1. Техническое задание.

**1 Введение**

**1.1 Наименование программы**

Наименование программы – «Выполнение функции».

**1.2 Краткая характеристика области применения**

Система «Выполнение функции» показывает пользователю график, который рисует программа в осях OX, OY. Программа берет значения отрисовки из текстового документа.

**2 Основания для разработки**

Основанием для разработки является Договор 12 от 18.04.2022. Договор утвержден Директором ООО «Гос. образование» \_\_\_, именуемым в дальнейшем Заказчиком, \_\_\_ (самозанятый), именуемым в дальнейшем исполнителем, 18.04.2022.

Согласно Договору, Исполнитель обязан разработать и установить систему «Выполнение функции» на оборудовании Заказчика не позднее 19.04.2022, предоставить исходные коды и документацию к разработанной системе не позднее 19.04.2022.

Наименование темы разработки – «Разработка информационно-справочной системы Выполнение функции».

Условное обозначение темы разработки (шифр темы) – «Функция-01».

**3 Назначение разработки**

Программа будет использоваться для отображения заданной в текстовом документе функции.

**3.1 Функциональное назначение**

Для пользователя - видит график который выводит программа.

**3.2 Эксплуатационное назначение**

Программа должна эксплуатироваться для экзамена. Запущенная программа, позволяет просматривать построенный график, который читается с текстового файла.

**4 Требования к программе или программному изделию**

**4.1 Требования к функциональным характеристикам**

**4.1.1 Требования к составу выполняемых функций**

Программа позволяет выполнять поставленные задачи любому пользователю:

* Просмотр выведенного графика
* Редактирование текстового документа с заданными параметрами графика

При просмотре графика выводится окно:

Примерный вид окна для любого пользователя выглядит так – выведенный график.

**4.1.2 Требования к организации входных и выходных данных**

Данные с входными параметрами графика содержатся в текстовом документе. Текстовый документ обеспечивает доступ к параметрам графика и обеспечивает считывание для отрисовки графика.

**4.1.3 Требования к временным характеристикам**

После изменения пользователем данных, находящихся в текстовом документе, новая информация о товарах отображается не позднее, чем через 5 секунд.

**4.2 Требования к надежности**

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети (связи приложений оператора и текстового документа).

**4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы**

В связи с тем, что в текстовом документе хранятся информация о построении графика — текстовый документ стоит резервировать (резервирование замещением).

Надежное (устойчивое) 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»;

**4.2.2 Время восстановления после отказа**

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

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

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

**4.3 Условия эксплуатации**

Программа (клиент) запускается на компьютерах оператора (пользователя) доступна экзаменаторам. Текстовый документ находится на компьютере оператора (пользователя). Должна существовать устойчивая связь по сети между пользователем и текстовым документом.

Окно программы должно быть открыто не на весь экран, должна быть возможность закрыть, свернуть приложение или запустить любое стороннее программное обеспечение.

**4.3.1 Климатические условия эксплуатации**

Специальные условия не требуются.

**4.3.2 Требования к видам обслуживания**

Программа не требует проведения каких-либо видов обслуживания.

**4.3.3 Требования к численности и квалификации персонала**

При установке и настройке системы необходим обычный пользователь. В процессе эксплуатации с программой работает один пользователь.

Пользователь должен иметь среднее образование. В перечень задач, выполняемых пользователем, должны входить:

* установка клиентских приложений;
* настройка текстового документа;

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

К квалификации посетителя просматривающего график функции специальные требования не предъявляются.

**4.4 Требования к составу и параметрам технических средств**

Состав технических средств:

* Компьютер оператора, включающий в себя:
  + процессор x86 с тактовой частотой, не менее 1 ГГц;
  + оперативную память объемом, не менее 1 Гб;
  + видеокарту, монитор, мышь, клавиатура.
* Компьютер посетителя, включающий в себя:
  + процессор x86 с тактовой частотой, не менее 1 ГГц;
  + оперативную память объемом, не менее 1 Гб;
  + видеокарту, монитор, мышь.
* Два компьютера для СУБД (основной и резервный), включающий в себя:
  + процессор x86 с тактовой частотой, не менее 1 ГГц;
  + оперативную память объемом, не менее 1 Гб;
  + видеокарту, монитор, мышь.

