2 Цели создания системы.

Целью создания системы является:

Выполнение задания, данного заказчиком.

# 3. Требования к системе

## 3.1. Требования к системе в целом.

### 3.1.1. *Требования к структуре и функционированию системы.*

АРНИ массива должна включать в себя следующий функционал:

* создание массива с размером size
* выбор типа массива (целочисленный, дробный, дробный с фиксированным кол-вом знаков)
* генерация элементов массива с диапазоном [min;max]
* выполнение арифметических операций над элементами массива

1. Создание массива с размером size:

* Выбор типа массива (целочисленный, дробный, дробный с фиксированным кол-вом знаков)

1. Генерация элементов массива с диапазоном [min; max)
2. Выполнение арифметических операций над элементами массива

### *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.*

АРНИ массива может быть модернизирована следующим образом:

1. улучшение интерфейса
2. оптимизация создания decimal чисел для увеличения скорости
3. реализация операций с decimal числами с сохранением количества знаков после запятой.

### *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 10/11.

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

Массив содержится исключительно в оперативной памяти, то есть не сохраняется.

Результаты вычислений сохраняются и записываются в файл “reports.txt”

## 3.2. Требования к задачам, выполняемым системой.

### *Перечень функций, подлежащих автоматизации:*

1. *Создание массива размером size:*

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.

1. *Генерация элементов массива с диапазоном [min; max):*

Пользователь сам выбирает границы диапазона генерации элементов массива путём ввода значений min и max. Число min включено в диапазон, число max не включено 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 БАРНИ 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 и лучше;
* 4GB и более оперативной памяти;
* 100 MB – жесткий диск;
* Клавиатура;
* Мышь.

# 4. Тестирование программного обеспечения

## 4.1. Тестирование предельных значений программы:

Путём тестов было выяснено, что максимальное значение size для 16GB приблизительно равно 500000000 (значение может превышать данное в зависимости от объёма доступной оперативной памяти в данный момент). То есть в программе можно создать массив максимум на 500000000 элементов.

Значения, отвечающие за диапазон массива, также имеют свои ограничения равные минимальным и максимальным значениям типа np.int64 (для целочисленных массивов) и типа np.float32 (для дробных массивов).

4.2. Конфигурация рабочей машины

Все тесты проходили на ноутбуке с данной конфигурацией:

|  |  |
| --- | --- |
| Процессор | Intel(R) Core (TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz |
| Оперативная память | 16,00 ГБ |
| Тип системы | 64-разрядная Windows 10,  процессор x64 |
| Видеоадаптер | NVIDIA GeForce MX330 |

## 4.3. Исследование работы программы при различных параметрах (size, min, max, prec):

**Малое количество элементов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Int** | **Float** | **Double** |
| **Информация о массиве** | [min, max]: [1,10)  size: 100 | [min, max): [1,10)  size: 100 | [min, max): [1,10)  size: 100  значащих цифр в числе: 200 |
| **Создание массива** | t: 8.29089e-05 | t: 8.750008e-05 | t: 0.0028621 |
| **Сложение** | res: 579  t: 2.500019e-05 | res: 572.4887644627674  t: 3.820005804300308e-05 | res: 529.86937....25358  t: 7.409998215734959e-05 |
| **Вычитание** | res: -579  t: 2.4999957e-05 | res: -572.4887644627674  t: 3.25001310557e-05 | res: - 529.86937….25358  t: 7.93999060988e-05 |
| **Умножение** | res: 0 (overflow)  t: 7.62001e-05 | res:1.5418752700357597e+70  t: 5.3300056e-05 | res: 21680....110592388.97071.. 8538  t: 0.0006372998468577862 |
| **Деление** | res: 2.706650454833e-68  t: 7.9100029e-05 | res: 5.971146158536777e-69  t: 5.5499840e-05 | res: 3.469152….51535E-65  t: 0.0008315001614 |
| ***заметки*** | *Ошибочный результат в умножении* | *Числа генерируются с небольшим кол-вом знаков после запятой* | *Заметно куда более долгое создание массива типа decimal (из-за особенностей реализации генерации decimal чисел).*  *Особенность типа decimal: отсутствие фиксированного количества знаков после запятой* |

