**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Томский политехнический университет»

|  |  |
| --- | --- |
| Школа | Инженерная школа информационных технологий и робототехники |
| Обеспечивающее подразделение | Отделение информационных технологий |
| Направление подготовки / специальность | **09.03.01. Информатика и вычислительная техника** |
| Образовательная программа (направленность  (профиль)) | **Информатика и вычислительная техника** |

**ОТЧЕТ О ПРАКТИКЕ**

|  |  |
| --- | --- |
| Вид практики | Учебная |
| Тип практики | Практика по получению профессиональных умений и опыта профессиональной деятельности |
| Место практики | г. Томск, ТПУ, Инженерная школа информационных технологий и робототехники |

|  |  |
| --- | --- |
| Выполнил обучающийся | Сурков Дмитрий Андреевич |
| Группа | 8В01 |

(подпись обучающегося)

Руководитель практики ТПУ:

(степень, звание, должность) (Ф. И. О.)

Дата проверки 20 г.

Допустить / не допустить к защите

Подпись \_\_\_\_\_\_

Итоговая оценка по практике

(традиционная оценка, балл)

Томск 2021

**РЕФЕРАТ**

Цель практики: Освоение средства языка программирования Rython для создания научной графики. Также формирование определенного состава компетенций для подготовки к профессиональной, научно-исследовательской, педагогической деятельности. В ходе выполнения заданий студент получает навыки программирования на языке Python и знакомится с библиотеками mathplotlib, numpy, openCV.

Перечень основных выполненных работ:

Для начала были установлены все необходимое ПО. В частности, язык Python, библиотеки mathplotlib и numpy.

Затем, используя средства языка Phyton, выполнены следующие задания.

1. Создан многооконный рисунок размером (16 х 12) дюймов с 6 областями (2х3) для размещения диаграмм, графиков и иллюстраций.

Задание 1:

1.1. В 1-ю область помещена иллюстрация на выбор.

1.2. Во 2-ю область помещен стандартный график Python с заливкой

Задание 2:

2.1. В 3-ю область помещена столбчатая диаграмма с аннотацией

2.2. В 4-ю область помещен график функций с легендой

2.3. В 5-ю область помещена диаграмма рассеяния (разброса)

2.4. В 6-ю область помещен график фигур Лиссажу с заливкой

2. Были решены 4 задания на обработку изображений.

Использовались библиотеки mathplotlib, numpy, openCV. Для этого был произведен анализ литературы: документации и учебных пособий.

В ходе практической части, использовала теоретическую информацию, а также в заданиях использовал наблюдения (например, для установки удобных для восприятия границ графиков). В задании с построением фигуры Лиссажу экспериментируя подобрал оптимальный интервал параметра.

В результате работы получен многооконный рисунок с 6 областями для размещения диаграмм, графиков и иллюстраций. Также были получены программы для обработки изображений: программа, разбивающая изображение на четыре равные части. В каждой значение только одного канала R,G и B, а в четвертой градации серого цвета. Программа заменяющая все точки синего цвета на точки красного цвета, программа, инвертирующая изображение в градациях серого цвета в негатив и программа, изменяющая яркость изображения.

Были освоены средства языка программирования Rython для создания научной графики.

**ВВЕДЕНИЕ**

Все выполнялось на индивидуальном ноутбуке на праформе разработки PhyCharm. Основные сведения об использованных ЭВМ:

1. Интерпретатор Python.
2. Библиотеки mathplotlib, numpy, openCV.

Все программы были написаны на языке Python. Это высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памяти, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным — всё является объектами. Это язык более высокого уровня нежели Pascal, C++ и, C, что достигается, в основном, за счет встроенных высокоуровневых структур данных (списки, словари, тьюплы).

Использовалась библиотека mathplotlib для визуализации данных двумерной или трехмерной графики. С помощью этой библиотеки были построены графики, диаграммы разброса и столбчатые диаграммы. С помощью библиотеки numpy мы смогли использовать многомерные массивы и разные математические функции. Для работы с изображениями использовали openCV.

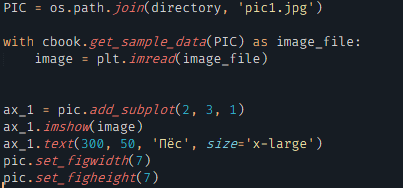
**ОСНОВНАЯ ЧАСТЬ ОТЧЕТА**

**Первая часть заданий**

**Задание 1.1 «Вставка изображения с надписью»:**

Осуществить вывод любого изображения с надписью.