**4.5 Требования к информационной и программной совместимости**

Приложения оператора обмениваются с текстовым документом сообщениями по средству одной папки в котором находятся два файла, текстовый документ и «Python-файл».

**4.6 Требование к маркировке и упаковке**

Программное изделие передается по сети Интернет в виде архива — загружается с официального сайта производителя. Специальных требований к маркировке не предъявляется. Для проверки подлинности программного обеспечения рекомендуется проверять контрольные суммы загруженных файлов со значениями, указанными на официальном сайте.

**4.7 Требования к транспортированию и хранению**

Специальных требований не предъявляется.

**4.8 Специальные требования**

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

**5 Требования к программной документации**

Предварительный состав программной документации:

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

**6 Технико-экономические показатели**

Программа «Выполнение функции» пригодна для экзаменационного билета.

Функциональность программы совпадает с аналогами.

**7 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. техническое задание;
2. технический (и рабочий) проекты;
3. внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

* разработка программы;
* разработка программной документации;
* испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

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 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.

**8 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.

2.

import matplotlib.pyplot as plt

from matplotlib.ticker import MultipleLocator

import numpy as np

import re

plt.style.use('seaborn-whitegrid')

fig = plt.figure(num='Билет №3')

ax = fig.add\_subplot(1, 1, 1)

ax.grid(which='both')

ax.grid(which='minor', alpha=0.6)

ax.grid(which='major', alpha=1)

# with open('coef3.txt', 'r') as file:

# line = file.readline().split()

class App:

FUNCTION\_TEMPLATE = '{0} \*\* x + {1}'

@staticmethod

def average(arr):

return max(arr) - min(arr)

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

self.ox\_points = []

self.oy\_points = []

with open(filename, 'r') as file:

line = file.readline().split()

self.abc = line[0:2]

self.function = self.FUNCTION\_TEMPLATE.format(\*self.abc)

if len(line) > 2:

self.ct\_x\_step = float(1 / eval(line[2].replace(',', '.'))) # 1 > x > 0

self.ct\_y\_step = float(1 / eval(line[3].replace(',', '.'))) # 1 > y > 0

def calc\_point(self, x):

return eval(self.function.replace('x', f'({str(x)})'))

def create\_points(self, start=-20, end=20, step=2.0):

if re.search(r'^[^}]+$', self.function):

for X in np.arange(start, end, step):

self.ox\_points.append(X)

self.oy\_points.append(self.calc\_point(X))

else:

raise AttributeError(f'Incorrect function: {self.function}')

def create\_plot(self, ct\_x\_steps=None, ct\_y\_steps=None, int\_steps=False):

print(self.ct\_x\_step, self.ct\_y\_step)

if not self.ox\_points:

raise Exception('Не заданы точки!')

ct\_x\_steps = ct\_x\_steps or self.ct\_x\_step

ct\_y\_steps = ct\_y\_steps or self.ct\_y\_step

ox\_step = self.average(self.ox\_points) / ct\_x\_steps

oy\_step = self.average(self.oy\_points) / ct\_y\_steps

if int\_steps:

ox\_step = ox\_step > 1 and int(ox\_step) or ox\_step

oy\_step = oy\_step > 1 and int(oy\_step) or oy\_step

ox\_step = float('{:.2f}'.format(ox\_step))

oy\_step = float('{:.2f}'.format(oy\_step))

print(f'Шаг по X: {ox\_step}', f'Кол-во отрезков: {ct\_x\_steps}', min(self.ox\_points), max(self.ox\_points))

print(f'Шаг по Y: {oy\_step}', f'Кол-во отрезков: {ct\_y\_steps}', min(self.oy\_points), max(self.oy\_points))

ax.set\_xticks(np.arange(min(self.ox\_points), max(self.ox\_points), ox\_step))

ax.set\_yticks(np.arange(min(self.oy\_points), max(self.oy\_points), oy\_step))

ax.xaxis.set\_minor\_locator(MultipleLocator(ox\_step / 5))

ax.yaxis.set\_minor\_locator(MultipleLocator(oy\_step / 5))

plt.plot(self.ox\_points, self.oy\_points)

plt.show()

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

app = App('coef3.txt')

app.create\_points(10, 20, step=0.05)

app.create\_plot(10, 10, int\_steps=True)

3. Отладка и тестирование.

