МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON. РЕАЛИЗАЦИЯ МАСОНСКОГО ШИФРА**Пояснительная записка

Отчёт по дисциплине

«Учебная практика»

ТПЖА 090302.049 ПЗ

Разработал студент гр. ИТб–1302–01–00 \_\_\_\_\_\_\_\_\_\_\_\_/ Лазарев Д.Л./

(подпись)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_/ Родионов К.В./

(подпись)

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_»\_\_\_\_\_\_\_\_\_\_2020 г.

Киров 2020

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Утверждаю | | | | | | | | |
| Зав. каф. | | | | | САУ | | | |
|  | | | |  | наименование | | | |
|  | | | |  | Ланских Ю. В. | | | |
| подпись | | | | | Ф.И.О. | | | |
| « | 06 | » | июля | | | 20 | 20 | г. |

ЗАДАНИЕ

на практику

|  |  |
| --- | --- |
| по дисциплине | Учебная практика |
|  | полное название дисциплины |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студенту |  | , обучающемуся по образовательной программе | | |
| 09.03.02 – Информационные системы и технологии | | | | |
| полное название направления подготовки (специальности) | | | | |
| первый | | |  | очная |
| курс обучения | | |  | форма обучения |

|  |  |
| --- | --- |
| Индивидуальные задания, выполняемые в период практики: | 1. Выполнить обучающие задания, приведённые в методических указаниях, для ознакомления с методами работы с языком Python. |
| 2. Реализовать криптографический алгоритм (Масонский шифр) при помощи языка Python, не используя библиотечные функции, связанные непосредственно с шифрованием. Разработанная программа должна осуществлять шифрование и дешифрование текста в соответствии с вариантом. В приложении должна быть возможность ввода текста (вручную или загрузка из файла – на усмотрение разработчика), ключа (если этого требует алгоритм) и зашифрованного текста. | |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Исходные данные: | задание в соответствии с вариантом | | |
| 2. Основные разделы: |  | | |
| 1. Обучающие задания. Анализ предметной области. Определение цели и задач практики | | | |
| 1. Обзор математической составляющей шифра | | | |
| 1. Алгоритмы шифрования и дешифрования | | | |
| 3. График выполнения: | 6 – 8 июля ознакомление с python | | |
| 8 – 12 июля выполнение обучающих заданий | | | |
| 12 – 16 июля реализация шифрования и дешифрования | | | |
| Представить выполненную работу по практике на проверку не позднее: | |  | 16.07.20 |
|  | |  | Дата |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Руководитель работы |  |  |  | Родионов К.В. |  | 06.07.20 |
|  |  | Подпись руководителя |  | Ф.И.О. руководителя |  | Дата |
| Задание принял |  |  |  |  |  | 06.07.20 |
|  |  | Подпись обучающегося |  | Ф.И.О. обучающегося |  | Дата |

**Реферат**

Лазарев Д.Л. Программирование на языке Python. Реализация Масонского шифра: ТПЖА.090302.049 ПЗ: Учебная практика / ВятГУ, каф. САУ; рук. К.В. Родионов. – Киров, 2020. ПЗ 24 с., 9 рис., 3 источников, 3 прил.

ПЕРЕЧЕНЬ КЛЮЧЕВЫХ СЛОВ: Масонский шифр, k–means, k–nearest, Python, Sublime text 3, интерфейс, библиотека numpy.

Объект исследования и разработки – Масонский шифр.

Цель работы – написать программу для шифрования и дешифрования Масонским шифром.

Разработана программа шифрования и дешифрования Масонским шифром в соответствии с заданием и исходными данными, также проанализирована задача и описаны полученные результаты.

*Количество*

*листов*

*Примеч.*

*№ экз.*

*Наименование*

*Обозначение*

*Формат*

*№ строки*

Изм.

Лист

№ докум.

Подпись

Дата

Лист

1

*ТПЖА 090302.049 ПЗ*

Разраб.

*Лазарев Д.Л.*

Провер.

Родионов К.В.

Реценз.

Н. Контр.

Утверд.