**Дробный диапазон для проверки возможностей типов double и float**

|  |  |  |
| --- | --- | --- |
|  | **Float** | **Double** |
| **Информация о массиве** | [min, max): [0.5, 1.5)  size: 1000000 | [min, max): [0.5, 1.5)  size: 1000000  значащих цифр в числе: 100 |
| **Создание массива** | t: 0.008410600014030933 | t: 18.382410999853164 |
| **Сложение** | res: 1000082.5237832919  t: 0.06445659999735653 | res: 1000007.054187989….12811  t: 0.13882950018160045 |
| **Вычитание** | res: -1000082.5237832919  t: 0.0649999999441206 | res: -1000007.0541….3912811  t: 0.13688670005649328 |
| **Умножение** | res: 5e-324  t: 0.2466593000572174 | res: 8.6037178592183….04772018E-19634  t: 0.6440141999628395 |
| **Деление** | res: inf  t: 0.1185379000380635 | res: 9.028180173413503….850654E+19632  t: 0.953028899943456 |

**Большое количество элементов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Int** | **Float** | **Double** |
| **Информация о массиве** | [min, max]: [1,10)  size: 500000000 | [min, max): [1,10)  size: 500000000 | [min, max): [1,10)  size: 5000000  prec: 100 |
| **Создание массива** | t: 7.53287580003961 | t: 3.91683009988628 | t: 86.76358040003 |
| **Сложение** | res: 2749934179  t: 26.4599156999029 | res: 2749971565.8061013  t: 35.5053929998539 | res: 27492919.52….24  t: 0.6463817998301238 |
| **Вычитание** | res: -2749934179  t: 28.03673449996859 | res: -2749971565.8061013  t: 35.2950500000733 | res: -27492919.52….24  t: 0.6143795000389218 |
| **Умножение** | res: 0 (overflow)  t: 41.1260536999907 | res: inf  t: 46.9542788001708 | res: overflow  t: -- |
| **Деление** | res: 0.0  t: 59.1001502098 | res: 0.0  t: 55.0063666 | res: 0E+999999  t: 1.809903000016 |
| ***заметки*** | *Ошибочные результаты при умножении, при делении* | *Ошибочные результаты при умножении, при делении* | *- Заметно куда более долгое создание массива типа decimal (из-за особенностей реализации генерации decimal чисел).*  *- Ошибка в умножении, отсутствие результата, отсутствие времени выполнения* |

4.4. Результаты исследования и выводы

В ходе исследования было выяснено:

1. тип float в python учитывает лишь небольшое количество знаков после запятой, откуда можно сделать вывод о его неспособности точно работать с дробными величинами и корректно их выдавать, следовательно числа типа float недопустимо использовать в вычислениях, где необходима скрупулёзная точность
2. **Поведение программы при равенстве искомого числа значению превышающему максимально возможное число данного типа**  
   2.1) при перемножении массива int даже с относительно малым диапазоном (1-10) при подсчете получается выход за границу допустимого значения int, из-за чего программа выдает ошибку (overflow) и результат 0, время вычисления при этом считается корректно

2.2) при выходе за максимально возможное число типа float программа возвращает inf, при выходе за минимально возможное положительное возвращает 0.0

1. в типе double функция getcontext().prec отвечает за количество значащих символов в числе, соответственно при сложении этих чисел точность снижается ввиду постепенного уменьшения количества знаков после запятой (при увеличении целой части); непосредственно возможность задавать количество знаков после запятой в этой программе не реализована
2. библиотека numpy существенно увеличивает скорость работы с массивами типов int и float
3. тип double позволяет проводить самые точные расчеты из всех типов реализованных здесь
4. Время создания массивов и выполнения арифметических операций напрямую зависит от объёма доступной оперативной памяти.

**5. Техническое руководство**

## 5.1. Измерение времени выполнения операций

Измерение времени реализовано через библиотеку timeit и функции default\_timer():

st = default\_timer()

<тело операции>

fn = default\_timer()-st

## 5.2. Реализация взаимодействий пользователя и программы (UI):

Графический интерфейс был реализован при помощи библиотеки PyQt5 языка Python. При помощи различных виджетов (а именно: QLabel, QPushButton, QTestEdit) был осуществлен интерактивный способ взаимодействия **пользователя с программой**.