**Описание применения средств отладки**

В этом разделе показано умение применять средства отладки.

В ходе написания программы при попытке запустить скрипт было получено данное сообщение:

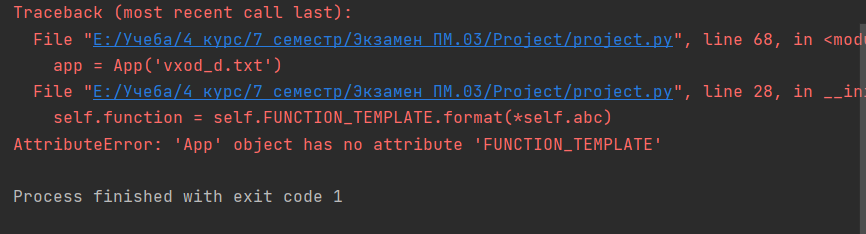


Рисунок 1. До применения средств отладки

После получения данного сообщения были просмотрены 28 и 68 строки модуля project.py и была обнаружена ошибка, которая впоследствии была устранена, а после попытки запуска скрипта получено данное сообщение:

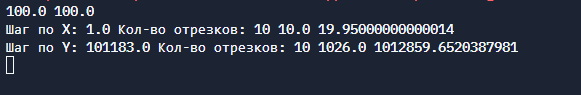


Рисунок 2. После применения средств отладки

Это означает что ошибка была устранена и скрипт запустился.

**Описание тестовых наборов модулей**

В этом разделе будут продемонстрированы результаты тестирования «черного ящика».

Тест 1. Корректный ввод входных данных и проверка правильной работы программы.

Действия: Ввод корректных входных данных.

Ожидаемый результат: Правильная работа программы.

Результат теста:

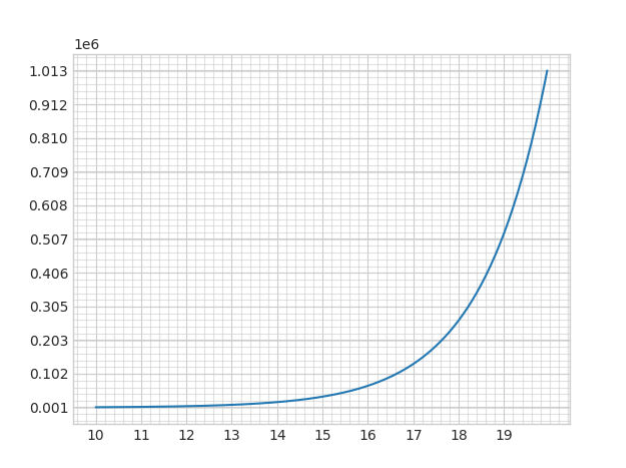


Рисунок 1. Корректный вывод оповещения.

Тест 2. Некорректный ввод входных данных и проверка правильной работы программы.

Действия: Ввод некорректных входных данных

Ожидаемый результат: Сообщение об ошибке

Результат теста:

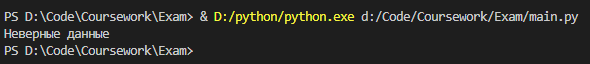


Рисунок 2. Некорректный вывод оповещения

Тест 3. Ввод пустого значения данных

Действия: Удаление любых символов в текстовом документе

Ожидаемый результат: Сообщение что список пуст

Результат теста:

