Министерство образования и молодежной политики

Свердловской области

Государственное автономное профессиональное образовательное

учреждение Свердловской области «Ирбитский политехникум»

КУРСОВАЯ РАБОТА

по ПМ.01 Разработка программных модулей программного обеспечения для

компьютерных систем

МДК 01.01. Системное программирование

Тема:Термометрия, оптическая передача данных с термометра на языке Python

Выполнил:

студент группы ПКС-306

специальность 09.02.03

Программирование

в компьютерных системах

Останин Иван Васильевич

Проверил**:** преподаватель

Вишнякова Н.В.

Ирбит 2022СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc94527326)

[1. ОСОБЕННОСТИ ЯЗЫКА ПРОГРАММИРОВАНИЯ И РАЗРАБОТКА МОДЕЛИ ПРОГРАММЫ. 4](#_Toc94527327)

[1.1. Особенности языка программирования Python. 4](#_Toc94527328)

[1.2. Используемые библиотеки и инструменты. 4](#_Toc94527329)

[1.3. Принципы формирования изображения на термометре. 5](#_Toc94527330)

[1.4. Интерфейс программы. 5](#_Toc94527331)

[2. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ. 9](#_Toc94527332)

[2.1. Разработка структуры программы. 9](#_Toc94527333)

[2.1.1 Общая структура программы. 9](#_Toc94527334)

[2.1.2 Структура класса VideoScaner. 9](#_Toc94527335)

[2.2. Проблемы, возникшие во время разработки. 11](#_Toc94527336)

[2.3. Тестирование программы. 11](#_Toc94527337)

[ЗАКЛЮЧЕНИЕ 11](#_Toc94527338)

[СПИСОК ЛИТЕРАТУРЫ 12](#_Toc94527339)

# ВВЕДЕНИЕ

На сегодняшний день язык программирования Python является одним из самых востребованных в мире. Область его применения охватывает множество сфер деятельности человека: обработка данных, искусственный интеллект, разработка приложений с интерфейсом, работа с видео и многое другое.

Также данный язык программирования обладает огромной базой пользовательских библиотек и расширений, созданных сообществом и находящихся в открытом доступе. Это расширяет возможности языка до невообразимых масштабов. Работа с сайтами, различные боты для социальных сетей, построение сложнейших приложений и кроссплатформенная разработка. Стоит отметить и то, что он обладает крайне простым синтаксисом и очень прост в освоении.

Основываясь на вышеперечисленное, было принято решение использовать Python для разработки инструмента оптической передачи данных с термометра.

Данный инструмент может пригодится в проведении различных химический и физических экспериментов для протоколирования температуры на определённом промежутке времени.

Цель курсовой работы – создание программного средства для оптической передачи данных с термометра на языке Python.

Задачи:

1. подобрать и систематизировать литературу по теме;
2. изучить принципы формирования изображения термометра (модель);
3. рассмотреть особенности разработки программного продукта на языке Python;
4. разработать интерфейс взаимодействия пользователя с программой;
5. разработать алгоритм, реализовать его на языке программирования Python и провести тестирование и отладку программного средства;
6. оформить документацию на программный продукт.

* Объект исследования: Цифровой термометр.
* Предмет исследования: Язык программирования Python.

Перейдём к рассмотрению языка программирования Python.

# ОСОБЕННОСТИ ЯЗЫКА ПРОГРАММИРОВАНИЯ И РАЗРАБОТКА МОДЕЛИ ПРОГРАММЫ.

## 1.1. Особенности языка программирования Python.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества. Язык является полностью объектно-ориентированным в том плане, что всё является объектами.

## 1.2. Используемые библиотеки и инструменты.

PyCharm

IDE PyCharm Community Edition 2020.2 – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

Git

Git – распределённая система управления версиями. Она позволяет хранить всю историю изменений в проекте и удобно синхронизировать их между рабочими станциями.

OpenCV

OpenCV – библиотека для языка Python, предназначенная для работы с фото видео информацией и содержит огромное множество механизмов её обработки, включая поддержку библиотек компьютерного зрения и искусственного интеллекта.

В данной работе будут использованы только базовые механизмы этой библиотеки.