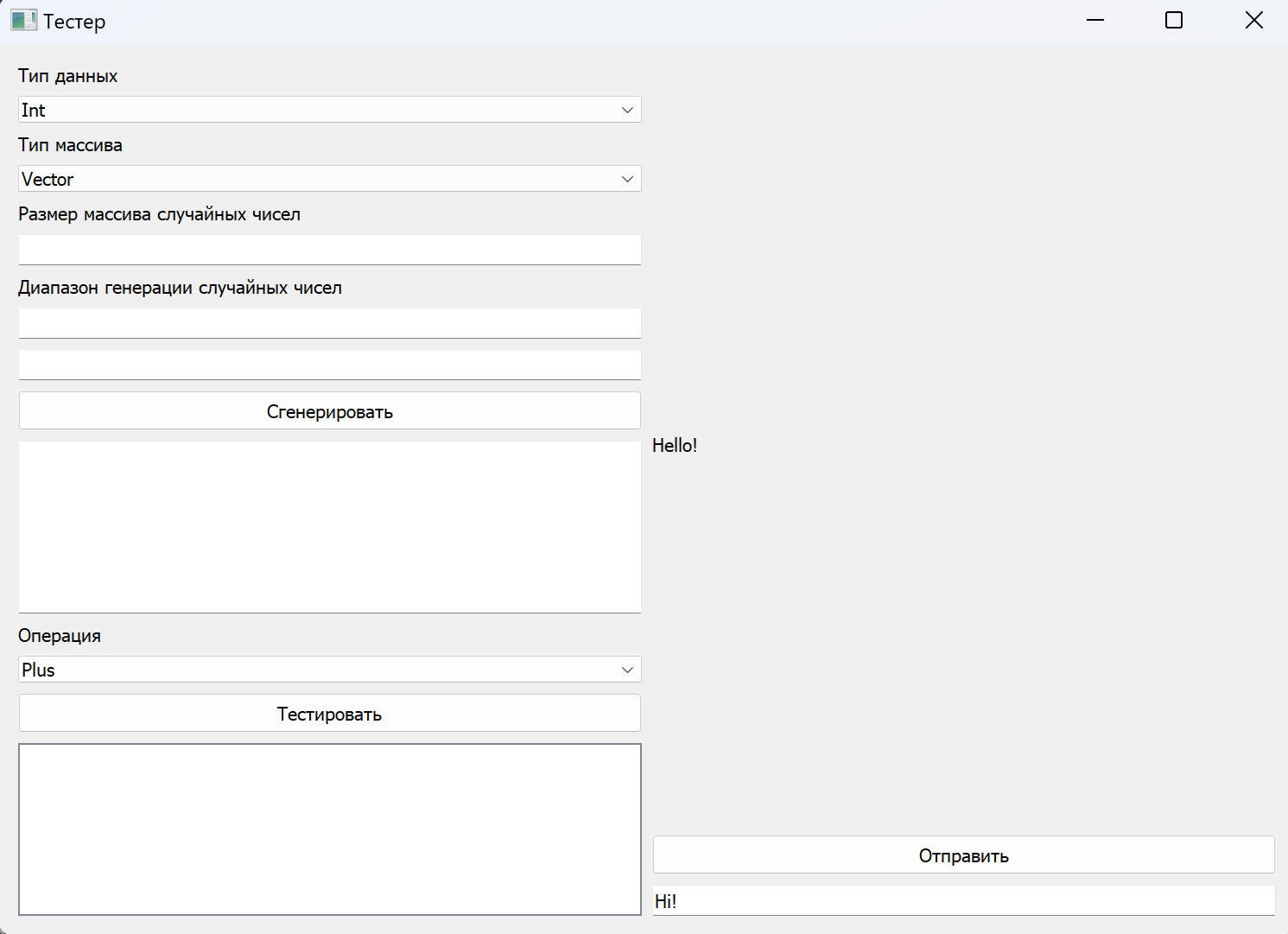
Нужное расположение виджетов было оформлено при помощи метода setLayout.

Взаимодействие **программы с пользователем** (т.е. фиксирование и передача данных о вычислениях пользователю) осуществлено с помощью:

1) текстового поля внизу экрана, отображающего информацию о последней операции.

2) вывода каждого действия в консоль

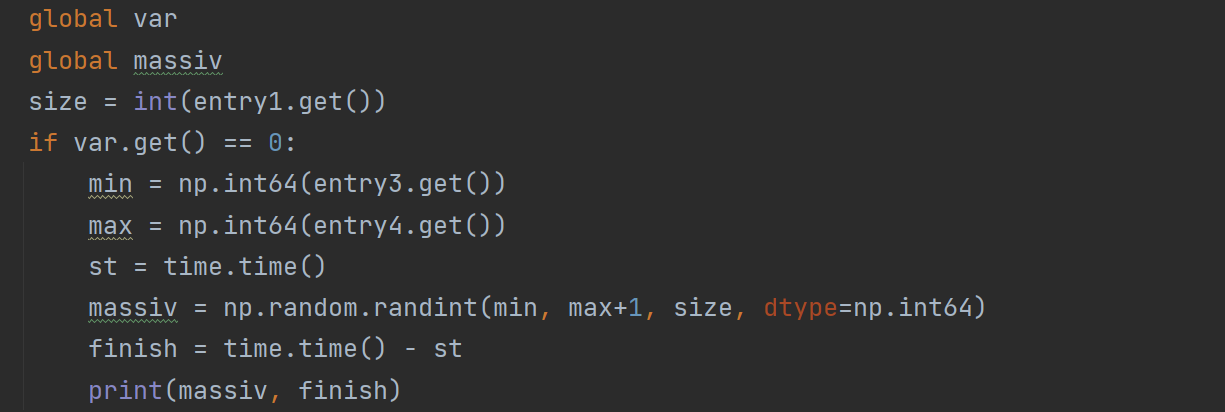
3) фиксирования каждого действия в файл отчетов “reports.txt”