**Код программы задания:**



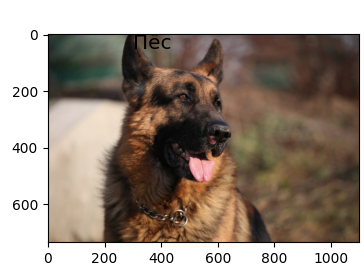
*Рисунок 1 – Код программы для задания 1.1*

С помощью get\_sample\_data загружено изображение, далее считали его в массив numpy с помощью функции imread модуля pyplot.

pic – это контейнер самого верхнего уровня, та область на которой все нарисовано. Таких областей может быть несколько, каждая из которых может содержать несколько контейнеров ax.

ax – это та область на которой чаще всего и отражаются графики (данные в виде графиков), а так же все вспомогательные атрибуты (линии сетки, метки, указатели и т.д.). Часто, установка этой области сопровождается с вызовом subplot, который и помещает ax на регулярную сетку.

**Результат работы программы:**

****

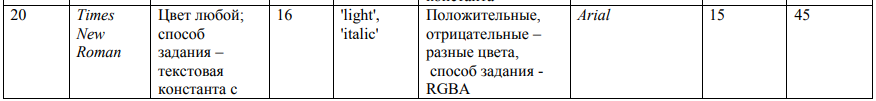
*Рисунок 2 – Вывод изображения*

**Задание 1.2 «Стандартный график Python с заливкой.»:**

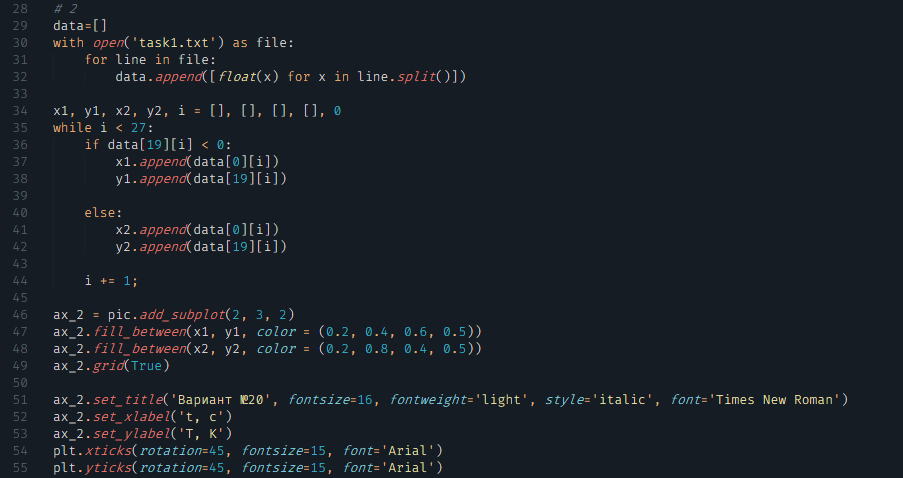
1) Прочитать информацию из файла.

2) Построить график с разной заливкой относительно координат.

3) Задания по 20 варианту:

