1. Техническое задание.

**1 Введение**

**1.1 Наименование программы**

Наименование программы – «Воспроизведение формулы».

**1.2 Краткая характеристика области применения**

Система «Воспроизведение формулы» 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» Гусятинером Леонидом Борисовичем, именуемым в дальнейшем Заказчиком, Зайцев Никита Евгеньевич (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.

**Код:**

import matplotlib.pyplot as plt

from matplotlib.ticker import MultipleLocator

import numpy as np

import math

import re

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

fig = plt.figure(num='Билет №7')

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} \* cos(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()

if not re.search(r'^[\d /\*-+.]+$',''.join(line)):

raise AttributeError("Введен неверный символ")

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):

res = eval(self.function.replace('cos(x)', f'({str(math.cos(x))})'))

return res

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

if start <= 0:

raise AttributeError('Start must be greater than zero')

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

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('coef2.txt')

app.create\_points(2, 20, step=0.05)

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

**Описание применения средств отладки**

В этом разделе показано умение применять средства отладки.

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

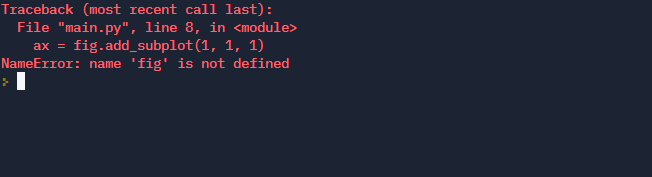


Рисунок 1. До применения средств отладки

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:

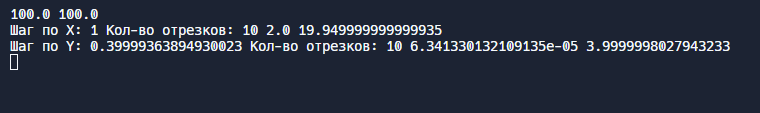


Рисунок 2. После применения средств отладки

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:

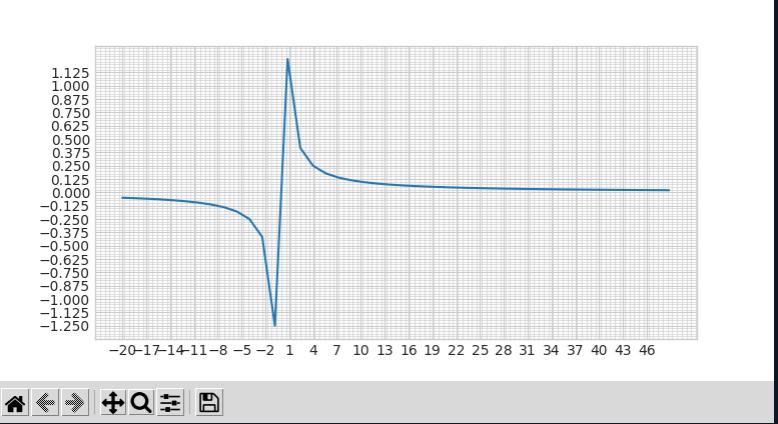


Рисунок 1. Корректный вывод оповещения.

Тест 2. Пустой файл(отсутствие данных).

Действия: Отсутствие данных.

Ожидаемый результат: Ошибка ” Пустой список”.

Результат теста:

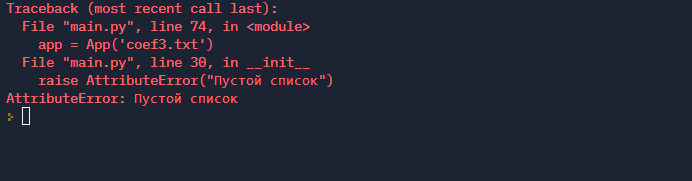


Рисунок 2. Корректный вывод оповещения.

Тест 3. Неверный формат

Действия: Неверный формат данных

Ожидаемый результат: Ошибка “Введён неверный символ”.