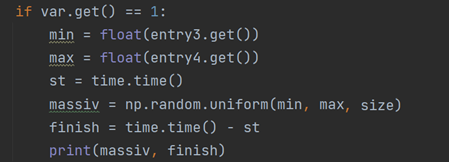
Итоговый вид окна функционирующей программы

## 5.3. Создание массива заданного размера, заданного типа и заданного диапазона:

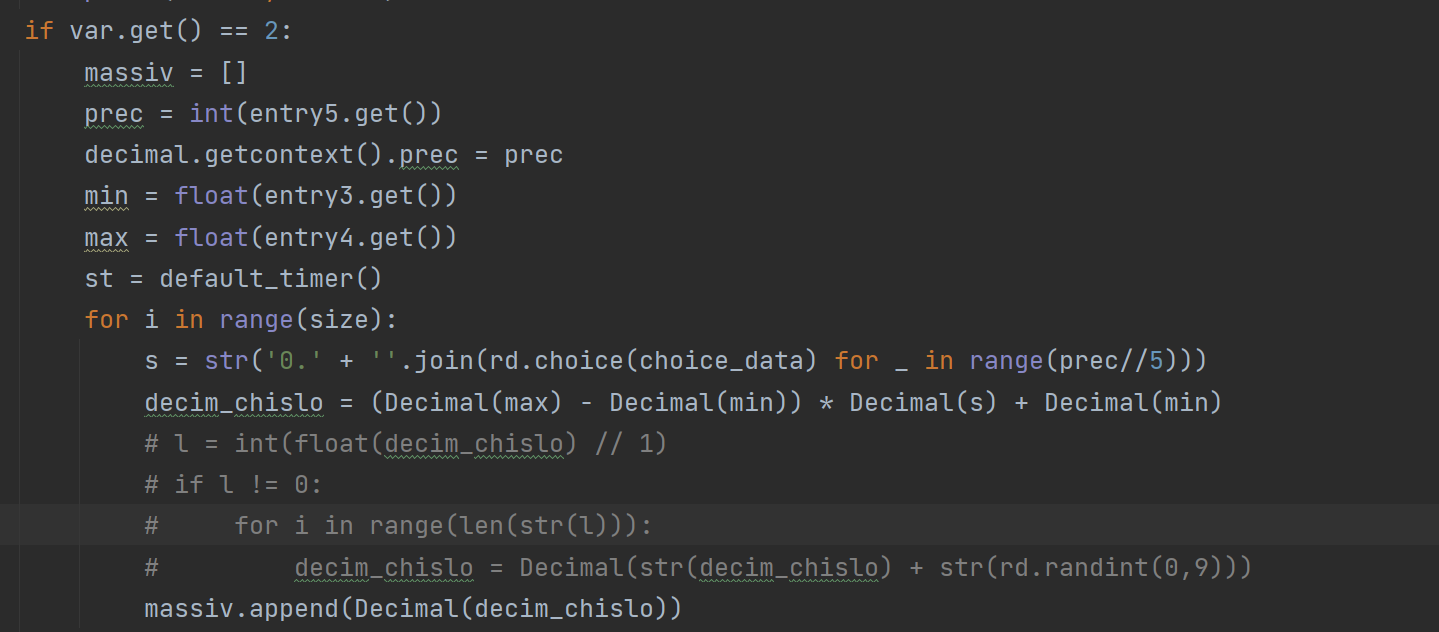
Массив создается нажатием кнопки “Сгенерировать”. Выполняется функция generation: считываются данные, введенные пользователем, и на их основе создается массив случайных значений используя np.random.randint и np.random.uniform (для целочисленных и дробных значений соответственно); массив decimal чисел формируется при помощи добавления в massiv decimal чисел, формируемых по алгоритму представленному ниже