NumPy

NumPy – библиотека с открытым исходным кодом для языка Python для работы с многомерными массивами и высокоуровневыми математическими функциям, предназначенных для работы с ними.

Данная библиотека предоставляет функционал для обрезки, масштабирования и вращения изображений библиотеки OpenCV, которые и являются многомерными массивами.

MatplotLib

MatplotLib - библиотека на языке программирования Python для визуализации данных двумерной графикой, использовалась по прямому назначению.

ConfigParser

ConfigParser – встроенная библиотека python, дающая удобный доступ к конфигурационным файлам. Через эту библиотеку к программе был привязан файл конфигурации config.ini.

JSON

JSON - встроенная библиотека python для кодирования и декодирования данных JSON. Используется для экспорта данных в формат json.

Xlsxwriter

XlsxWriter – библиотека языка python предназначенная для записи файлов в формату Excel. Используется для экспорта данных в таблицу Excel.

## 1.3. Принципы формирования изображения на термометре.

В качестве объекта исследования стал цифровой погружной термометр TP300. Он обладает большим диапазоном измеряемых температур (-50°С - +300°С), погрешность измерений не превышает ±1°С на всём диапазоне, точностью в 0.1°С, а также термометр оснащён щупом из нержавеющей стали.

Показания термометра отображаются на стандартном семисегментном дисплее. Для преобразования цифрового сигнала в вид, пригодный для отображения на индикаторе, используется микросхема-дешифратор К176ИД2. Она и выполняет всю работу по формированию изображения. На рисунке 1 представлены изображения цифр отображаемых на дисплее термометра.



Рис. 1 Изображения цифр на дисплее термометра TP300.

## 1.4. Интерфейс программы.

Работа с программной разделена на 5 этапов:

* Transforming – этап предварительной подготовки видео, в котором происходит настройка масштабирования и вращения;
* Placement – этап расстановки ключевых точек;
* Naming – этап присваивания каждой точке имени, положения её сегмента в семисегментном индикаторе и распределения их между различными цифрами индикатора;
* Scanning – этап сканирования в котором на основании всех ключевых точек собирается информация о состоянии каждого элемента индикатора, эта информация дешифруется в формат целого числа и экспортируется в удобном формате;
* Fixing – этап коррекции положения ключевых точек, на случай если в процессе исследования температуры камера и термометр сместились друг относительно друга и расположение сегментов на изображении изменилось.

Последовательность этапов представлена на схеме:



Рис. 2 Последовательность этапов работы программы.

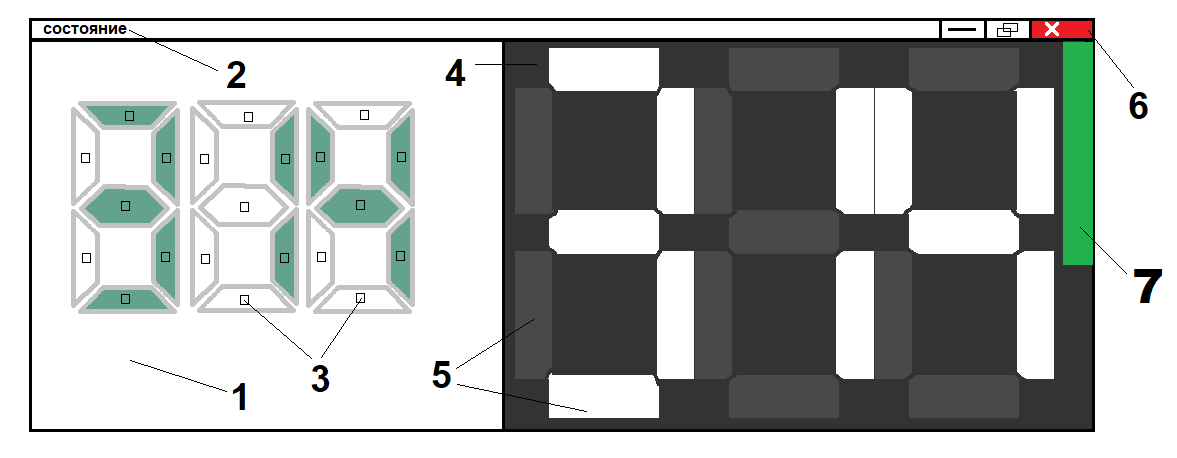


Рис. 3 Интерфейс программы.

