******

Отчёт по “MathGraphPy”.

Выполнил студент: Волков Максим

Группа: П1-18

Преподаватель: Наталья Юрьевна Бобкова

Вячеслав Николаевич Попов

Королев, 2022

# Оглавление

[1 Оглавление 2](#_Toc101170934)

[Схема в Ramus 4](#_Toc101170935)

[Техническое задание. 7](#_Toc101170936)

[1 Введение 7](#_Toc101170937)

[1.1 Наименование программы 7](#_Toc101170938)

[1.2 Краткая характеристика области 7](#_Toc101170939)

[1.3 Основания для разработки 7](#_Toc101170940)

[1.4 Назначение разработки 7](#_Toc101170941)

[1.5 Функциональное назначение 7](#_Toc101170942)

[1.6 Эксплуатационные назначения 7](#_Toc101170943)

[2 Требования к программе или программному изделию 7](#_Toc101170944)

[2.1 Требования к функциональным характеристикам 7](#_Toc101170945)

[2.1.1 Требования к составу выполняемых функций 7](#_Toc101170946)

[2.1.2 Требование к организации входных и выходных данных 7](#_Toc101170947)

[2.1.3 Требования к временным характеристикам 7](#_Toc101170948)

[2.2 Требования к надежности 7](#_Toc101170949)

[2.2.1 Требования к обеспечению надёжного функционирования программы 7](#_Toc101170950)

[2.2.2 Время восстановления после отказа 8](#_Toc101170951)

[2.2.3 Отказы из-за некорректных действий оператора 8](#_Toc101170952)

[2.3 Условия эксплуатации 8](#_Toc101170953)

[2.3.1 Климатические условия эксплуатации 8](#_Toc101170954)

[2.3.2 Требования к численности и квалификации 8](#_Toc101170955)

[2.4 Требования к составу и параметры технических средств 8](#_Toc101170956)

[2.5 Требования к информационной и программной совместимости 8](#_Toc101170957)

[2.6 Требования к маркировке и упаковке 8](#_Toc101170958)

[2.7 Требования к транспортированию и хранению 8](#_Toc101170959)

[2.8 Специальные требования 8](#_Toc101170960)

[3 Требования к программной документации 9](#_Toc101170961)

[4 Технико-экономические показатели 9](#_Toc101170962)

[5 Стадии и этапы разработки 9](#_Toc101170963)

[6 Порядок контроля и приемки 10](#_Toc101170964)

[Руководство оператора 10](#_Toc101170965)

[1. Назначение программы 10](#_Toc101170966)

[1.2. Функциональное назначение программы 10](#_Toc101170967)

[1.3. Эксплуатационное назначение программы 10](#_Toc101170968)

[2. Условия выполнения программы 10](#_Toc101170969)

[2.1. Минимальный состав аппаратных средств 10](#_Toc101170970)

[2.2. Требования к пользователю 11](#_Toc101170971)

[3. Выполнение программы 11](#_Toc101170972)

[3.1. Загрузка и запуск программы 11](#_Toc101170973)

[4. Сообщения оператору 12](#_Toc101170974)

[Руководство программиста 12](#_Toc101170975)

[1. Назначение и условия применения программы 12](#_Toc101170976)

[1.1. Функции программы 12](#_Toc101170977)

[2. Условия выполнения программы 12](#_Toc101170978)

[3. Характеристики программы 12](#_Toc101170979)

[3.1 Временные характеристики 12](#_Toc101170980)

[3.2. Режим работы 12](#_Toc101170981)

[3.3. Загрузка и запуск программы 12](#_Toc101170982)

[4. Обращение к программе 12](#_Toc101170983)

[5. Входные и выходные данные 12](#_Toc101170984)

[6. Сообщения оператору 12](#_Toc101170985)

[7 Выполнение программы 13](#_Toc101170986)

[7.1 Запуск программы 13](#_Toc101170987)

[7.2 Отладка 14](#_Toc101170988)