Генерация массива целочисленных значений



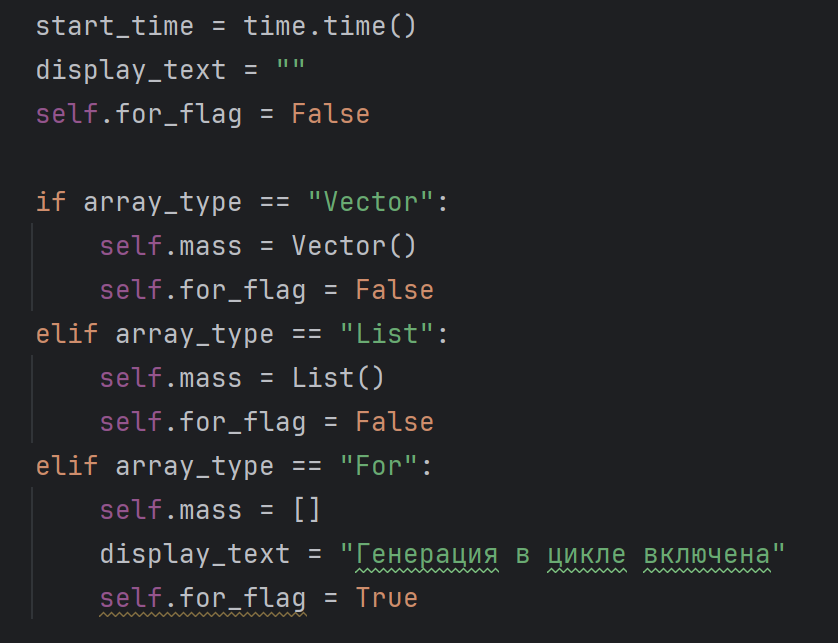
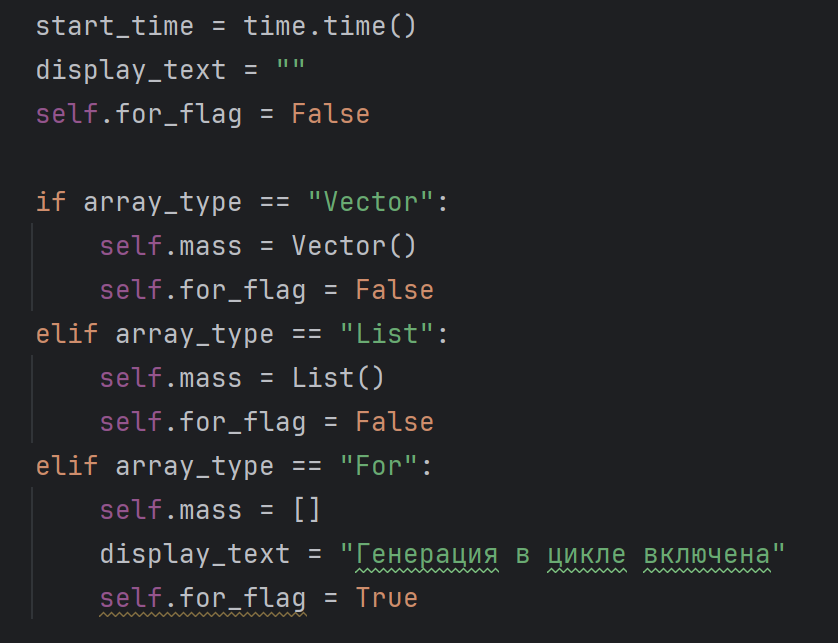
Генерация массива дробных значений