Интерфейс программы состоит из 6 элементов:

1. Область просмотра исходного изображения;
2. Заголовок окна, в котором отображается текущий этап работы с программой;
3. Расставленные на исходном видео, ключевые точки. По цвету пиксля изображения определяется текущее значение на дисплее термометра;
4. Область предпросмотра текущей температуры, основанный на цвете ключевых точек;
5. Сегменты показателя;
6. Элементы управления окном.
7. Индикатор выполнения, представляющий собой прямоугольную область, которая «заполняется» областью зелёного цвета по мере сканирования.

В качестве механизма вывода полученных данных программа предоставляет 7 форматов экспорта:

1. RawTXT - текстовый файл, где все значения идут последовательно;
2. PythonList - текстовый файл, содержащий список всех значений, для исполнения в Python;
3. PythonDict - текстовый файл содержащий словарь всех значений с ключами, которые являются временем и пригодный для исполнения в Python;
4. JSON - текстовый файл JSON формата;
5. NumpyArray - бинарный файл массива библиотеки Numpy;
6. Excel – Еxcel таблица;
7. Graph - отображение графика на экране.

Программа настраивается через конфигурационный файл config.ini, который находится в одной директории с программой. Через него можно настроить следующие параметры:

1. videoPath – путь к видеофайлу;
2. startSec – секунда с которой начинается сканирование;
3. decimalPoint – число знаков после запятой в выходном файле;
4. exportFormat – формат экспорта выходных данных, возможные значения перечислены выше;
5. exportFileName – имя выходного файла.

Структура файла config.ini представлена на рисунке 4.



Рис. 4 Структура файла config.ini.

В данном случае сканируемое видео находится по относительному пути Experiments/E-1/video.mp4, сканирование начинается с 5 секунды, и термометр имеет один знак после запятой.

Выходные данные экспортируются в формат "Excel" в файл с названием data.xlsx.

# 2. РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ.

## 2.1. Разработка структуры программы.

### 2.1.1 Общая структура программы.

Программа разрабатывалась в методологии объектно-ориентированного программирования. Благодаря этому программу можно разделить на несколько независимых частей, каждая из которых играет определённую роль. Были созданы несколько классов. Два базовых класса:

* «VideoScaner» - Основной класс, который отвечает за обработку всей графической информации;
* «App» - класс приложения, отвечающий за работу с классом VideoScaner и экспорт полученных данных.

Также были созданы два класса, которые имеют физический аналог:

* «Segment» - объект, являющийся записью об определённом сегменте дисплея
* «Digit» - объект, являющийся записью об определённой цифре дисплея термометра и хранящий в себе массив объектов класса Segment.

Согласно руководству по написанию кода на python PEP 8, все слова в имени классов начинаются с большой буквы.

### 2.1.2 Структура класса VideoScaner.

#### 2.1.2.1 Поля класса VideoScaner.

Класс VideoScaner содержит поля представленные в таблице:

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип** | **Описание** |
| config | ConfigParser | Парсер конфигурационного файла |
| path | str | Путь к видеофайлу |
| capture | cv2.VideoCapture | Объект захвата видео из видеофайла |
| fps | float | Количество кадров в секунду видео |
| cropping | ((int, int),  (int, int)) | Координаты двух точек, прямоугольник между который и является отображается в результате обрезки |
| croppingHistory | list<((int, int),  (int, int))> | История всех обрезок |
| croppingArea | list<((int, int),  (int, int))> | Необработанные данные о обрезке |
| state | SetterState | Текущий этап работы программы |
| scaleF | float | Коэффициент масштабирования |
| rotate | int | Число от 0 до 3, для которого 90° \* rotate, равен углу поворота изображения |
| digits | list<Digit> | Список всех цифр дисплея |
| noNamedSegments | list<Segment> | Список ещё неназванных сегментов |
| segmentsHistory | list<Segment> | Историй расстановки сегментов |
| nameHistory | list<Segment> | Историй наименований сегментов |
| name\_index | int | Подсчет номера сегмента |
| noNamedDigits | list<Digit> | Список ещё не названных цифр |
| error\_count | int | Количество цифр, которые при сканировании выдали не точные данные |
| selection | list<Segment> | Список выделенных сегментов на этапе коррекции |
| decimalPoint | int | Число знаков после запятой в итоговом числе, полученное из конфигурационного файла |
| totalFrameCount | int | Общее число кадров в видеофайле |
| global\_scan\_data | dict<int, float> | Словарь итоговых данных, где ключ - время видеофайле, значение - показания на дисплее |
| currentSecScan | int | Текущая секунда сканируемого файла |
| scan\_data | list<dict<  SE, bool>> | Список значений всех сегментов в своих цифрах, отправляемый на дешифровку |