Результат теста:

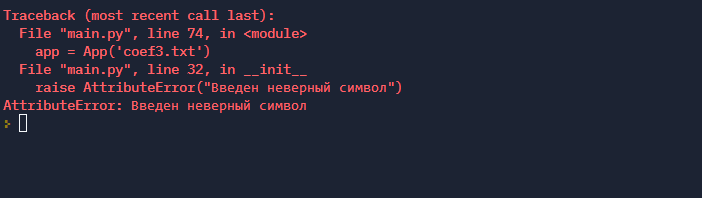


Рисунок 3. Корректный вывод оповещения.

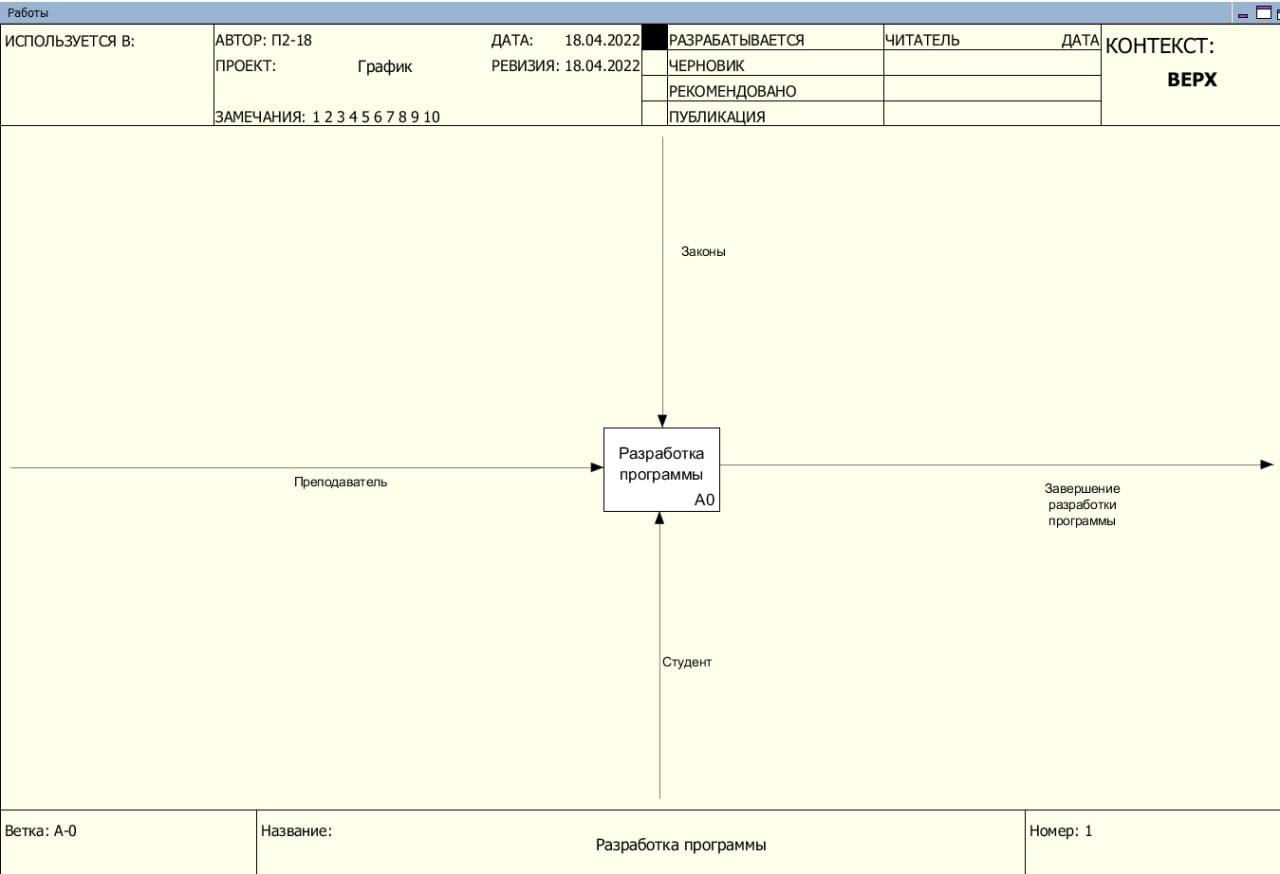


Рисунок 4. Фотография Ramus.