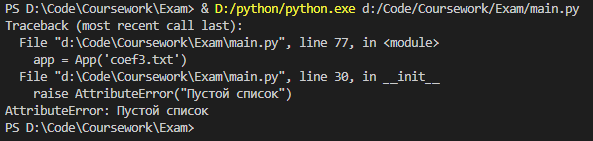
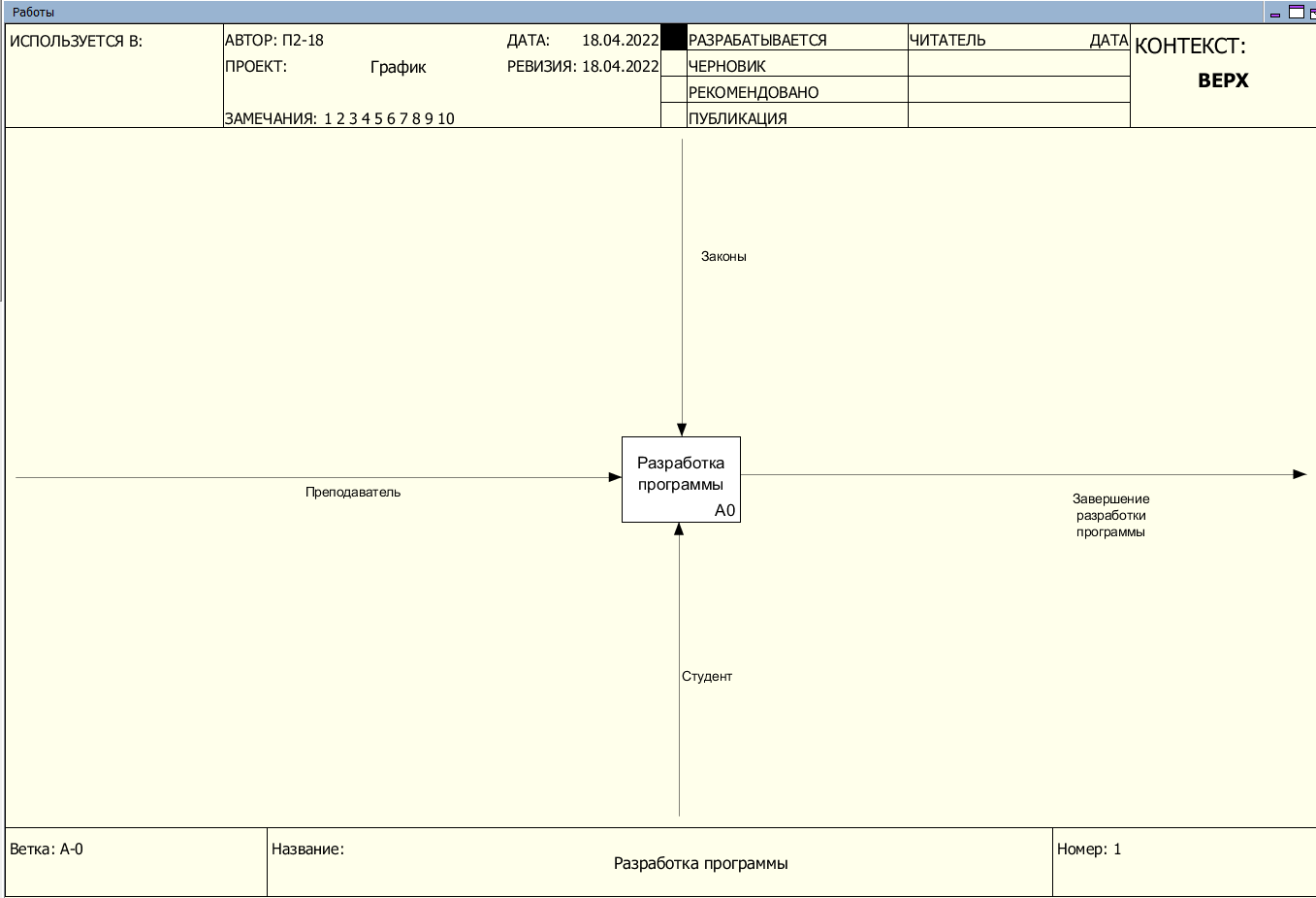