# Схема в Ramus

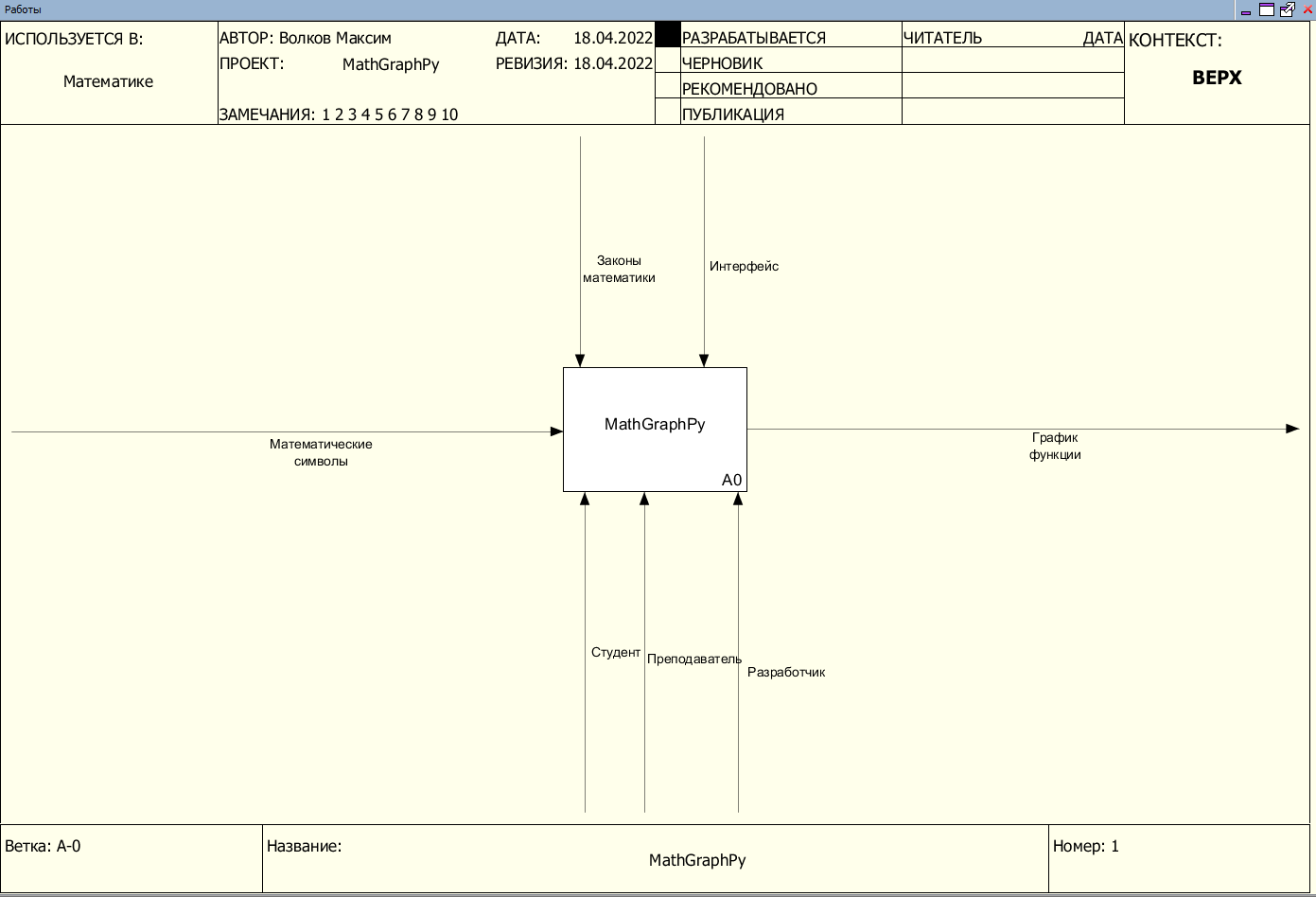


Рисунок 1. Первый уровень