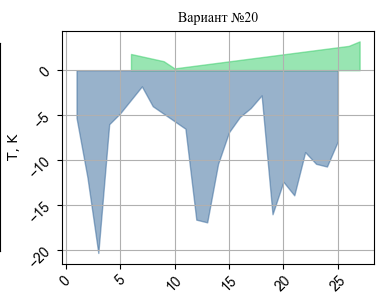


**Код программы задания:**



*Рисунок 3 – Код программы для задания 1.2*

**Результат работы программы:**



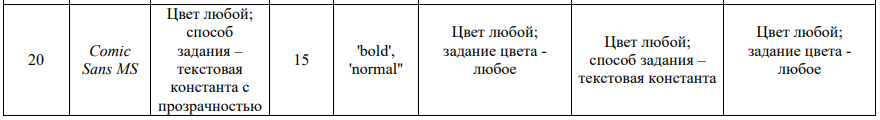
*Рисунок 4 – Полученный график*

**Выводы по 1-му заданию:**

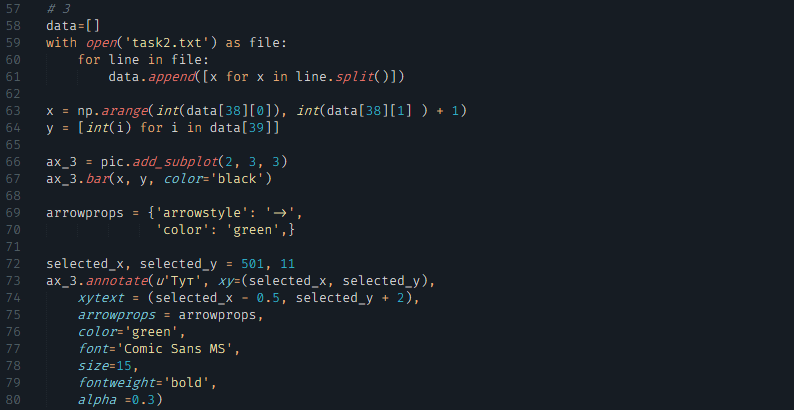
Были установлены все необходимые ПО. В частности, язык Python, библиотеки mathplotlib и numpy. Была написана программа выводящая изображение и надпись на нем. Далее с помощью построчного считывания с преобразованием была прочитана строка файла и построен график с разной заливкой относительно координат. Затем по номеру варианта были видоизменены некоторые параметры.

**Задание 2.1 «Столбчатая диаграмма с аннотацией»:**

Построить столбчатую диаграмму пользуясь данными из таблицы и текстового документа. Далее построить аннотацию к диаграмме, включающую стрелку с поясняющим текстом.



**Код программы задания:**

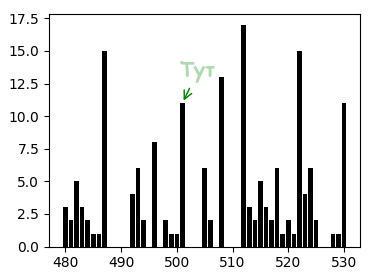


*Рисунок 5 – Код программы для задания 2.1*

С помощью функции with open был прочитан файл, откуда были взяты данные для построения столбчатой диаграммы. Далее с помощью метода bar была построена сама диаграмма, цвет указан методом HEX.

C помощью метода annotate указали текст аннотации, расположение точки, координаты для отображения графика, стиль, цвет указывающей стрелки. Затем поменяли шрифт и размер делений для лучшей видимости.

**Результат работы программы:**

****

*Рисунок 6 – Столбчатая диаграмма с аннотацией*

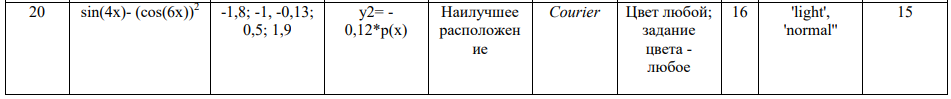
**Задание 2.2 «График с легендой»:**

1) Вычислить коэффициенты уравнения по корням полинома.

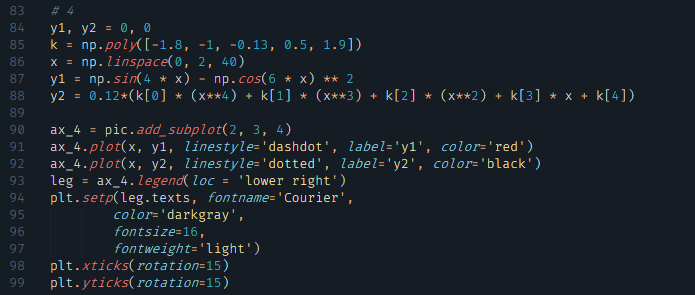
2) Построить многочлен.

3) Взять данные с таблицы.

3) Вычислить значения двух y1 и y2.



**Код программы задания:**

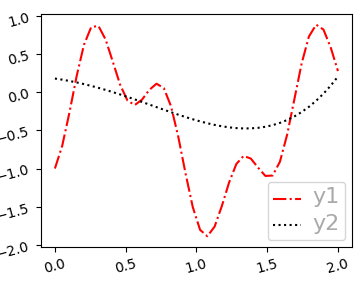
****

*Рисунок 7 – Код программы для задания 2.2*

С помощью метода plot нарисовали два различных графика. Функцию y2 умножили на многочлен, построенный по корням, приведенным в варианте. А многочлен мы построили, вычислив коэффициент уравнения по корням полинома, с чем нам помог метод poly из библиотеки numpy.

Далее воспользовавшись методом legend построили легенду, выбрали наилучшее для нее местоположение. В методе plot также указали стиль и цвет линий, назвали наши графики. С помощью функции setp поменяли шрифт, размер, жирность текста в легенде.

**Результат работы программы:**



*Рисунок 8 – Полученные графики*

**Задание 2.3 «Диаграммы рассеяния»:**

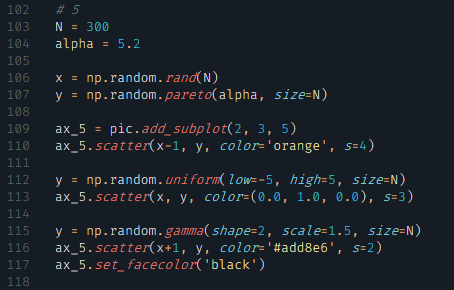
1) Использовать различные методы random

2) Использовать значения из таблицы.

2) Нарисовать 3 диаграммы разброса методом scatter.

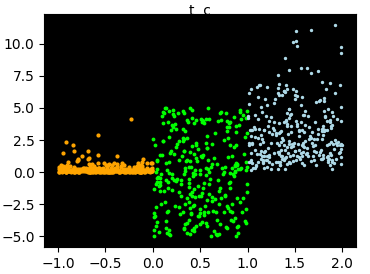


**Код программы задания:**



*Рисунок 9 – Код программы для задания 2.3*

**Результат работы программы:**



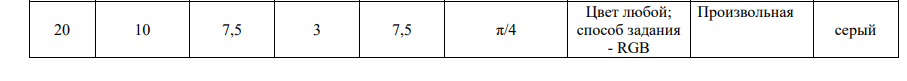
*Рисунок 9 – Полученная диаграмма рассеяния*

**Задание 2.4 «График фигур Лиссажу с заливкой»:**

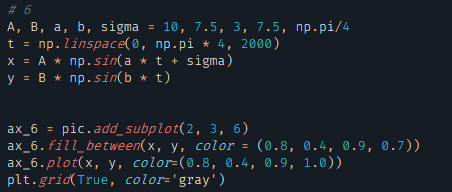
1) Считать данные из таблицы

2) По считанным данным построить фигуру Лиссажу по своему варианту.

2) Закрасить полученную фигуру.

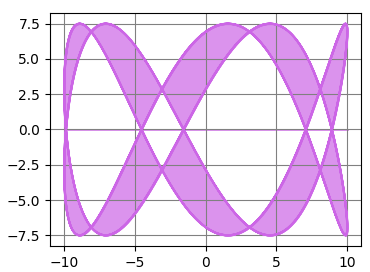
****

**Код программы задания:**



*Рисунок 10 – Код программы для задания 2.4*

**Результат работы программы:**

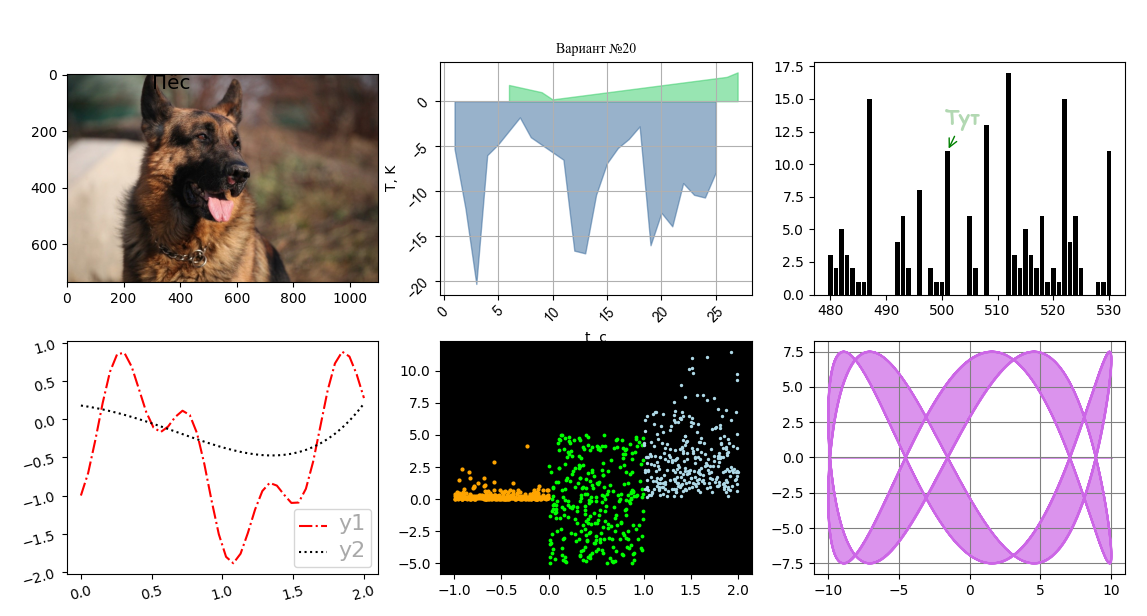


*Рисунок 11 – Фигура Лиссажу*

В конце создал многооконный рисунок с 6 областями и разместил полученные диаграммы, графики и иллюстрации.

С помощью plt.subplots() создаем 3 строки и 2 ряда, то есть 6 независимых друг от друга областей ax на одной pic.

Результат:



*Рисунок 13 - Многооконный рисунок*

**Выводы по 2-му заданию:**

Было повторено считывание данных с файла. Затем были построены столбчатые диаграммы с аннотацией, включающей стрелку и поясняющую надпись.

Далее с помощью метода poly вычислили коэффициенты уравнения по корням полинома. Построили график и поясняющую легенду с помощью legend.

Были изучены различные методы random. Изменены размер и цвет точек диаграммы рассеяния.

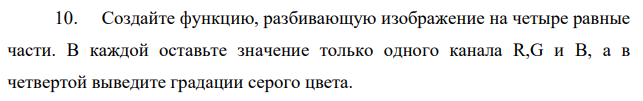
Разобрано как можно вывести функцию с параметром. Далее закрасили замкнутую фигуру Лиссажу.

Создан многооконный рисунок, в который были помещены все графики и диаграммы.

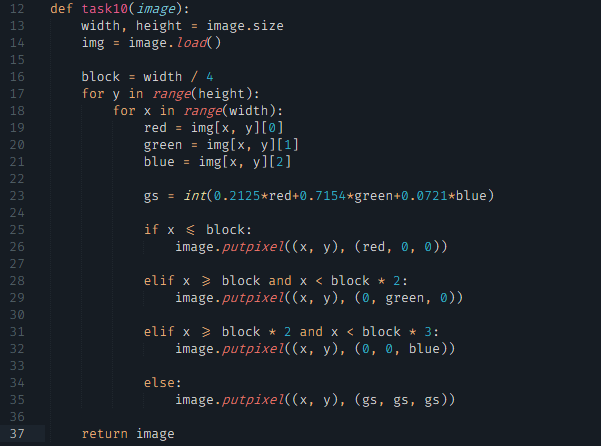
**Вторая часть заданий**

**Задание 3. (задачи номер 10, 11, 12, 13 по варианту № 20)**

**Задача номер 10:**

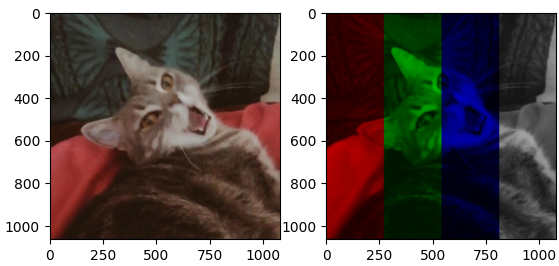


**Код программы задания:**



*Рисунок 14 – Код программы для задачи 10*

**Результат работы программы:**

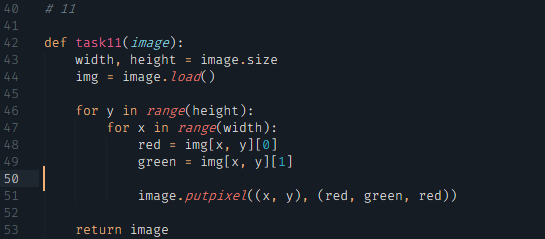
****

*Рисунок 16 – Результат работы функции*

**Задача номер 11:**

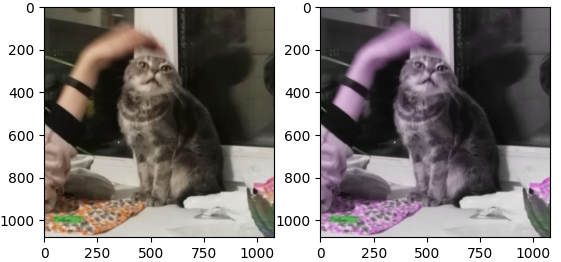


**Код программы задания:**

****

*Рисунок 17 – Код программы для задачи 11*

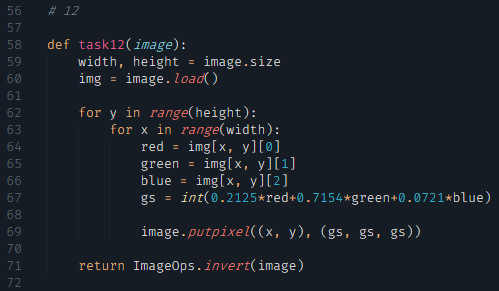
**Результат работы программы:**



**Задача номер 12:**

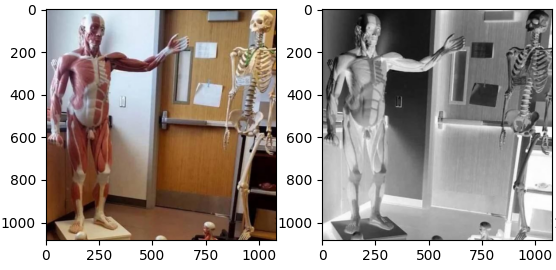


**Код программы задания:**



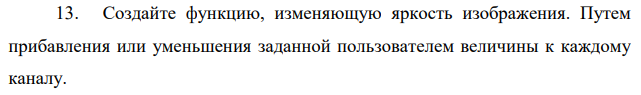
*Рисунок 21 – Код программы для задачи 12*

**Результат работы программы:**

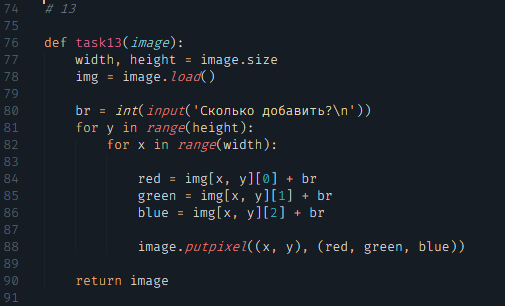


*Рисунок 22 – Оригинал и измененное изображение*

**Задача номер 13:**



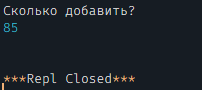
**Код программы задания:**

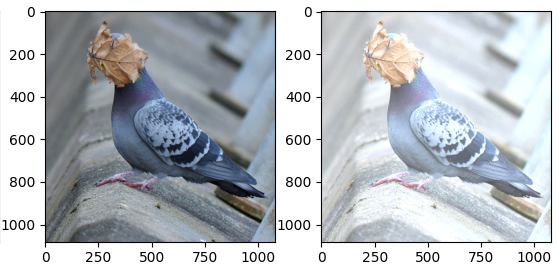