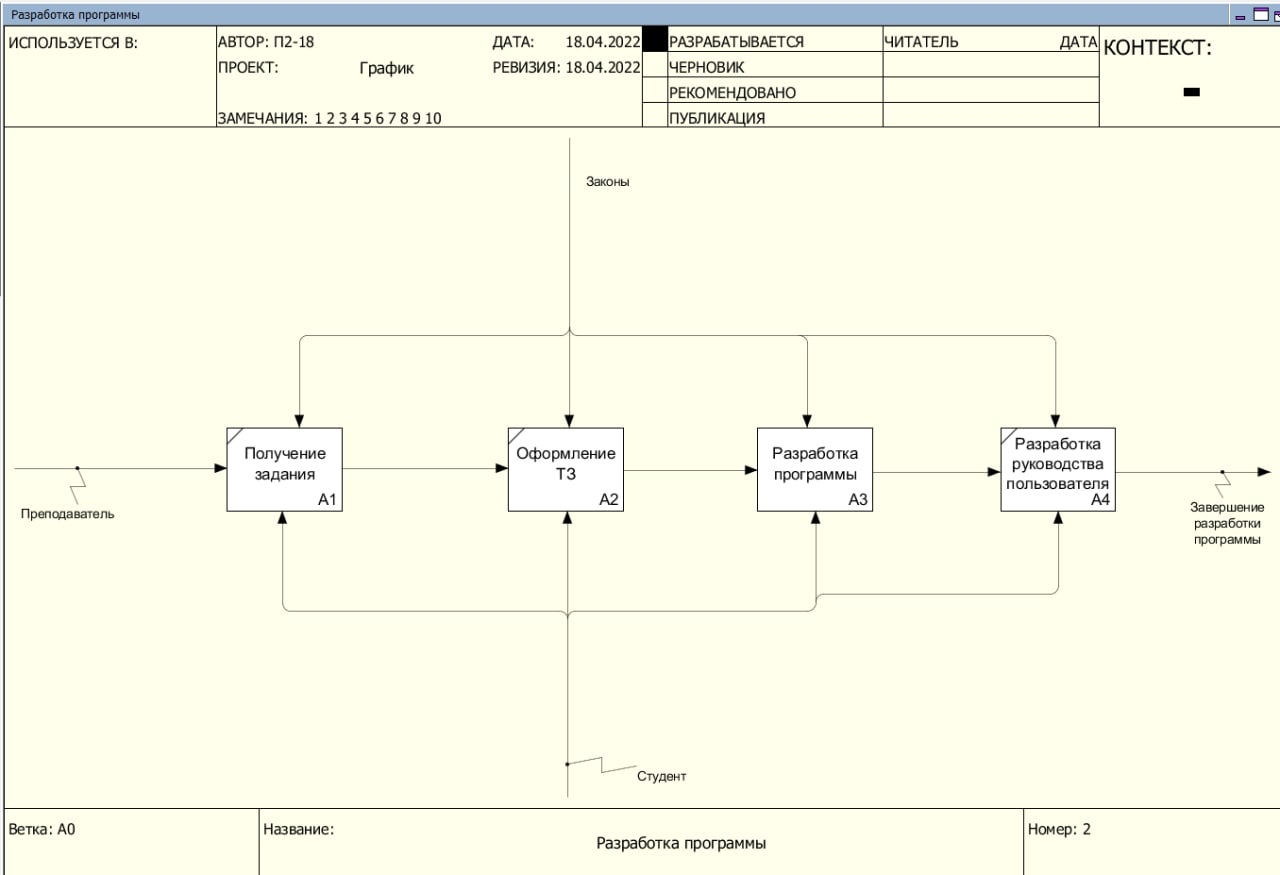