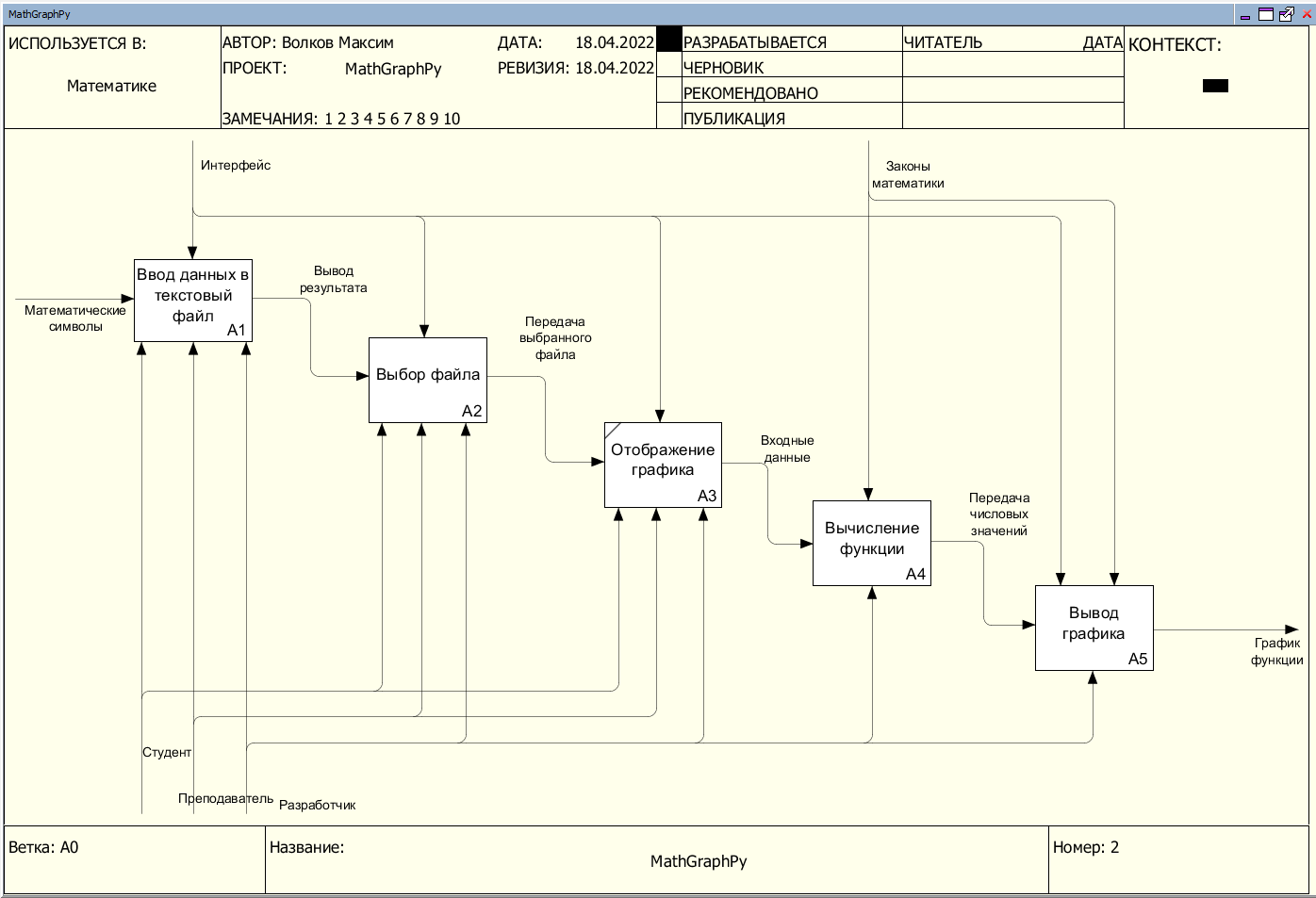
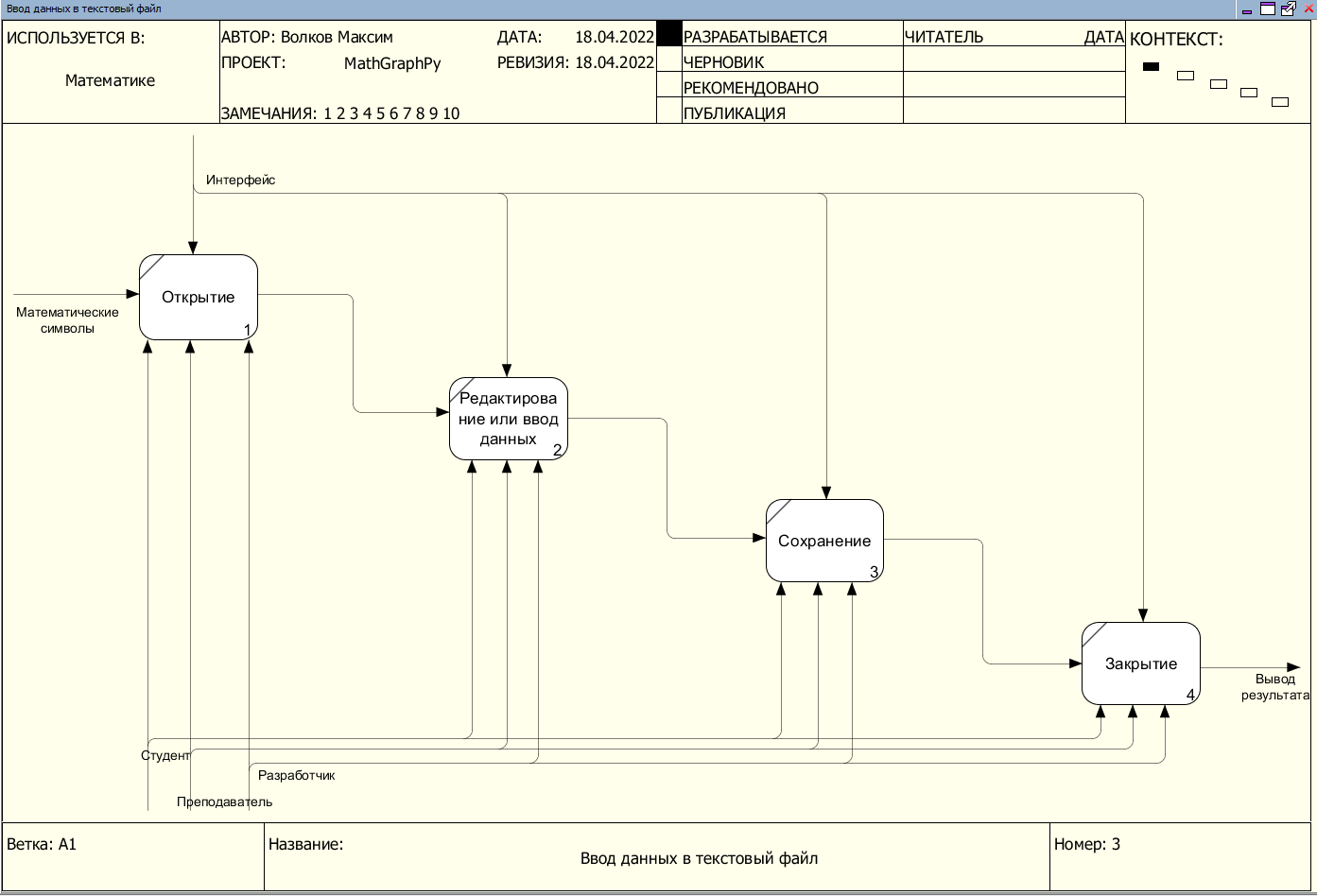