Далее рассмотрим механизмы данного класса.

#### 2.1.2.2 Отображения изображения на экране.

За вывод изображения на экран отвечает метод showFrame. Его код представлен далее:

self.frame = self.source\_img.copy()  
  
self.\_cropping()  
self.\_scale()  
self.\_rotate()  
  
self.sizeY, self.sizeX, \_ = self.frame.shape  
  
self.\_drawSegments()  
self.\_drawBad()  
  
cv2.imshow('Frame', self.frame)

Сначала в поле объекта frame мы записываем копию исходного кадра, полученного из видеофайла. Далее полученное изображение обрезается методом \_cropping, масштабируется для лучшего отображения методом \_scale, вращается методом \_rotate. Ширина и высота трансформированного изображения записываются в поля sizeX и sizeY соответственно. Рисуются все сегменты и область предпросмотра, если такая необходима, методами \_drawSegments и \_drawPreview. И в конце с помощью функции imshow в окне с идентификатором “Frame” отображается полученное изображение.

Все вышеперечисленные методы не предназначены для доступа извне класса, поэтому они являются защищёнными и их идентификаторы начинаются с символа нижнего подчёркивания.

#### 2.1.2.3 Реакция программы на события мыши.

Далее представлен код метода set.

def set(self):  
 self.showFrame()  
 cv2.setMouseCallback('Frame', self.onClick)  
  
 self.transform()  
 self.placement()  
 self.naming()

В данном методе, отвечающем за вызов этапов конфигурации программы, после первого отображения кадра их видеофайла вызывается модуля cv2 setMouseCallback, которая устанавливает для окна с идентификатором “Frame” метод onClick, как функцию вызываемую при действии мыши.

В методе onClick выполняется проверка текущего этапа работы программы из поля класс state, и, в зависимости от него, срабатывает различная логика. Их описание будет в частях описания этапов.

В случае если текущий этап «Transforming», по нажатию на левую кнопку мыши координаты записываются в первый элемент списка в поле “croppingArea”, а при отжатии кнопки мыши координаты записываются во второй элемент того же списка. Далее этот над этим списком производятся необходимые операции, выполняется проверка на размеры полученного прямоугольника. В случае успешного её прохождения, он преобразуется в неизменяемый картеж и записывается в поле cropping. Вызывается метод showFrame, изображение на экране обрезается.

Если текущий этап «Placement», вызывается метод setSegment с передачей координат мыши в качестве аргумента. В методе создаётся экземпляр класса Segment по полученным координатам. Полученный объект добавляется в список неназванных сегментов и в список истории сегментов. В случае нажатия клавиши backspace и программа не перешла на следующий этап, через метод removeLast, удаляется последний элемент списка segmentHistory.

Если текущий этап «Naming», относительно позиции мыши выполняется поиск ближайшего сегмента из списка неназванных и

## 2.2. Проблемы, возникшие во время разработки.

Ыва

## 2.3. Тестирование программы.

плр

# ЗАКЛЮЧЕНИЕ

# СПИСОК ЛИТЕРАТУРЫ

1. <https://docs.python.org/3/> - официальная документация по Python
2. <https://numpy.org/doc/> - официальная документация по NumPy
3. <https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html> - официальная документация по OpenCV для языка Python.
4. [https://docs.opencv.org/](https://docs.opencv.org/%20) - официальная документация по OpenCV для языка программирование C.
5. <http://www.catb.org/jargon/html/S/syntactic-sugar.html>
6. <https://www.python.org/dev/peps/pep-0008/> - руководство по написанию кода на Python