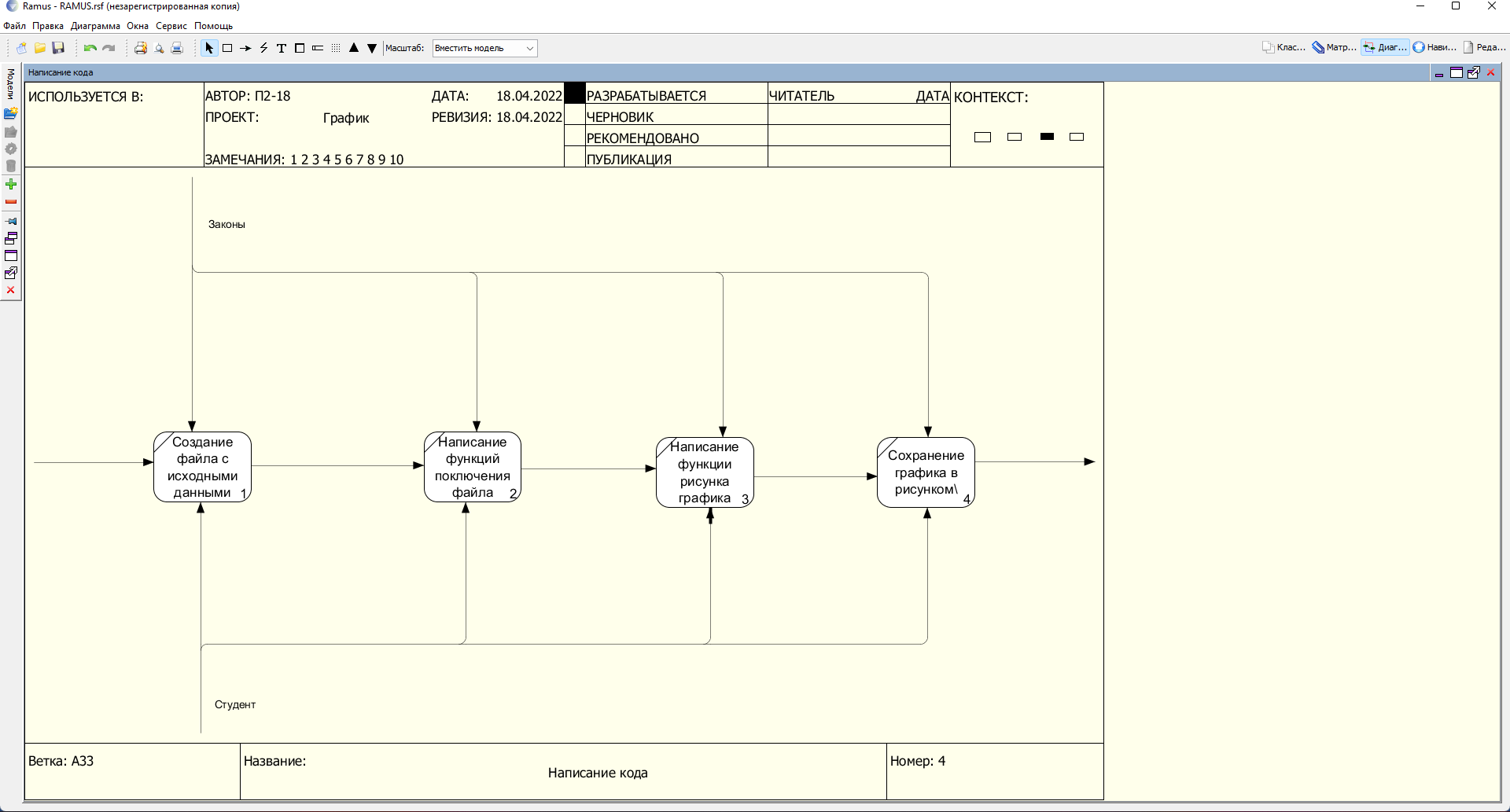
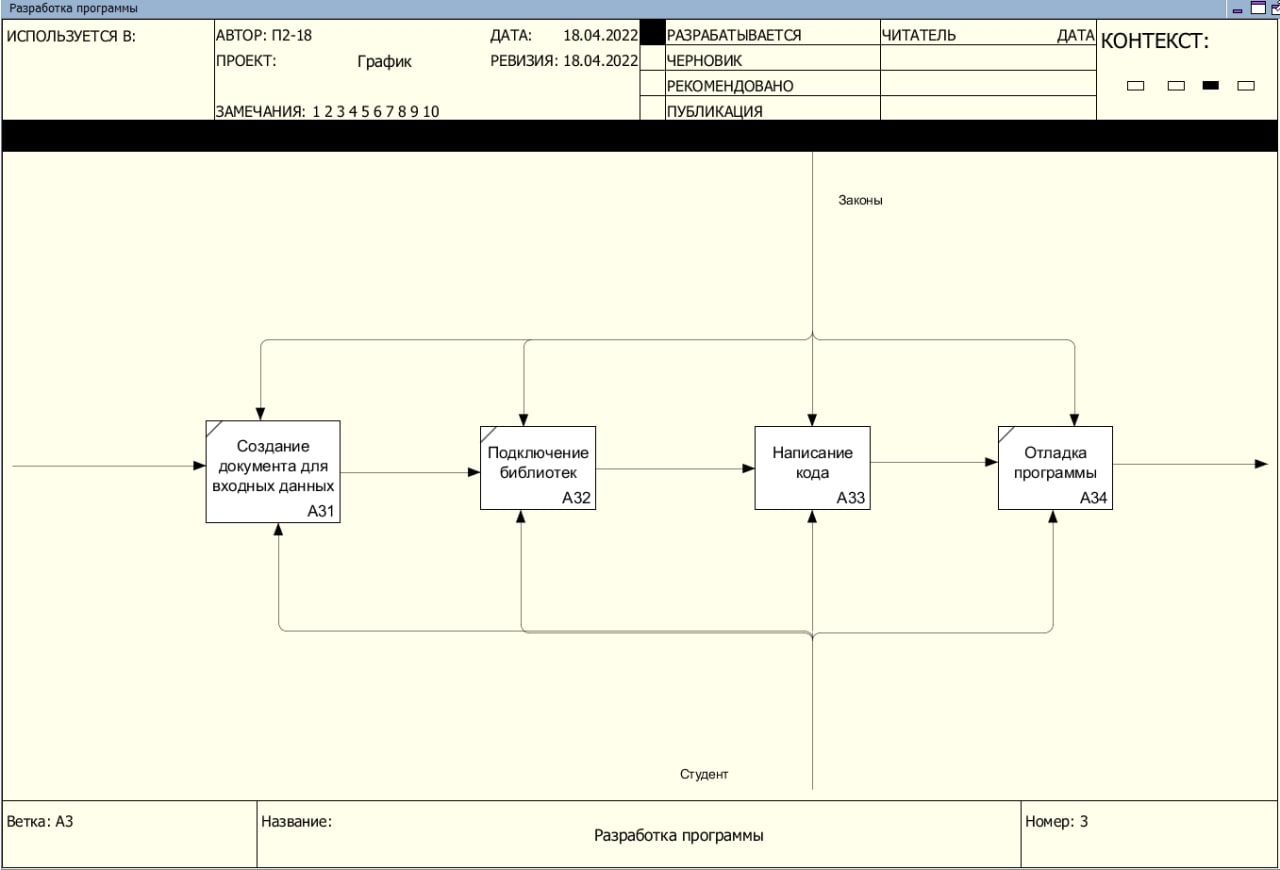