Рисунок 2. Второй уровень

Рисунок 3. DFD уровень ветки A1

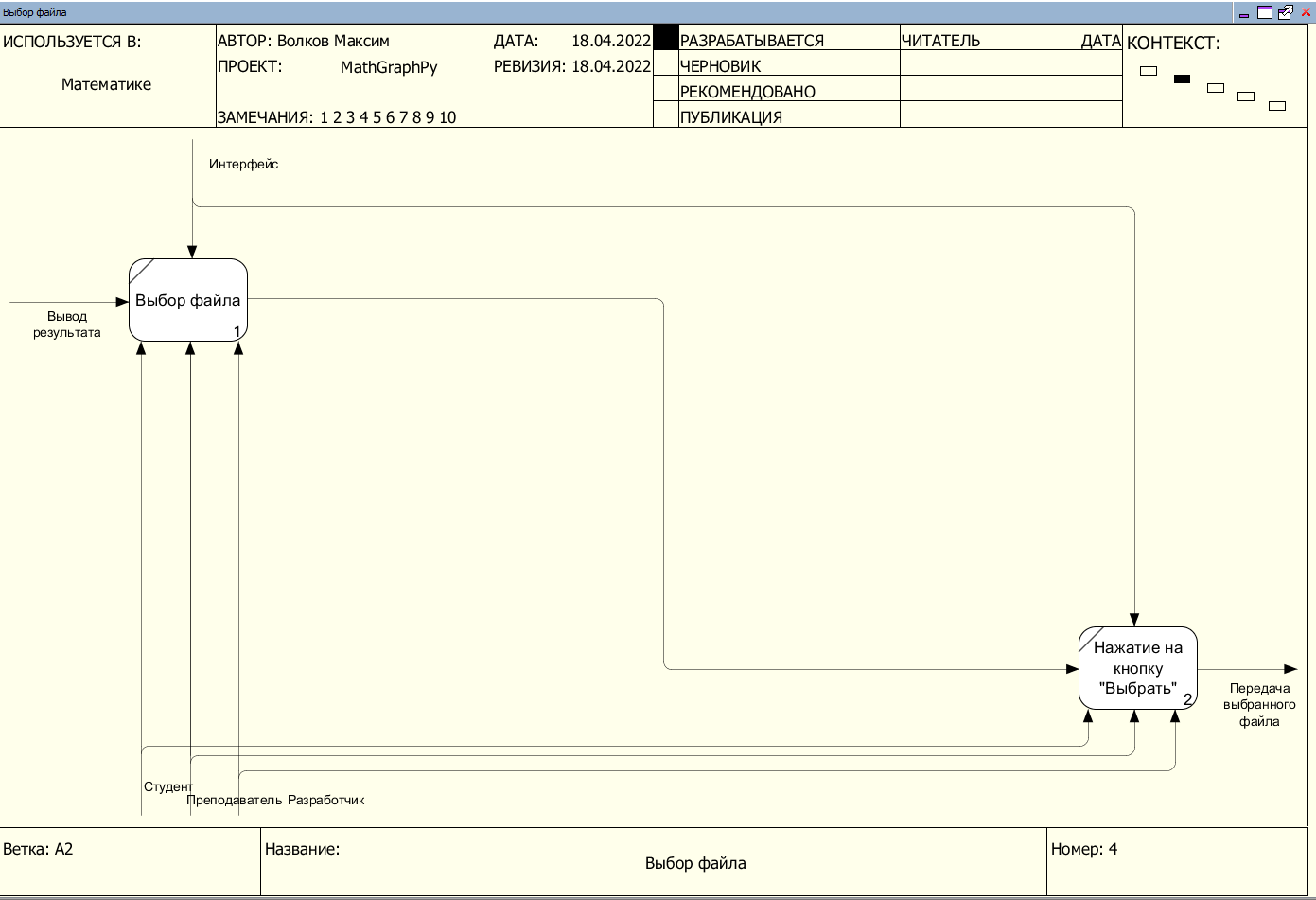


Рисунок 4. DFD уровень ветки A2

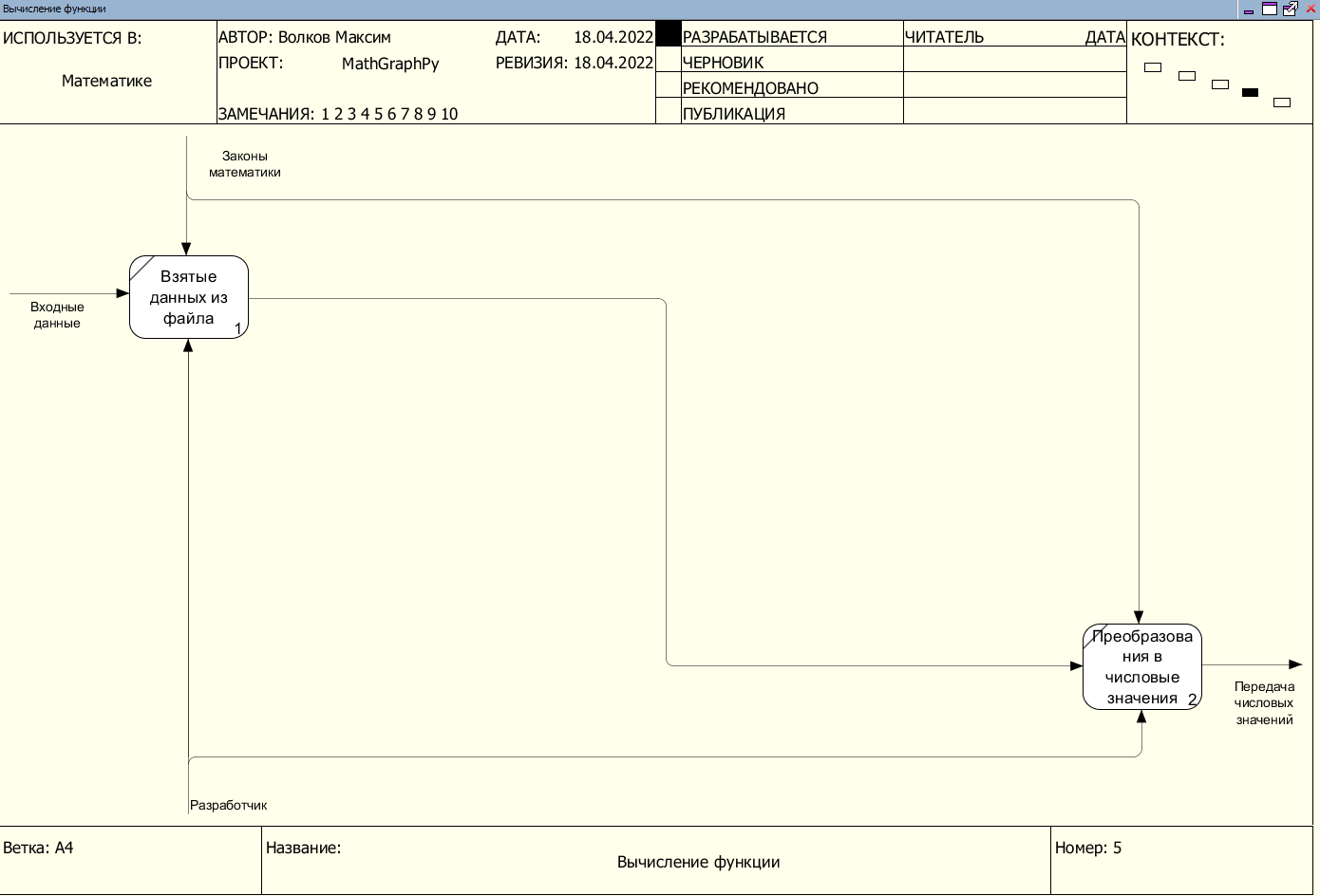


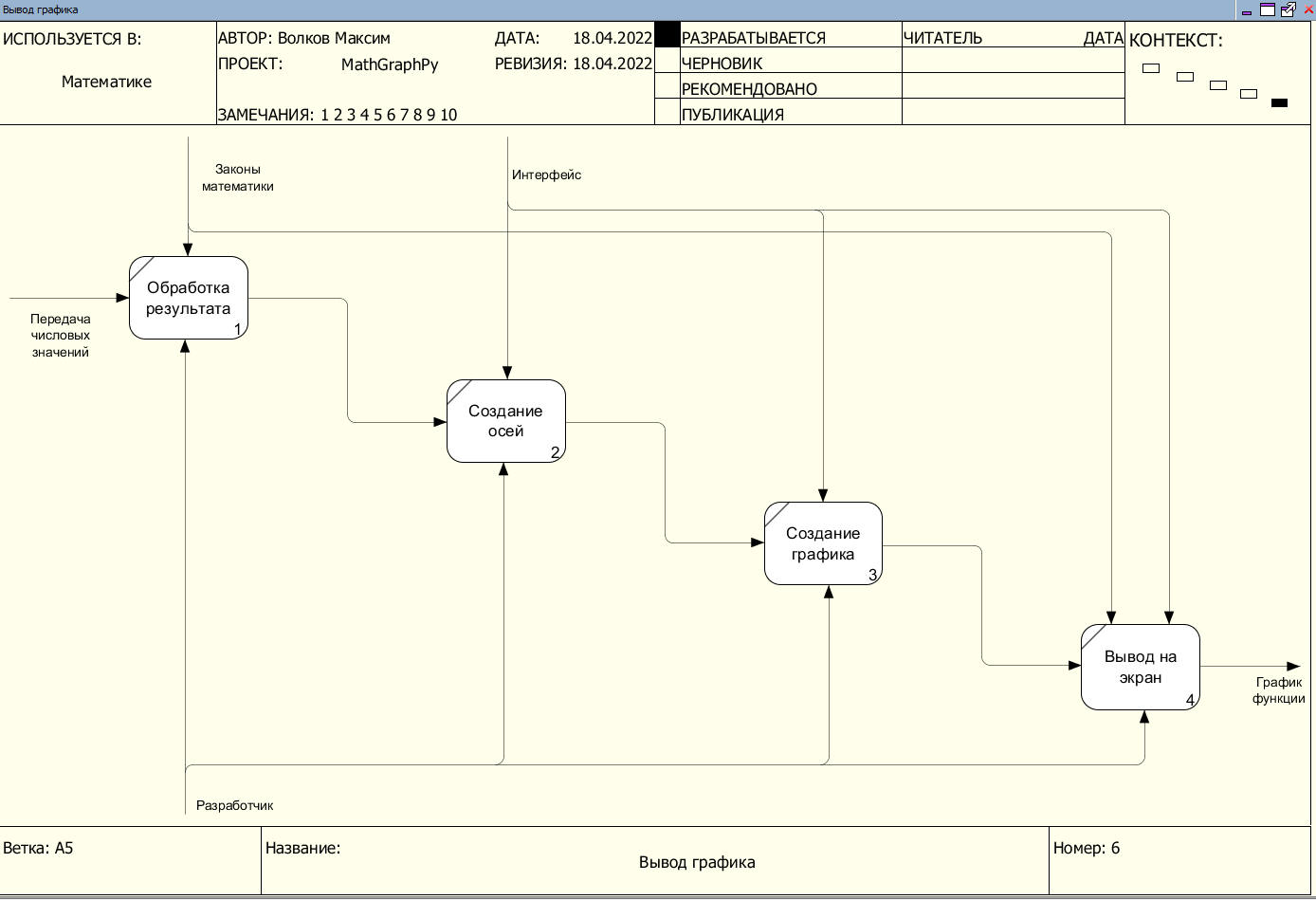
Рисунок 5. DFD уровень ветки A4 

Рисунок 6. DFD уровень ветки A5

# Техническое задание.

# 1. Введение

## Наименование программы

Наименование программы – “MathGraphPy ”.

## Краткая характеристика области

Приложение “MathGraphPy” предназначена для построения графика функции на оси OX и OY.

## Основания для разработки

Основанием для разработки является Договор 12 от 01.08.2020. Договор утвержден Директором Технологического Университета имени дважды Героя Советского Союза, летчика-космонавта А.А. Леонова, ККМТ. Ниже именуемым в дальнейшем Заказчиком и Волков Максим Александрович, именуемым в дальнейшем исполнителем.

Согласно Договору, Исполнитель обязан разработать и установить программу “MathGraphPy” на оборудовании Заказчика не позднее 19.04.2021, предоставить исходные коды и документацию к разработанной системе не позднее 19.04.2021.

Наименование темы разработки – “MathGraphPy”

## Назначение разработки

Программа “MathGraphPy” предназначается для использования в колледже студентами и преподавателями.

## Функциональное назначение

Программа позволяет составить график функций.

.

## Эксплуатационные назначения

Приложение должно эксплуатироваться только в учебных целях.