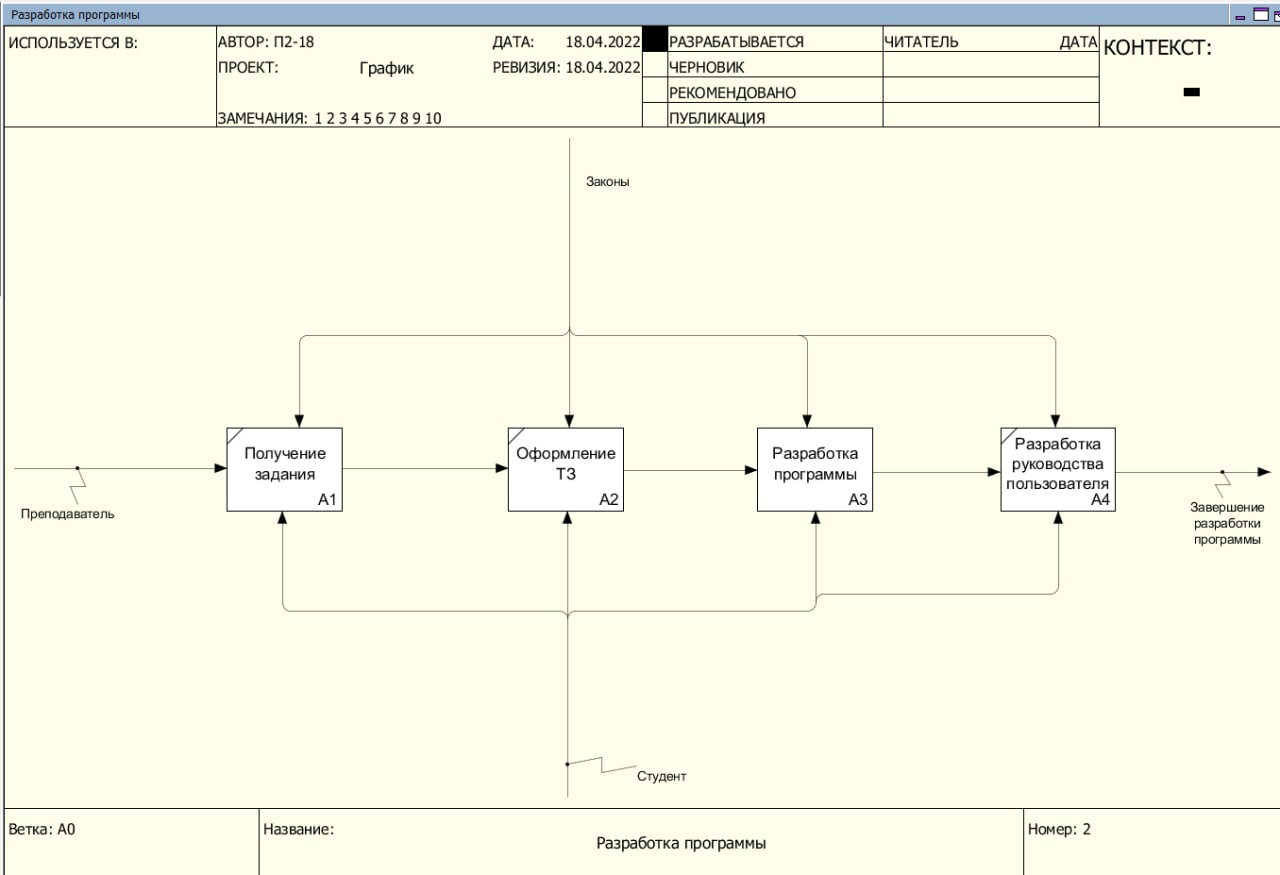


Рисунок 5. Фотография Ramus.