Рисунок 3. Пустой ввод данных







4.

**Экзаменационное задание «Выполнение функции»**

**Руководство оператора**

2022

**Аннотация**

Данный программный документ представляет собой руководство оператора Экзаменационного задания «Выполнение функции», предназначенной для вывода графика функций.

В разделе «Назначение программного комплекса» указаны назначение и функции, выполняемые программным комплексом.

В разделе «Условия выполнения программного комплекса» приведены условия, необходимые для работы программного комплекса (требования к техническим и программным средствам).

В разделе «Выполнение программного комплекса» приведены подробные описания выполнения функций программного комплекса.

В разделе «Сообщения оператору» перечислены виды сообщений об ошибках или неправильных действиях оператора, их [сообщений] описание и решения возникших проблем

**Оглавление**

[**1.** **Назначение программы** 4](#_Toc101204696)

[**1.1.** **Назначение программы** 4](#_Toc101204697)

[**1.2.** **Описание функций программы** 4](#_Toc101204698)

[**2.** **Условия выполнения программы** 5](#_Toc101204699)

[**2.1.** **Требование к техническим (аппаратным) средствам** 5](#_Toc101204700)

[**3.** **Выполнение программы** 6](#_Toc101204701)

[**3.1.** **Запуск программы** 6](#_Toc101204702)

[**4.** **Сообщение оператору** 7](#_Toc101204703)

[**4.1.** **Сообщения при ошибках** 7](#_Toc101204704)

# **Назначение программы**

## **Назначение программы**

Программное изделие (далее – Программа), показывает пользователю график, который рисует программа в осях OX, OY.

Функциональные возможности:

* Отрисовка графика

## **Описание функций программы**

В состав программы входят следующие составные части:

1. Редактирование значений аргументов функции
2. Получение информации о функции

# **Условия выполнения программы**

## **Требование к техническим (аппаратным) средствам**

Требования, предъявляемые к пользовательским ПЭВМ, соответствуют следующим границам:

1. Компьютер оператора, включающий в себя:
2. процессор x86 с тактовой частотой, не менее 1 ГГц;
3. оперативную память объемом, не менее 1 Гб;
4. видеокарту, монитор, мышь, клавиатура.