# Требования к программе или программному изделию

## Требования к функциональным характеристикам

### Требования к составу выполняемых функций

После запуска программы пользователю отображается график. В графике функций показана графическое представление математической функции

### Требование к организации входных и выходных данных

Входные данные являются цифры, переменные и математические символы. Подаются значение в текстовый файл, после чего выводится на график функций.

### Требования к временным характеристикам

После того, как преподаватель или студент написали значения в текстовый файл, выводятся данные на график функций.

## Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности версии программы.

### Требования к обеспечению надёжного функционирования программы

Надежное функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения;
* регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
* регулярным выполнением требований. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

### Время восстановления после отказа

Время восстановления после отказа неопределённо даже при условии соблюдения условий эксплуатации технических и программных средств. Так как отказ может быть вызвано неисправностью технических средств, фатальным сбоем операционной системы.

### Отказы из-за некорректных действий оператора

Отказы программы возможен из-за внутренних багов или неполадок самого приложения.

## Условия эксплуатации

Программа запускается на компьютере. Окно программы должно быть открыто в любой IDE, где есть Python, должна быть возможность запустить текстовый файл вместе с программой. Запуск программы осуществляться студентом или преподавателем.

### Климатические условия эксплуатации

Специальные условия не требуются.

### Требования к численности и квалификации

Программа не требует проведения каких-либо видов обслуживания.

## Требования к составу и параметры технических средств

Состав технических средств:

* Компьютер преподавателя или студента, включающий в себя:
* Процессор с тактовой частотой, не менее 1 ГГц;
* Оперативная память объемом, не менее 500Мб;
* Жёсткий диск, не менее 10 Гб;
* Операционная система Windows.

## Требования к информационной и программной совместимости

Должен быть соблюден правильный порядок цифр и их количество в текстовом файле.

## Требования к маркировке и упаковке

Программа передаётся на флэшке в виде папки. Специальных требований к маркировке не предъявляется.

## Требования к транспортированию и хранению

Для транспортировки программы ничего не требуется.

## Специальные требования

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы.

# 3. Требования к программной документации

Предварительный состав программной документации:

* техническое задание (включает описание применения);
* программа и методика испытаний;
* руководство системного программиста;
* руководство оператора;
* ведомость эксплуатационных документов;

# 4. Технико-экономические показатели

Программа “MathGraphPy” пригодная учебных целях. Эта программа упращает вычисления и ускоряет процесс.

# 5. Стадии и этапы разработки

Всего этапов разработки:

* Разработка требований
* Проектирование
* Реализация
* Тестирование
* Внедрение

На стадии “Разработка требований” должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии “Проектирование” утверждается интерфейс программы, а также внутренние устройство программы.

На стадии “Реализации” должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;

На стадии “Тестирование” проверяется работа функций программы и в целом корректность её работы.

На стадии “Внедрение” должен быть выполнено внедрение заказчику на компьютер.

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* постановка задачи;
* определение и уточнение требований к техническим средствам;
* определение требований к программе;
* определение стадий, этапов и сроков разработки программы и документации на нее;
* согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* разработка, согласование и утверждение порядка и методики испытаний;
* проведение приемо-сдаточных испытаний;
* корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

# 6. Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний. На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

# Руководство оператора

# Назначение программы

Программа “MathGraphPy” предназначается для использования студентами и преподавателями.

## 1.1. Функциональное назначение программы

Программа “MathGraphPy” будет использоваться в колледже студентами и преподавателями. Эта программа позволяет студентам или преподавателям продемонстрировать математическую функцию на графике.

Функциональные возможности:

* Добавление, удаление и изменение текстового файла
* Просмотр графика функций

## Эксплуатационное назначение программы

Программа должна эксплуатироваться студентами и преподавателями колледжа. Программу можно начать использовать, когда вписаны определенные значения в текстовый файл.

# 2. Условия выполнения программы

## 2.1. Минимальный состав аппаратных средств

Минимальный состав используемых технических (аппаратных) средств:

* Процессор с тактовой частотой, не менее 1 ГГц;
* Оперативная память объемом, не менее 500Мб;
* Жёсткий диск, не менее 10 Гб;
* Операционная система Windows

## 2.2. Требования к пользователю

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы Windows.

# 3. Выполнение программы

## 3.1. Загрузка и запуск программы

Загрузка на компьютере осуществляется с помощью любого IDE, где есть Python, который можно скачать в интернете. Детальные сведения по установке Python должно быть известно оператору.