*Программирование на языке Python. Реализация Масонского шифра*

Литер

Листов

1

*Кафедра САУ*

*группы ИТб–1302*

24

*Пояснительная записка*

*А4*

*ТПЖА.090302.049 ПЗ*

*4*

*Документация общая*

*Вновь заработанная*

*27*

*28*

*16*

*11*

*10*

*9*

*8*

*7*

*6*

*5*

*3*

*2*

*17*

*18*

*1*

*13*

*14*

*15*

*21*

*20*

*19*

*22*

*23*

*24*

*25*

*26*

*12*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Содержание   |  |  | | --- | --- | | Введение | 3 | | 1 Определение цели и задач практики | 4 | | 1.2 Среда разработки sublime text 3 | 4 | | 1.3 Обучающие задания | 5 | | 1.3.1 Нахождение k–means | 5 | | 1.3.2 Нахождение k–nearest | 6 | | 1.3.3 Выполнение математическая статистика | 6 | | 1.4 Вывод по разделу 1 | 7 | | 2 Масонский шифр | 8 | | 2.1 С чем работает программа | 8 | | 2.2 Вывод по разделу 2 | 9 | | 3 Алгоритмы шифрования и дешифрования | 10 | | 3.1 Шифрование | 10 | | 3.1.2 Вывод на экран | 10 | | 3.1.3 Результат работы программы шифрования | 11 | | 3.2 Дешифрование | 11 | | 3.2.2 Вывод на экран | 11 | | 3.2.3 Результаты работы программы дешифрования | 11 | | 3.3 Вывод по разделу 3 | 12 | | 4 Заключение | 13 | | Приложение А (обязательное) Листинг программного кода, реализация алгоритма шифрования | 14–18 | | Приложение Б (обязательное) Листинг программного кода, реализация алгоритма дешифрования | 19–23 | | Приложение В (справочное) Библиографический список | 24 | | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | |
|  |  |  |  |  |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| *Разраб.* | | *Лазарев* |  |  | *Программирование на языке Python. Реализация Масонского шифра* | *Литер* | | | *Лист* | *Листов* | |
| *Пров.* | | *Родионов* |  |  |  |  |  | *2* | *1* | |
| *Т.контр* | |  |  |  | *Кафедра САУ*  *группы ИТб–1302* | | | | | |
| *Н.контр* | |  |  |  |
| *Утв.* | |  |  |  |
| Введение  Криптография – наука о защите информации и данных. История криптографии насчитывает около 4 тысяч лет. Расширение человеческого общества привело к необходимости защиты информации, чтобы она оставалась тайной от всех, кроме определенных лиц(а).  Применение криптографии в настоящее время неотъемлемо от жизни человека – она используется в электронной коммерции, банкинге, телекоммуникации и электронном документообороте.  Python – [высокоуровневый язык программирования](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%BE%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D0%B5%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. [Синтаксис](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) ядра Python минималистичен. В то же время [стандартная библиотека](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_Python) включает большой объём полезных функций.  Python – активно развивающийся [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года. Язык не подвергался официальной стандартизации, роль стандарта де– факто выполняет [CPython](https://ru.wikipedia.org/wiki/CPython), разрабатываемый под контролем автора языка. В настоящий момент Python занимает третье место в [рейтинге TIOBE](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81_TIOBE) с показателем 10,2 %. Аналитики отмечают, что это самый высокий балл Python за все время его присутствия в рейтинге.  Целью данной практики является изучение шифрования с помощью Масонского шифра и его реализация на языке программирования Python. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *3* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 1 Определение цели и задач практики.  Целью работы является выполнение обучающих задний и разработка программы на основе полученных знаний для шифрования и дешифрования текста с помощью шифра Масона .  **1.2 Среда разработки sublime text 3**  Sublime Text 3 – проприетарный текстовый редактор. Поддерживает плагины на языке программирования Python. Разработчик позволяет бесплатно и без ограничений ознакомиться с продуктом, однако программа уведомляет о необходимости приобретения лицензии.    Рисунок 1 – среда разработки Sublime text 3 | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *4* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 1.3 Обучающие задания  Первое задние заключается в написании кода для нахождения k–means.  Во втором задании надо найти k–nearest.  Третье выполнить математическую статистику предоставленных условий.  