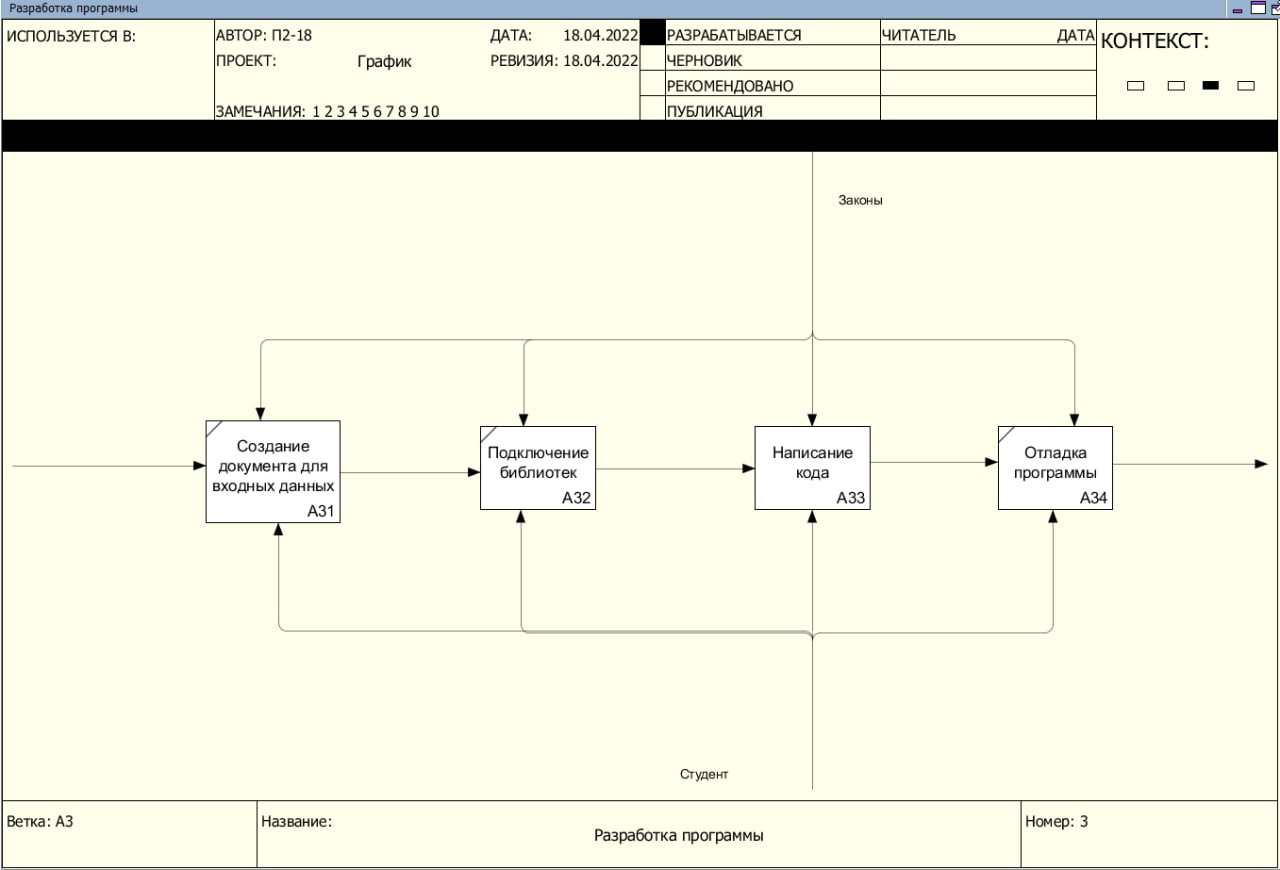


Рисунок 6. Фотография Ramus.

Информационная система  
**«Воспроизведение формулы»**

**Рабочая документация**

**Руководство оператора**

Королев 2022

# Аннотация

Настоящий документ представляет собой руководство оператора (далее Руководство) информационной системы выполнение функции.

Пользовательский интерфейс программы обеспечивает информационную поддержку деятельности оператора.

Руководство определяет порядок работы с программой для оператора.