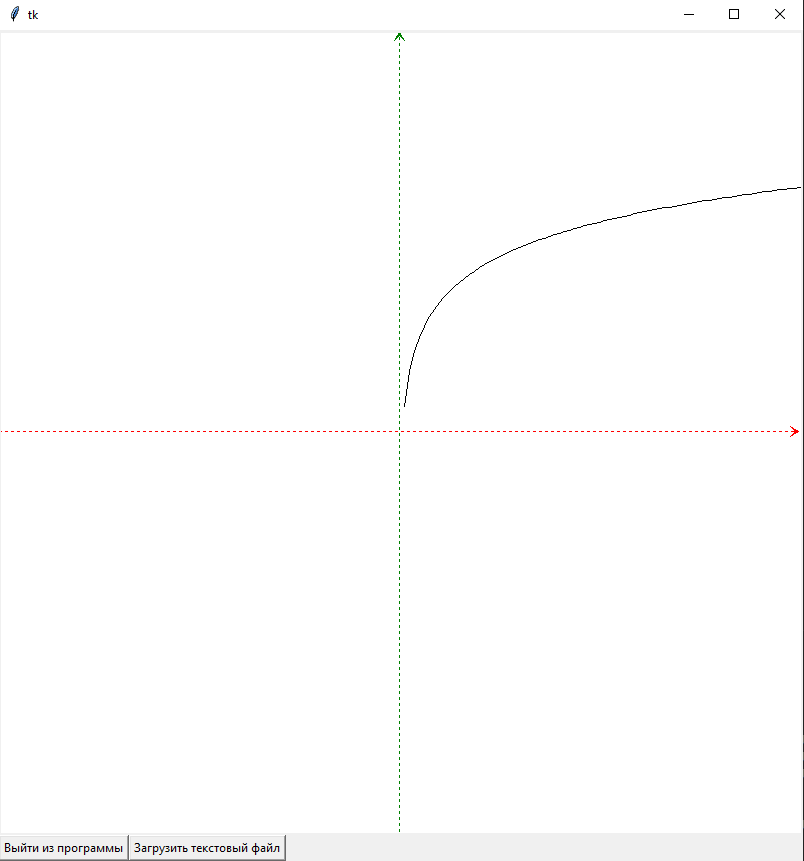


Рисунок 7. График функций.

Запуск настольного приложения осуществляется нажатием файл с названием main с расширением .py. Оператор должен осуществлять запуск настольного приложения на своём компьютере куда и было установлено app.py и main.py.

# 4. Сообщения оператору

При неправильном вводе данных в текстовом файле, адресуется следующая ошибка.

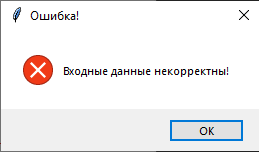


Рисунок 8. Ошибка ввода.

# Руководство программиста

# Назначение и условия применения программы

Программа “MathGraphPy” предназначается для использования студентами и преподавателями.

## Функции программы

Программа “MathGraphPy” будет использоваться в колледже студентами и преподавателями. Эта программа позволяет студентам или преподавателям продемонстрировать математическую функцию на графике.

# Условия выполнения программы

Минимальный состав используемых технических (аппаратных) средств:

* Процессор с тактовой частотой, не менее 1 ГГц;
* Оперативная память объемом, не менее 500Мб;
* Жёсткий диск, не менее 10 Гб;
* Операционная система Windows

# Характеристики программы

## 3.1 Временные характеристики

Временные характеристики отсутствуют в приложении.

## 3.2. Режим работы

У данного приложения режим работы по доступу на изменения

## 3.3. Загрузка и запуск программы

Загрузка на компьютере осуществляется с помощью любого IDE, где есть Python, который можно скачать в интернете. Детальные сведения по установке Python должно быть известно оператору.

# Обращение к программе

Обращение к программе производится нажатием на кнопку запуска в специальных IDE.

# Входные и выходные данные

Входными данными: являются цифры, которые нужно вписывать в текстовый файл.

Выходные данные: Вывод графического графика функции

# Сообщения оператору

При неправильном вводе данных в текстовом файле, адресуется следующая ошибка.

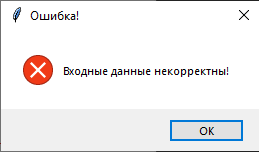


Рисунок 9. Ошибка ввода.

# Выполнение программы

## Запуск программы

Для запуска программы требуется запустить IDLE, затем выбрать main.py, прожать кнопку Run и после этого нажать Run Module

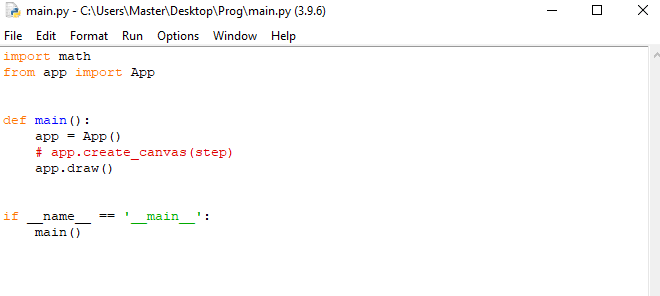


Рисунок 10. Файл запуска

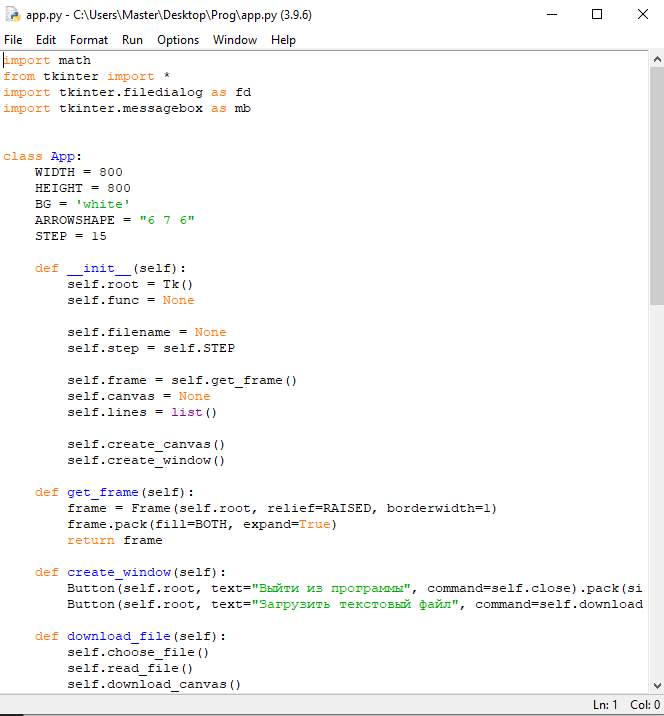


Рисунок 11. Файл с графиком функции.

После этого выводится график функции.