1.3.1 Нахождение k–means  Алгоритм k средних (англ. k–means) – один из алгоритмов машинного обучения, решающий задачу кластеризации. Этот алгоритм является неиерархическим, итерационным методом кластеризации, он получил большую популярность благодаря своей простоте, наглядности реализации и достаточно высокому качеству работы. Результат работы программы представлен на рисунке 2.  В ней происходит расчет расстояния расстояний между точками матриц, которые пришли в данную функцию, и записываются в массив «distances». После, данный метод возвращает массив индексов с максимальным значением в строках данной матрицы. Далее, программа заходит в цикл, где условие выхода – полное совпадение массивов «prev\_iteration» и «curr\_iteration». В нем происходит расчет нового массива «curr\_iteration», после расчета центра, и сравнивается со старым. Данный метод «k– means», после выхода из цикла возвращает матрицу с центрами (2 кластерами) и отдает управление файлу «run.py». После, на экран пользователя выводятся 10 точек из матрицы «X» и 2 кластера из массива «kmeans».    Рисунок 2 – Результат работы программы k – means | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *5* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 1.3.2 Нахождение k–nearest  Метод ближайших соседей – метрический алгоритм для автоматической классификации объектов или регрессии. В случае использования метода для классификации объект присваивается тому классу, который является наиболее распространённым среди соседей данного элемента, классы которых уже известны.  Мы имеем изначально данные об исследованиях обезьян, которые мы заполняем матрицу Х, после мы пишем данные, которые хотим проверить и вызываем функцию kNN, где мы и ищем ближайших соседей и возвращаем самый близкий объект. Код представлен в приложении А, результат в приложении В рисунок 2.    Рисунок 3 – Результат работы программы k – nearest  1.3.3 Выполнение математическая статистика  Код, реализующий задания представлен в приложении Б. Результаты работы на рисунке 4.  Мы переписывали из методички некоторую часть кода и дополняли его, если того требовало задание.  Во 2 задании импортируется текстовый файл, содержащий матрицу.  В 3 задании импортируется данные из 1D и оцениваются параметры одномерной случайной величины, используя команды NumPy: np.max, np.min, np.median, np.mean, np.var, np.std.  В 4 и 5 заданиях выводятся графики одномерных случайных величин и их плотности распределения. На одном графике отображается случайная величина, уровень среднего значения и дисперсия и выводиться на графике автокорреляция заданной одномерной случайной величины.  В заданиях 6 и 7 импортируются из ND. Строится матрица корреляции для всех входных и выходных величин.    Рисунок 4 – Результат выполнения математической статистики | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *6* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 1.4 Вывод по разделу 1  Были рассмотрены обучающие задания, также представлена история создания Масонского шифра. Изучены задания и цели на практику и кратко рассказано о среде разработки, в которой работали. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *7* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 2 Масонский шифр  Масонский шифр – это геометрический шифр подстановки, в котором каждой букве алфавита ставится свое место в одной из четырех сеток. Примером ключа будет служить то, какое место в одной из четырех сеток мы определим каждой букве алфавита. Точное время происхождения шифра неизвестно, однако некоторые из найденных записей этой системы датируются 18 веком. Вариации этого шифра были использованы орденом розенкрейцеров и масонами, хотя последние использовали его так часто, что шифр стали называть шифром масонов. Они начали его использовать в начале 18 века, чтобы сохранить в тайне записи своей истории и обрядов, а также переписки лидеров своего движения. На надгробиях масонов можно встретить гравюры, на которых есть надписи, использующие данный шифр. На одном из самых ранних камней на кладбище в Церкви Троицы в Нью –Йорке, которая открылась в 1967 году, содержится надпись, использующая этот шифр, которая расшифровывается как "Помни смерть". Армия Джорджа Вашингтона вела документацию о похожей системе шифрования с наиболее случайным алфавитом. И во время гражданской войны в США данная система шифрования была использована заключенными, принадлежащими Союзу США в федеральных тюрьмах Конфедераций Штатов США. Пример масонского шифра на рисунке 5.    