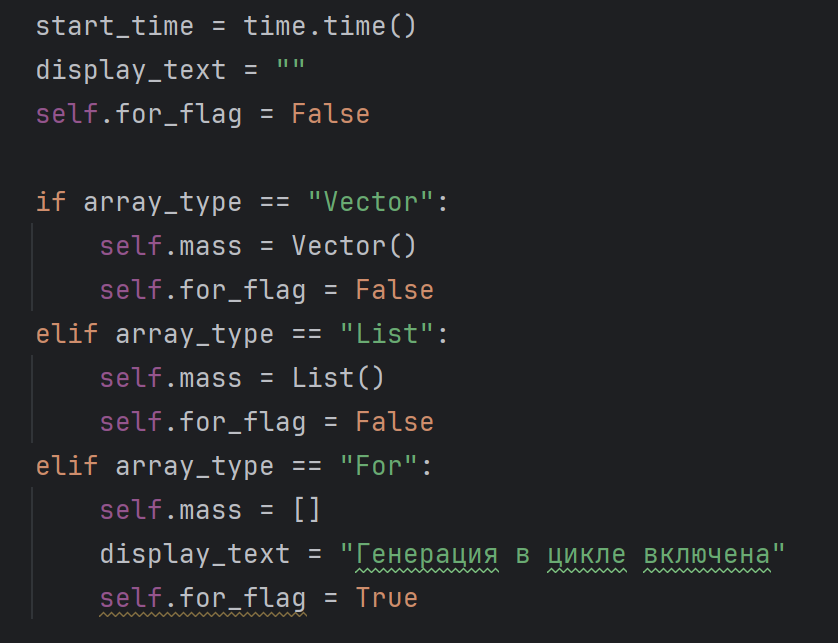


Генерация массива элементов типа decimal

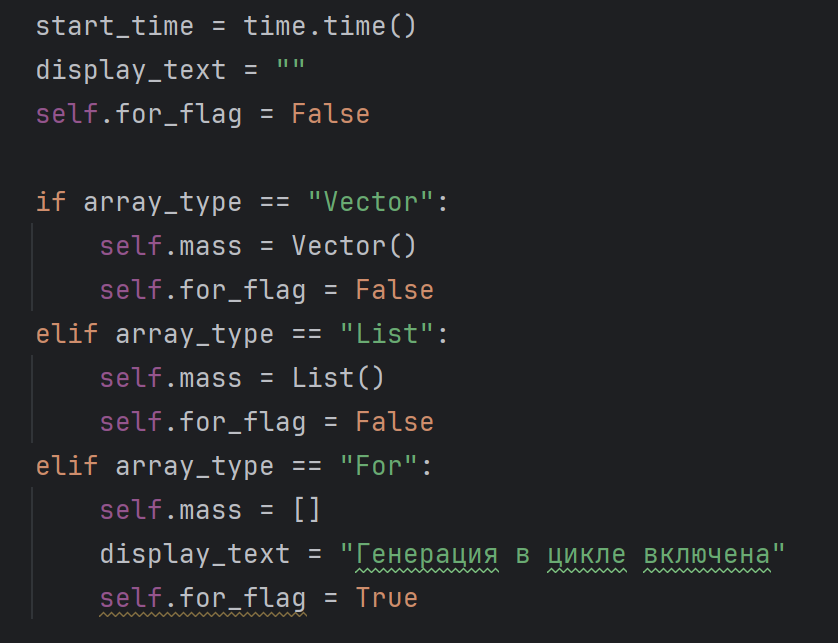
## 5.4. Выполнение арифметических операций над элементами массива

Реализация функций для выполнения арифметических операций над элементами массива не зависит от того, какой массив требуется обработать (int или float)

* + **Сложение:**
  + ****
  + **Вычитание:**
  + ****
  + **Умножение**:

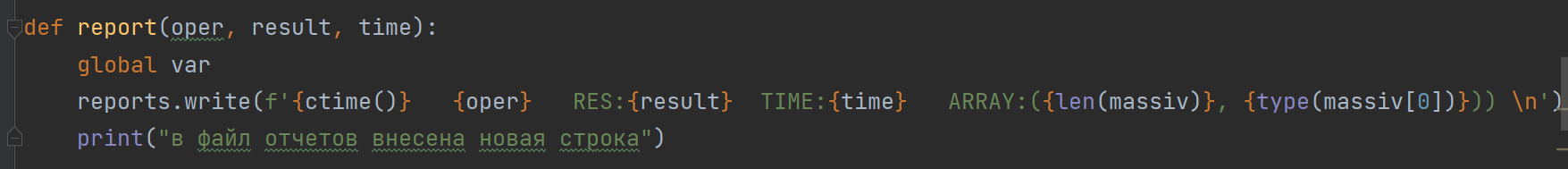


* + **Деление**:

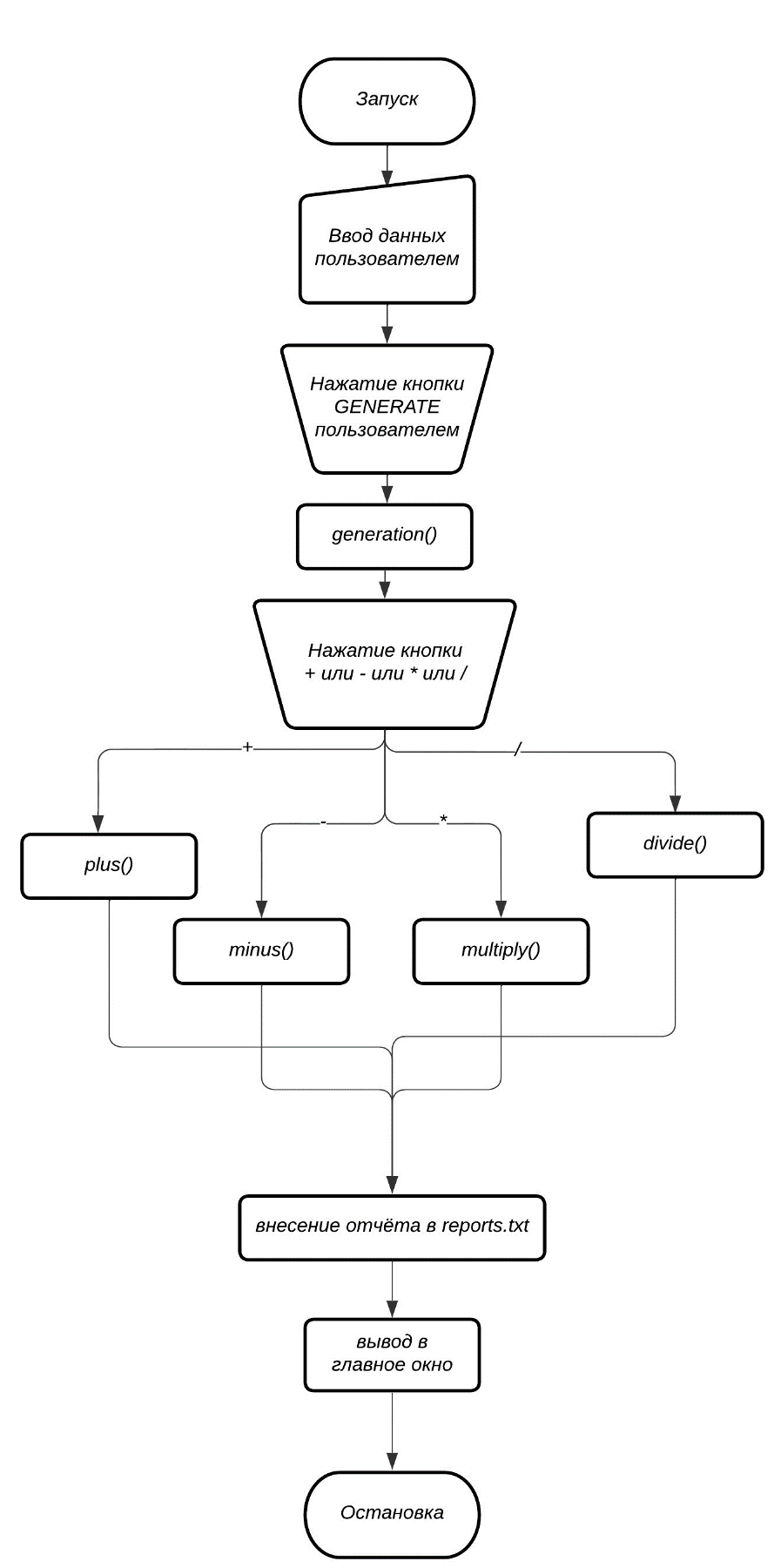


Также реализована функция внесения отчета совершенной операции в “reports.txt”