*Рисунок 23 – Код программы для задачи 13*

**Результат работы программы:**

Пользователь ввел значение 85:





*Рисунок 24 – Результат работы функции*

**Выводы по 3-му заданию:**

В ходе данного задания был изучен процесс анализа и работы с цифровым изображением, направленный на улучшение или изменения изображения, или извлечения информации для дальнейшего использования.

Были отображены изображения и выполнены основные операции. Были изучена Python-библиотека для работы с изображениями, а именно OpenCV .

Получены базовые знания для обработки изображений.

**ЗАКЛЮЧЕНИЕ**

**Выводы по результатам выполнения работ:**

Освоены средства языка программирования Python для создания научной графики, в частности, библиотека matplotlib. С помощью него были построены графики, столбчатые диаграммы, графики разброса.

Были изучены такие методы считывания данных из файла как: построчное считывание с преобразованием, преобразование при помощи функции map, преобразование с использованием регулярного выражения, с использованием CSV Reader, Numpy loadtxt, Numpy genfrontxt, Pandas read\_csv.

Также были изучены методы axes pyplot, закрашивание областей fill, fill-between и средства ввода, вывода, обработки, редактирования и анализа изображений.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Краткое руководство по Matplotlib <https://pyprog.pro/mpl/mpl_short_guide.html> .
2. Настройка графиков. Свойства класса Text в matplotlib <https://devpractice.ru/matplotlib-lesson-3-4-text-class-properties/> .
3. Типы графиков в matplotlib <https://pythonru.com/biblioteki/tipy-grafikov-v-matplotlib-plt3> .
4. Цвета HTML <https://colorscheme.ru/html-colors.html>
5. Самоучитель Python <https://stud.lms.tpu.ru/pluginfile/php/816733/mod_resource/content/2/Методическиеуказания.pdf>