Рисунок 5 – таблица для шифрования масонским методом  2.1 С чем работает программа  Программа работает с интерфейсом. Каждой кнопке привязан свой соответствующий символ и при нажатии выводит его на экран, таким образом составляется закодированный текст. При шифровании, так и при дешифровании, нужно следовать таблице, алфавит может состоять только из букв. Пример таблицы представлен на рисунке 5. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *8* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 2.2 Вывод по разделу 2  Здесь была рассмотрена теоретическая составляющая Масонского шифрования, здесь есть примеры, на которые будут ссылаться в алгоритме шифрования и дешифрования. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *9* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 3 Алгоритмы шифрования и дешифрования  В Масонском шифре я выделил 4 этапа:   1. Функция загрузки изображения символа. 2. Функция вывода изображения. 3. Функция, присваивающая вывод определенного изображения, к определенной кнопке. 4. Функция, обрабатывающая нажатие кнопки.   3.1 Шифрование  Процесс шифрования начинается с преобразование текста в символы. Каждой кнопке привязан конкретное изображение. Функция загрузки изображения представлена на рисунке 6. При нажатии на кнопку экранной клавиатуры, вызывается функция вывода изображения, которая выводит символ, соответствующий нажатой букве, на экран.    Рисунок 6 – функция загрузки изображения    Рисунок 6.1 – привязка загруженных изображений к конкретным переменным  3.1.2 Вывод на экран  Вывод на экран при шифровании осуществляется вызовом соответствующей функции (рисунок 7).    Рисунок 7 – функция вывода изображения символа | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *10* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 3.1.3 Результат работы программы шифрования  Результат работы программы представлен на рисунке 8. Код программы шифрования представлен в приложении А    Рисунок 8 – результат работы программы шифрования  3.2 Дешифрование  Дешифрование происходит аналогичным способом, как и шифрование. При нажатии клавиши с определенным символом на экранной клавиатуре, выводится на экран соответствующая буква.  3.2.2 Вывод на экран  Вывод на экран при дешифровании осуществляется также, как и при шифровании, что представлено в пункте 3.1.2.  3.2.3 Результаты работы программы дешифрования  Результат работы программы представлен на рисунке 9. Код программы шифрования представлен в приложении Б    Рисунок 9 – результат работы программы дешифрования | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *11* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 3.3 Вывод по разделу 3  В главе третьей был показан алгоритм, по которому был написан код программы шифрования Масонским шифром, по которому мы можем понять, что основной упор в данном шифровании делается на интерфейс и функции. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *12* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| **4 Заключение**  В процессе выполнения практики был реализован Масонский шифр, методы k –means и k –nearest и частично изучен язык программирования «Python» в среде разработки «Sublime text 3», также его библиотека NumPy, которую мы активно применяли, и интерфейс, точнее библиотека tkinter. | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *13* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение А  (обязательное)  Листинг программного кода, реализация алгоритма шифрования  # Шифрование  from tkinter import \*  from PIL import ImageTk, Image  from math import floor  window = Tk()  window.title("Encoder") # Название  window.geometry('855x600') # размеры  canvas = Canvas(window, width=850, height=300, bg="white")  canvas.pack()    # Функция загрузки изображения  def load\_image(name):  img = Image.open(name)  img.thumbnail((200, 200), Image.ANTIALIAS)  return ImageTk.PhotoImage(img)  # Функция смещения символа  x = 0# Начальная позиция по горизонатли конечная 825, всего 33 символа по горизонтали  y = 0# Начальная позиция по вертикали конечная 275, всего 11 строк  count = 0  # Функция вывода изображения  def set\_image(image):  global x, y, count  x = (count \* 50) % 850  y = floor((count \* 50) / 850) \* 50  canvas.create\_image(x+25,y+25,image=image)  count += 1  # МАСОНСКИЙ АЛФАВИТ  imageSpace = load\_image ("Space.png")  imageA = load\_image("A.png")  imageB = load\_image("B.png")  