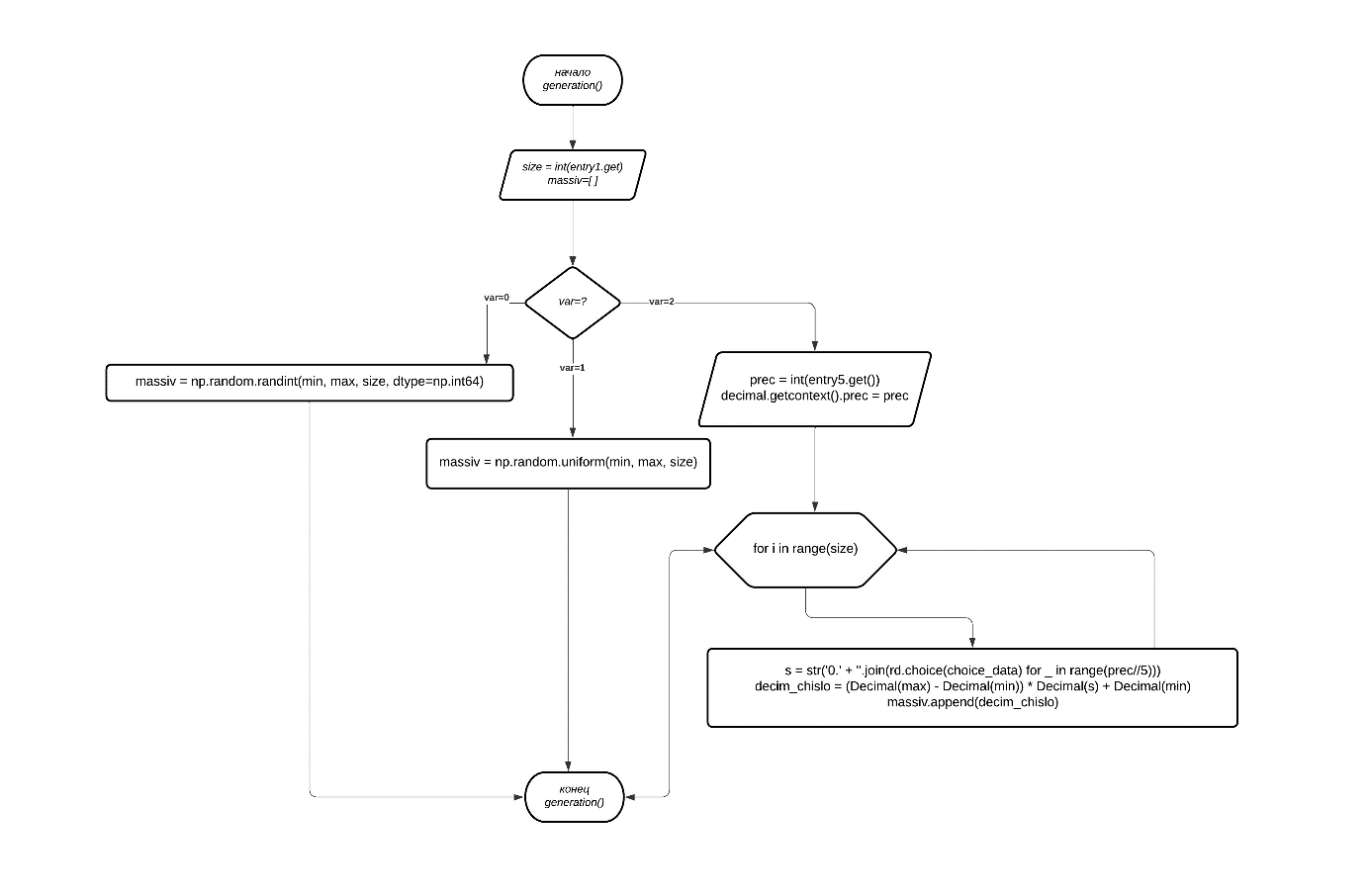
* + **Внесение отчёта в сторонний файл**:



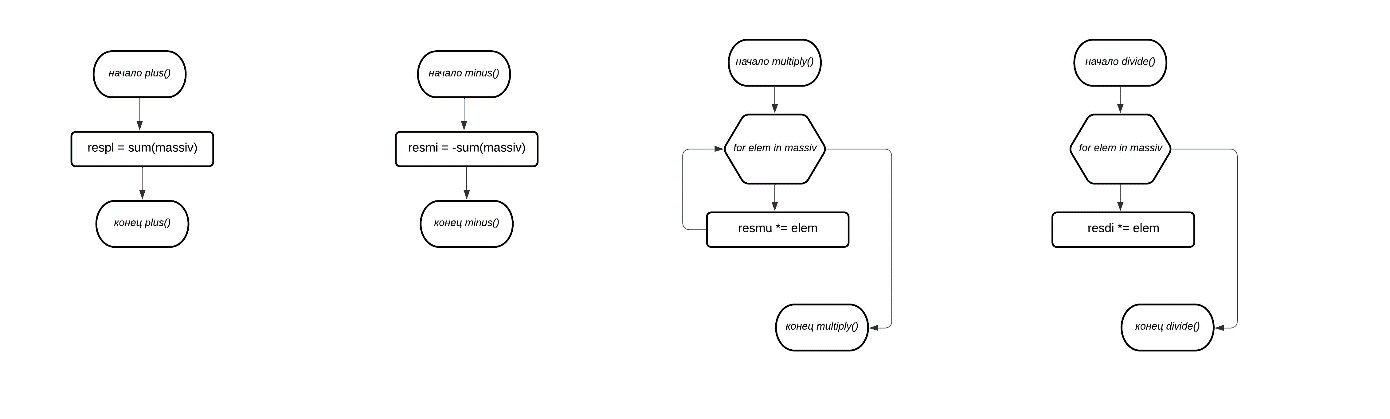
## 5.5 UML диаграммы для разработчика



Основная программа



Функция generation()

Функции арифметических операций