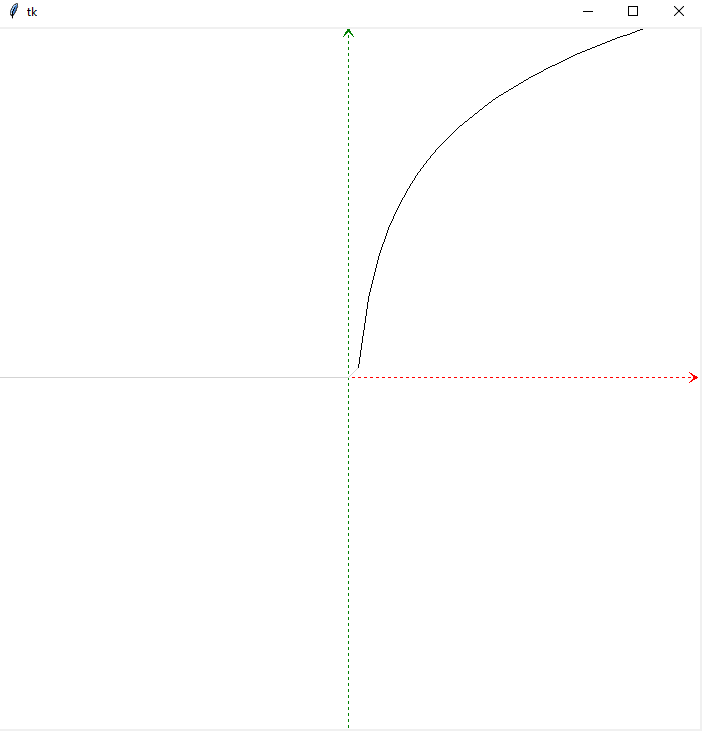


Рисунок 12. График функции.

## 7.2. Отладка

При запуске программы была выявлена ошибка.

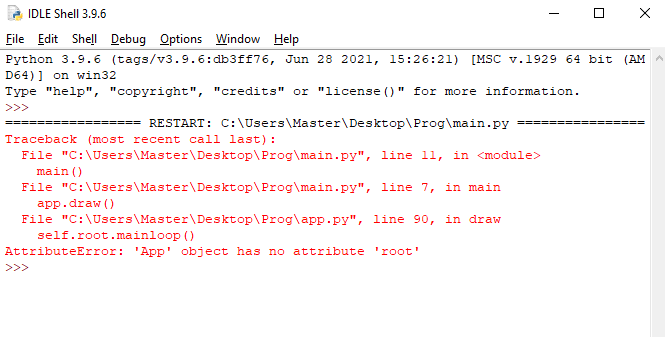


Рисунок 13. Ошибка в ходе запуска.

Приложение 1. main.py

import math

from app import App

def main():

app = App()

app.draw()

if \_\_name\_\_ == '\_\_main\_\_':

main()

Приложение 2. app.py

import math

from tkinter import \*

import tkinter.filedialog as fd

import tkinter.messagebox as mb

class App:

WIDTH = 800

HEIGHT = 800

BG = 'white'

ARROWSHAPE = "6 7 6"

STEP = 15

def \_\_init\_\_(self):

self.root = Tk()

self.func = None

self.filename = None

self.step = self.STEP

self.frame = self.get\_frame()

self.canvas = None

self.lines = list()

self.create\_canvas()

self.create\_window()

def get\_frame(self):

frame = Frame(self.root, relief=RAISED, borderwidth=1)

frame.pack(fill=BOTH, expand=True)

return frame

def create\_window(self):

Button(self.root, text="Выйти из программы", command=self.close).pack(side=LEFT)

Button(self.root, text="Загрузить текстовый файл", command=self.download\_file).pack(side=LEFT)

def download\_file(self):

self.choose\_file()

self.read\_file()

self.download\_canvas()

def choose\_file(self):

filetypes = (("Текстовый файл", "\*.txt"),)

filename = fd.askopenfilename(title="Открыть файл", initialdir="/", filetypes=filetypes)

if filename:

print(filename)

self.filename = filename

def read\_file(self):

input\_str = open(self.filename).read()

print(input\_str)

try:

a, b, self.step = list(map(float, input\_str.split()))

except Exception:

mb.showerror("Ошибка!", "Входные данные некорректны!")

self.func = lambda x: a \* math.log(x) + b

def create\_canvas(self):

self.canvas = Canvas(self.root, width=self.WIDTH, height=self.HEIGHT, bg=self.BG)

self.canvas.pack()

self.set\_guide\_lines()

def download\_canvas(self):

points = self.get\_points(self.func)

self.rendering\_points(points)

def set\_guide\_lines(self):

self.canvas.create\_line(0, self.HEIGHT // 2, self.WIDTH, self.HEIGHT // 2, fill='red', width=1, arrow=LAST, dash=(1, 1), activefill='red4', arrowshape=self.ARROWSHAPE)

self.canvas.create\_line(self.WIDTH // 2, self.HEIGHT, self.WIDTH // 2, 0, fill='green', width=1, arrow=LAST, dash=(1, 1), activefill='darkgreen', arrowshape=self.ARROWSHAPE)

@staticmethod

def get\_points(func):

points = list()

for x in range(1, 100):

try:

points.append((x, func(x)))

except Exception:

points.append((x, None))

return points

def rendering\_points(self, points):

last\_point = points[0]

for line in self.lines:

self.canvas.delete(line)

for i in range(1, len(points)):

x1, y1 = last\_point

x2, y2 = points[i]

clr = 'black'

if y1 is None or y2 is None:

last\_point = points[i]

continue

pk = self.canvas.create\_line(

x1 \* self.step + self.WIDTH // 2, -y1 \* self.step + self.HEIGHT // 2,

x2 \* self.step + self.WIDTH // 2, -y2 \* self.step + self.HEIGHT // 2,

fill=clr

)

self.lines.append(pk)

last\_point = points[i]

def draw(self):

self.root.mainloop()

def close(self):

self.root.quit()