imageC = load\_image("C.png")  imageD = load\_image("D.png")  imageE = load\_image("E.png")  imageF = load\_image("F.png")  imageG = load\_image("G.png")  imageH = load\_image("H.png")  imageI = load\_image("I.png")  imageJ = load\_image("J.png")  imageK = load\_image("K.png")  imageL = load\_image("L.png")  imageM = load\_image("M.png")  imageN = load\_image("N.png")  imageO = load\_image("O.png")  imageP = load\_image("P.png")  imageQ = load\_image("Q.png")  imageR = load\_image("R.png")  imageS = load\_image("S.png") | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *14* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение А  (обязательное)  Листинг программного кода, реализация алгоритма шифрования (продолжение)  imageT = load\_image("T.png")  imageU = load\_image("U.png")  imageV = load\_image("V.png")  imageW = load\_image("W.png")  imageX = load\_image("X.png")  imageY = load\_image("Y.png")  imageZ = load\_image("Z.png")  # Функция кнопки для вывода изображения  def ButtonA():  set\_image(imageA)  def ButtonB():  set\_image(imageB)  def ButtonC():  set\_image(imageC)  def ButtonD():  set\_image(imageD)  def ButtonE():  set\_image(imageE)  def ButtonF():  set\_image(imageF)  def ButtonG():  set\_image(imageG)  def ButtonH():  set\_image(imageH)  def ButtonI():  set\_image(imageI)  def ButtonJ():  set\_image(imageJ)  def ButtonK():  set\_image(imageK)  def ButtonL():  set\_image(imageL)  def ButtonM():  set\_image(imageM)  def ButtonN():  set\_image(imageN)  def ButtonO():  set\_image(imageO)  def ButtonP():  set\_image(imageP) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *15* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение А  (обязательное)  Листинг программного кода, реализация алгоритма шифрования (продолжение)  def ButtonQ():  set\_image(imageQ)  def ButtonR():  set\_image(imageR)  def ButtonS():  set\_image(imageS)  def ButtonT():  set\_image(imageT)  def ButtonU():  set\_image(imageU)  def ButtonV():  set\_image(imageV).  def ButtonW():  set\_image(imageW)  def ButtonX():  set\_image(imageX)  def ButtonY():  set\_image(imageY)  def ButtonZ():  set\_image(imageZ)  def ButtonSPACE():  set\_image(imageSpace)  # 1й ряд букв  Q = Button(window, text="Q", width=10,height=2,bg="white",fg="blue", command = ButtonQ)  Q.place(x=30, y=430)  W = Button(window, text="W", width=10,height=2,bg="white",fg="blue", command = ButtonW)  W.place(x=110, y=430)  E = Button(window, text="E", width=10,height=2,bg="white",fg="blue", command = ButtonE)  E.place(x=190, y=430)  R = Button(window, text="R", width=10,height=2,bg="white",fg="blue", command = ButtonR)  R.place(x=270, y=430)  T = Button(window, text="T", width=10,height=2,bg="white",fg="blue", command = ButtonT)  T.place(x=350, y=430)  Y = Button(window, text="Y", width=10,height=2,bg="white",fg="blue", command = ButtonY)  Y.place(x=430, y=430)  U = Button(window, text="U", width=10,height=2,bg="white",fg="blue", command = ButtonU)  U.place(x=510, y=430) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *16* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение А  (обязательное)  Листинг программного кода, реализация алгоритма шифрования (продолжение)  I = Button(window, text="I", width=10,height=2,bg="white",fg="blue", command = ButtonI)  I.place(x=590, y=430)  O = Button(window, text="O", width=10,height=2,bg="white",fg="blue", command = ButtonO)  O.place(x=670, y=430)  P = Button(window, text="P", width=10,height=2,bg="white",fg="blue", command = ButtonP)  P.place(x=750, y=430)  # Второй ряд (9 букв)  A = Button(window, text="A", width=10,height=2,bg="white",fg="blue", command = ButtonA)  A.place(x=55, y=470)  S = Button(window, text="S", width=10,height=2,bg="white",fg="blue", command = ButtonS)  S.place(x=135, y=470)  D = Button(window, text="D", width=10,height=2,bg="white",fg="blue", command = ButtonD)  D.place(x=215, y=470)  F = Button(window, text="F", width=10,height=2,bg="white",fg="blue", command = ButtonF)  F.place(x=295, y=470)  G = Button(window, text="G", width=10,height=2,bg="white",fg="blue", command = ButtonG)  G.place(x=375, y=470)  H = Button(window, text="H", width=10,height=2,bg="white",fg="blue", command = ButtonH)  H.place(x=455, y=470)  J = Button(window, text="J", width=10,height=2,bg="white",fg="blue", command = ButtonJ)  J.place(x=535, y=470)  K = Button(window, text="K", width=10,height=2,bg="white",fg="blue", command = ButtonK)  K.place(x=615, y=470)  L = Button(window, text="L", width=10,height=2,bg="white",fg="blue", command = ButtonL)  L.place(x=695, y=470)  # Третий ряд (7 букв)  Z = Button(window, text="Z", width=10,height=2,bg="white",fg="blue", command = ButtonZ)  Z.place(x=100, y=510)  X = Button(window, text="X", width=10,height=2,bg="white",fg="blue", command = ButtonX)  X.place(x=180, y=510)  C = Button(window, text="C", width=10,height=2,bg="white",fg="blue", command = ButtonC)  C.place(x=260, y=510)  V = Button(window, text="V", width=10,height=2,bg="white",fg="blue", command = ButtonV)  V.place(x=340, y=510)  B = Button(window, text="B", width=10,height=2,bg="white",fg="blue", command = ButtonB)  B.place(x=420, y=510)  N = Button(window, text="N", width=10,height=2,bg="white",fg="blue", command = ButtonN)  N.place(x=500, y=510) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *17* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение А  (обязательное)  Листинг программного кода, реализация алгоритма шифрования (продолжение)  M = Button(window, text="M", width=10,height=2,bg="white",fg="blue", command = ButtonM)  M.place(x=580, y=510)  # Пробел  SPACE = Button(window, text="SPACE", width=30,height=2,bg="white",fg="blue", command = ButtonSPACE)  SPACE.place(x=300, y=550)  # Вывод эскиза  window.mainloop() | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *18* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение Б  (обязательное)  Листинг программного кода, реализация алгоритма дешифрования  # Дешифрование  from tkinter import \*  from tkinter import Tk, Button, PhotoImage  from PIL import ImageTk, Image  from math import floor  window = Tk()  window.title("Decoder") # Название  window.geometry('860x610') # размеры  canvas = Canvas(window, width=850, height=300, bg="white")  canvas.pack()  # Функция загрузки изображения  def load\_image(name):  img = Image.open(name)  img.thumbnail((200, 200), Image.ANTIALIAS)  return ImageTk.PhotoImage(img)  # Функция смещения символа  x = 0# Начальная позиция по горизонатли конечная 825, всего 33 символа по горизонтали  y = 0# Начальная позиция по вертикали конечная 275, всего 11 строк  count = 0  # Функция вывода изображения  def set\_image(image):  global x, y, count  x = (count \* 50) % 850  y = floor((count \* 50) / 850) \* 50  canvas.create\_image(x+25,y+25,image=image)  count += 1  # Английский алфавит  imageSpace = load\_image ("Space.png")  imageAA = load\_image("AA.png")  imageBB = load\_image("BB.png")  imageCC = load\_image("CC.png")  imageDD = load\_image("DD.png")  imageEE = load\_image("EE.png")  imageFF = load\_image("FF.png")  imageGG = load\_image("GG.png")  imageHH = load\_image("HH.png")  imageII = load\_image("II.png")  imageJJ = load\_image("JJ.png")  imageKK = load\_image("KK.png")  imageLL = load\_image("LL.png")  imageMM = load\_image("MM.png")  imageNN = load\_image("NN.png")  imageOO = load\_image("OO.png")  imagePP = load\_image("PP.png")  imageQQ = load\_image("QQ.png")  imageRR = load\_image("RR.png") | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *19* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение Б  (обязательное)  Листинг программного кода, реализация алгоритма дешифрования (продолжение)  # МАСОНСКИЙ АЛФАВИТ  imageSpace = load\_image ("Space.png")  imageA = load\_image("A.png")  imageB = load\_image("B.png")  imageC = load\_image("C.png")  imageD = load\_image("D.png")  imageE = load\_image("E.png")  imageF = load\_image("F.png")  imageG = load\_image("G.png")  imageH = load\_image("H.png")  imageI = load\_image("I.png")  imageJ = load\_image("J.png")  imageK = load\_image("K.png")  imageL = load\_image("L.png")  imageM = load\_image("M.png")  imageN = load\_image("N.png")  imageO = load\_image("O.png")  imageP = load\_image("P.png")  imageQ = load\_image("Q.png")  imageR = load\_image("R.png")  imageS = load\_image("S.png")  imageT = load\_image("T.png")  imageU = load\_image("U.png")  imageV = load\_image("V.png")  imageW = load\_image("W.png")  imageX = load\_image("X.png")  imageY = load\_image("Y.png")  imageZ = load\_image("Z.png")  # Функция кнопки для вывода изображения  def ButtonA():  set\_image(imageAA)  def ButtonB():  set\_image(imageBB)  def ButtonC():  set\_image(imageCC)  def ButtonD():  set\_image(imageDD)  def ButtonE():  set\_image(imageEE)  def ButtonF():  set\_image(imageFF)  def ButtonG():  set\_image(imageGG)  def ButtonH():  set\_image(imageHH) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *20* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение Б  (обязательное)  Листинг программного кода, реализация алгоритма дешифрования (продолжение)  def ButtonI():  set\_image(imageII)  def ButtonJ():  set\_image(imageJJ)  def ButtonK():  set\_image(imageKK)  def ButtonL():  set\_image(imageLL)  def ButtonM():  set\_image(imageMM)  def ButtonN():  set\_image(imageNN)  def ButtonO():  set\_image(imageOO)  def ButtonP():  set\_image(imagePP)  def ButtonQ():  set\_image(imageQQ)  def ButtonR():  set\_image(imageRR)  def ButtonS():  set\_image(imageSS)  def ButtonT():  set\_image(imageTT)  def ButtonU():  set\_image(imageUU)  def ButtonV():  set\_image(imageVV)  def ButtonW():  set\_image(imageWW)  def ButtonX():  set\_image(imageXX)  def ButtonY():  set\_image(imageYY)  def ButtonZ():  set\_image(imageZZ)  def ButtonSPACE():  set\_image(imageSpace) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *21* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение Б  (обязательное)  Листинг программного кода, реализация алгоритма дешифрования (продолжение)  # 1й ряд букв  Q = Button(window, image = imageQ,bg="white", command = ButtonQ)  Q.place(x=30, y=400)  W = Button(window, image = imageW,bg="white", command = ButtonW)  W.place(x=110, y=400)  E = Button(window, image = imageE,bg="white", command = ButtonE)  E.place(x=190, y=400)  R = Button(window, image = imageR,bg="white", command = ButtonR)  R.place(x=270, y=400)  T = Button(window, image = imageT,bg="white", command = ButtonT)  T.place(x=350, y=400)  Y = Button(window, image = imageY,bg="white", command = ButtonY)  Y.place(x=430, y=400)  U = Button(window, image = imageU,bg="white", command = ButtonU)  U.place(x=510, y=400)  I = Button(window, image = imageI, command = ButtonI)  I.place(x=590, y=400)  O = Button(window, image = imageO, command = ButtonO)  O.place(x=670, y=400)  P = Button(window, image = imageP, command = ButtonP)  P.place(x=750, y=400)  # Второй ряд (9 букв)  A = Button(window, image = imageA, command = ButtonA)  A.place(x=55, y=450)  S = Button(window, image = imageS,fg="blue", command = ButtonS)  S.place(x=135, y=450)  D = Button(window, image = imageD, command = ButtonD)  D.place(x=215, y=450)  F = Button(window, image = imageF, command = ButtonF)  F.place(x=295, y=450)  G = Button(window, image = imageG, command = ButtonG)  G.place(x=375, y=450)  H = Button(window, image = imageH, command = ButtonH)  H.place(x=455, y=450)  J = Button(window, image = imageJ, command = ButtonJ)  J.place(x=535, y=450)  K = Button(window, image = imageK, command = ButtonK)  K.place(x=615, y=450) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *22* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| Приложение Б  (обязательное)  Листинг программного кода, реализация алгоритма дешифрования (продолжение)  L = Button(window, image = imageL, command = ButtonL)  L.place(x=695, y=450)  # Третий ряд (7 букв)  Z = Button(window, image = imageZ, command = ButtonZ)  Z.place(x=100, y=500)  X = Button(window, image = imageX, command = ButtonX)  X.place(x=180, y=500)  C = Button(window, image = imageC, command = ButtonC)  C.place(x=260, y=500)  V = Button(window, image = imageV, command = ButtonV)  V.place(x=340, y=500)  B = Button(window, image = imageB, command = ButtonB)  B.place(x=420, y=500)  N = Button(window, image = imageN, command = ButtonN)  N.place(x=500, y=500)  M = Button(window, image = imageM, command = ButtonM)  M.place(x=580, y=500)  # Пробел  SPACE = Button(window, image = imageSpace, command = ButtonSPACE)  SPACE.place(x=340, y=550)  # Вывод эскиза  window.mainloop() | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *23* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| **Приложение В**  (справочное)  **Библиографический список**   1. Шифр масонов [электронный ресурс] // Quest Hint 2.0 URL: [https://questhint.ru/shifr –masonov/](https://questhint.ru/shifr-masonov/) 2. Pigpen [электронный ресурс] // Википедия URL: <https://ru.wikipedia.org/wiki/Pigpen> 3. Масонский алфавит (код, шифр и т.д.) [электронный ресурс] [// Масонство России URL: https://cutt.ly/AakgvYd](%20//%20Масонство%20России%20URL:%20https://cutt.ly/AakgvYd) | | | | | | | | | | | |
|  |  |  |  |  | *ТПЖА 090302.049 ПЗ* | | | | | | *Лист* |
|  |  |  |  |  | *24* |
| *Изм* | *Лист* | *№ докум.* | *Подпись* | *Дата* |