Перед работой пользователя с программой рекомендуется внимательно ознакомиться с настоящим руководством.

Документ подготовлен в соответствии с РД 50-34.698-90 - в части структуры и содержания документов, и в соответствии с ГОСТ 34.201-89 - в части наименования и обозначения документов.

**Содержание**

[Аннотация 2](#_Toc100167133)

[1. Введение 4](#_Toc100167134)

[1.1 Область применения 4](#_Toc100167135)

[1.2 Краткое описание возможностей 4](#_Toc100167136)

[1.3 Уровень подготовки пользователя 4](#_Toc100167137)

[2. Назначение и условия применения 5](#_Toc100167138)

[2.1 Назначение 5](#_Toc100167139)

[2.2 Условия применения 5](#_Toc100167140)

[3. Подготовка к работе 6](#_Toc100167141)

[3.1 Порядок проверки работоспособности 6](#_Toc100167142)

[4. Описание операций 7](#_Toc100167143)

[4.1 Вход в программу под своей учетной записью 7](#_Toc100167144)

[4.2 Взаимодействие с программой 7](#_Toc100167145)

[5. Возможные ошибки и рекомендации по устранению 9](#_Toc100167146)

# Введение

## **Область применения**

Пользовательский интерфейс программы обеспечивает информационную поддержку деятельности пользователя при выполнении следующих операций:

- Получение информации о функции,

- Редактирование значений аргументов функции.

## **Краткое описание возможностей**

Данная программа позволяет осуществлять работу с графиком функции.

## **Уровень подготовки пользователя**

Для использования программы пользователю необходимо наличие базовых навыков работы с OS Windows.

# Назначение и условия применения

## **2.1** **Назначение**

Программа предназначена для построения графика функции.

Программа позволяет получать график функции и редактировать значения аргументов функции.

## **2.2** **Условия применения**

Программа может эксплуатироваться и выполнять заданные функции при соблюдении требований, предъявляемых к техническому, системному и прикладному программному обеспечению.

# Подготовка к работе

## **3.1** **Порядок проверки работоспособности**

Проверка работоспособности программы осуществляется путем выполнения операций, описанных в разделе 4 настоящего документа.

# Описание операций

В данном разделе приводится описание всех операций, существующих в программе.

## **4.1** **Ввод в текстовый документ аргументов функции**

Запускаем текстовый документ. В нем изменяем аргументы функции. Все значения вводятся через пробел. Первые два аргумента это значение “a” и “b” в функции. Вторые два аргумента это шаг сетки.

## **4.2** **Взаимодействие с программой**

После запуска программы оператору отображается график данной функции

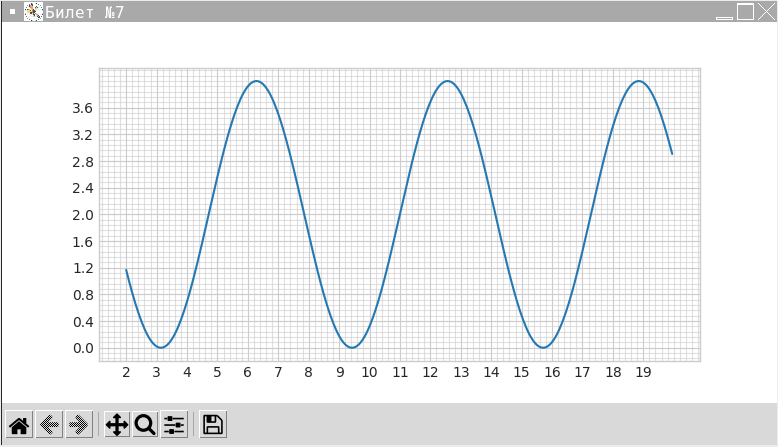


Рисунок 1. Рабочее окно приложения

# Возможные ошибки и рекомендации по устранению

|  |  |
| --- | --- |
| **Описание ошибки** | **Рекомендации по устранению** |
| Некорректный ввод аргументов в текстовый файл. | Ознакомиться с пунктом 4.1 |