# **Выполнение программы**

## **Запуск программы**

Для запуска Экзаменационного задания «Выполнение функции» на компьютере Оператора необходимо открыть Интерпретатор Python кода и запустить из папки файл main.py.

После запуска программы она выдаст построенный график считывая вводные параметры из текстового документа coef3.txt, представленной на нижеследующем рисунке (Рисунок 1).

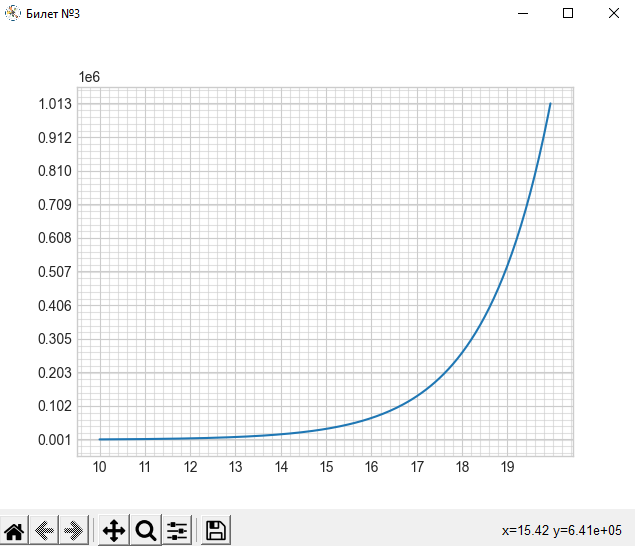


Рисунок 1. Вывод графика функции

# **Сообщение оператору**

Раздел содержит сообщения, выдаваемые Системой при совершении пользователем тех или иных действий.

## **Сообщения при ошибках**

При правильном вводе параметров в текстовом документе Программа сможет отрисовать график функции (Рисунок 2)

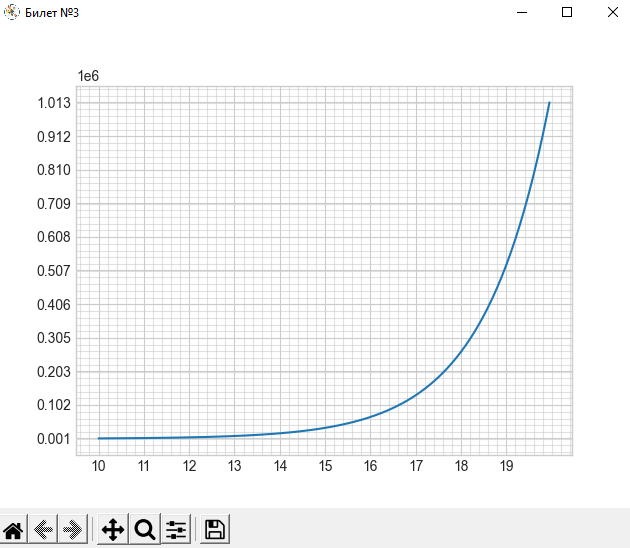


Рисунок 2. Правильный вывод графика

Если Оператор введет не верные значения в текстовый документ, то программа выдаст ошибку (Рисунок 3)

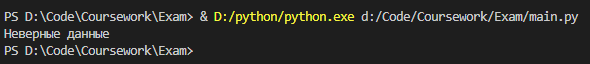


Рисунок 3. Введены неверные значения

При пустом вводе значений в текстовом документе, программа выведет сообщение, что Список пуст (Рисунок 4)

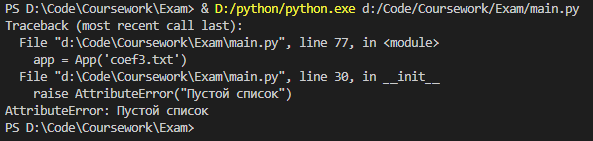


Рисунок 4. Выведен пустой список

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц) в докум. | № документ-та | Входящий № сопроводит. докум. и дата | Подпись | Дата |
| Изменения | измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |