**МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение Московской области «Люберецкий техникум имени Героя Советского Союза, летчика-космонавта Ю.А.Гагарина»**

**ОТЧЕТ**

**УЧЕБНОЙ ПРАКТИКИ**

**\_\_\_\_\_\_\_\_***Кленина Елена Алексеевна***\_\_\_\_\_\_\_\_**

(Фамилия, имя, отчество студента)

по профессиональному модулю

**\_\_\_\_\_\_\_\_***ПМ.01 «Разработка модулей программного обеспечения***\_\_\_\_\_\_\_\_**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ для компьютерных систем»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Специальность \_*09.02.07 "Информационные системы и программирование"*\_

Код, название

Курс \_3\_ Группа № \_195\_

Период практики с «\_12\_» \_ноября\_ 20\_21\_ г. по «\_24\_» \_ноября\_ 20\_21\_ г.

Руководитель практики

от техникума \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Люберцы 2021

**Оглавление**

**1.**

**2.**

**3.**

**1.Индивидуальное задание на практику**

**2.Дневник прохождения практики**

Студента \_\_\_\_курса

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО

специальность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Дата** | **Краткое содержание выполнения работ** | **Отметка руководителя практики**  **(оценка, подпись)** |
| 1 | 2 |  | 3 |
|  |  |  |  |

Руководитель практики

от техникума \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись Фио

**3. Содержание текста отчета (по индивидуальному заданию, с фото)**

# Введение

Зачем нужна учебная практика?

Целью учебной практики является реализация компьютерного зрения с помощью библиотеки OpenCV и Основы С++.

Для достижения данной цели требуется решить следующие задачи:

1. Знакомство с литературой
2. Изучить библиотеку OpenCV
3. Реализация приложения на WPF

В ходе учебной практики планируется освоение следующего вида деятельности: «Разработка модулей программного обеспечения для компьютерных систем » (ПК):

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

# Изучение предметной области

Зачем нужна библиотека OpenCV? Зачем Ваш курс? Для чего разрабатывали приложение WPF

OpenCV — это open source библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений. Широко используется в таких языках как C, C++, Python и Java.

Мой курс помог мне разобраться и научил работать с C++, что поможет мне в будущем с работой.

Разработка приложения на WPF помогла нам научиться разрабатывать соединять простейшее приложение с базой данных SQL management Studio 18.

# Изучение программных средств

1. Для подключения OpenCV на python достаточно:

* Открываем PyCharm и нажимаем New project, выбираем папку, где хотим, чтобы был наш проект, и окружение Virtualenv, слева появятся папки
* File - Settings - Project Interpretater - шестеренка(add) - new enviroment. После этого Apply. Слева появится папка venv. Теперь все установленные библиотеки будут в виртуальном окружении и не будут засорять память
* Там же нажимаем на +, вводим opencv-python, install. Теперь появятся библиотеки нужные (https://youtu.be/iMM4TImNJFM)
* Заходим в Terminal Local python Имя\_файла.py

1. Недостатком C# является само подключение, для которого нужны изменения в фалах проекта и добавление дополнительных библиотек:
   * Скачать Open CV с сайта https://opencv.org/releases/ и выбрать версию 4.5.3 для Windows.
   * Установить Open CV (желательно сохранить в диске С или D D:\opencv)
   * Создаем пустой проект C++
   * Кликаем по названию проекта и кликаем на "Добавить" --> “Создать элемент” --> “Файл C++(.cpp)”
   * После этого по названию проекта кликаем правой кнопкой мыши и выбираем "Свойства"
   * Выбираем конфигурацию All Configurations, платформу х64 (в двух местах))
   * Находим C/C++ --> Раскрываем и находим "Общие" --> Находим каталоги включаемых файлов --> указываем путь openCv "...\opencv\build\include"
   * Далее выбираем "Компоновщик" --> "Общие" --> "Дополнительные каталоги библиотек" --> указываем путь "...\opencv\build\x64\vc15\lib"
   * Так же "Компоновщик" находим "Ввод" --> "Дополнительные зависимости и из папки lib выбираем opencv\_world453d.lib
   * Из папки "...opencv\build\x64\vc15\bin" копируете opencv\_world453d.dll в папку ...\Название вашего проекта\x64\Debug
2. WPF это библиотека для создания десктопных приложений для работы с ним мы включаем Visual Studio и создаём проект с WPF.
3. C++ используется во всех сферах деятельности программирования: от высоконагруженных систем до программирования микроконтроллеров. На С++ можно написать как web-сервер, так и игры, любые компьютерные программы, компоненты и так далее.

# OpenCV

1. Теория

Файл Contour\_main.cpp - это код по нахождению контуров

Файл momentmain.cpp - это дополнительно использование ползунка createTrackbar

Файл VideoTH.cpp - это оператор Собеля

Оператор Собеля — дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма. Используется в области обработки изображений, в частности, часто применяется в алгоритмах выделения границ.

VideoCapture cap – оператор для использования веб камеры

detectMultiScale — общая функция для распознавания как лиц, так и объектов. Чтобы функция искала именно лица, мы передаём ей соответствующий каскад.

Функция detectMultiScale принимает 4 параметра:

* + - 1. Обрабатываемое изображение в градации серого.
* Параметр scaleFactor. Некоторые лица могут быть больше других, поскольку находятся ближе, чем остальные. Этот параметр компенсирует перспективу.
* Алгоритм распознавания использует скользящее окно во время распознавания объектов. Параметр minNeighbors определяет количество объектов вокруг лица. То есть чем больше значение этого параметра, тем больше аналогичных объектов необходимо алгоритму, чтобы он определил текущий объект, как лицо. Слишком маленькое значение увеличит количество ложных срабатываний, а слишком большое сделает алгоритм более требовательным.
* minSize — непосредственно размер этих областей.

Сохранение изображения

import cv2

image = cv2.imread("./импорт/путь.расширение")

cv2.imwrite("./экспорт/путь.расширение", image)

Рисование прямоугольников

import cv2

output = image.copy()

cv2.rectangle(output, (2600, 800), (4100, 2400), (0, 255, 255), 10)

viewImage(output, "Обводим прямоугольником лицо пёсика")

Эта функция принимает 5 параметров:

* Само изображение.
* Координата верхнего левого угла (x1, y1).
* Координата нижнего правого угла (x2, y2).
* Цвет прямоугольника (GBR/RGB в зависимости от выбранной цветовой модели).
* Толщина линии прямоугольника.

Распознавание лиц

<https://github.com/Alena1380/Educational-practice/tree/main/openCV/python>

Конторы и центры масс

<https://github.com/Alena1380/Educational-practice/tree/main/openCV/control>

Распознавание дорожной разметки

<https://github.com/Alena1380/Educational-practice/tree/main/openCV/road>

Просмотр видео с веб камеры оператор собеля

[https://github.com/Alena1380/Educational- practice/tree/main/openCV/Prosmotr%20video%20s%20veb%20kamery](https://github.com/Alena1380/Educational-%20%20practice/tree/main/openCV/Prosmotr%20video%20s%20veb%20kamery)

# Основы С++

При объявлении программного элемента, такого как класс, функция или переменная, его имя может быть "видимым" и использоваться в определенных частях программы. Контекст, в котором отображается имя, называется его областью действия. Например, если объявить переменную x внутри функции, x она будет видна только в теле этой функции. Он имеет локальную область. У вас могут быть другие переменные с одним и тем же именем в программе; пока они находятся в разных областях, они не нарушают правило одного определения и не вызывают ошибку.

Для автоматических нестатических переменных область также определяет, когда они создаются и уничтожаются в памяти программ.

Существует шесть видов областей:

Глобальная область

Глобальное имя — это объявление, объявленное вне любого класса, функции или пространства имен. Однако в C++ даже эти имена существуют с неявным глобальным пространством имен. Область глобальных имен расширяется с точки объявления до конца файла, в котором они объявляются. Для глобальных имен видимость также регулируется правилами [компоновки](https://docs.microsoft.com/ru-ru/cpp/cpp/program-and-linkage-cpp?view=msvc-170) , которые определяют, является ли имя видимым в других файлах программы.

Область пространства имен

Имя, объявленное в [пространстве имен](https://docs.microsoft.com/ru-ru/cpp/cpp/namespaces-cpp?view=msvc-170)вне любого класса или определения перечисления или блока функции, видимо от его точки объявления до конца пространства имен. Пространство имен может быть определено в нескольких блоках для разных файлов.

Локальная область

Имя, объявленное внутри функции или лямбда-выражения, включая имена параметров, имеет локальную область. Они часто называются "локальными". Они видны только от их точки объявления до конца функции или тела лямбда-выражения. Локальная область — это разновидность области блока, которая обсуждается далее в этой статье.

Область класса

Имена членов класса имеют область класса, которая расширяется по всему определению класса независимо от точки объявления. Доступность членов класса дополнительно контролируется с помощью public private protected ключевых слов, и. Доступ к открытым или защищенным членам возможен только с помощью операторов выбора членов (. или -> ) или операторы указателя на член (. \* или -> \* ).

Область инструкции

Имена, объявленные for в if while операторе,, или, switch видимы до конца блока инструкции.

Область действия функции

[Метка](https://docs.microsoft.com/ru-ru/cpp/cpp/labeled-statements?view=msvc-170) имеет область видимости функции, что означает, что она видна в теле функции, даже до ее точки объявления. Область функции позволяет писать инструкции goto cleanup , например перед cleanup объявлением метки.

Программы на языке С++ обычно создаются в ходе компоновки одного или нескольких объектных файлов (с расширением .obj или .o) с одной или несколькими библиотеками. Библиотекой называется набор компонуемых файлов, которые поставляются вместе с компилятором. Все компиляторы С++ укомплектованы библиотекой функций (или процедур) и классов, которые можно включить в программу. Чтобы создать исполняемый файл, необходимо выполнить следующие действия:

1. Создать файл исходного кода с расширением .cpp;
2. Скомпилировать исходный код в объектный файл с расширением .obj или .o;
3. Скомпоновать объектный файл со всеми необходимыми библиотеками, получив исполняемый файл с расширением .exe.

Основной код программы начинается с вызова функции main(). Каждая программа на языке С++ содержит эту функцию. Функция – это блок кода программы, который выполняет одно или несколько действий. Обычно функцию вызывает другая функция или оператор, но функция main() – особая: она вызывается автоматически при запуске программы.

Объект cout используется для вывода сообщений на экран (Console Output – вывод на консоль (экран)). Для ввода сообщений с клавиатуры используется объект cin (Console Input – ввод с консоли (клавиатуры)). За словом **cout** стоит оператор вывода (<<). Все, что следует за этим оператором, будет выводиться на экран.

Чтобы завершить выполнение программы, нужно ввести любой символ или число, а затем нажать клавишу . Переменная типа **char** (сокращение слова ***character*** – читается как «кар») используется для хранения символов.

Чтобы приступить к созданию и отладке программы, необходимо выполнить следующие действия:

1. Запустить компилятор;
2. Выбрать в меню **File** (Файл) пункт **New** (Новый);
3. На вкладке **Projects**(Проекты) выберите тип **Win32 Console Application** (Консольное приложение для **Win32**), введите имя проекта, например, **hello**, и щелкните по кнопке OK;
4. В появившемся окне мастера нового проекта выберите переключатель **An Empty Project** (Пустой проект) и щелкните по кнопке Finish (Готово). В результате на экране появится диалоговое окно, отображающее информацию о новом проекте;
5. Чтобы вернуться к основному окну, щелкните по кнопке OK;
6. Выберите в меню **File**пункт **New**;
7. Во вкладке **Files** (Файлы) выберите пункт **С++ Source File** (Исходный файл С++) и введите его имя **hello**. Это имя уже было введено в текстовое поле **File Name** (Имя файла);
8. Чтобы вернуться к основному окну, щелкните по кнопке OK;
9. Введите текст программы;
10. Выберите в меню **Build** (Создать) пункт **Build hello.exe** (Создать **hello.exe**);
11. Убедитесь в отсутствии ошибок компиляции. Эта информация отображается в нижней части окна редактора кода;
12. Для запуска программы выберите в меню **Build** (Создать) пункт **Execute hello.exe** (Выполнить **hello.exe**) или нажмите комбинацию клавиш .

**WPF**

**Windows Presentation Foundation** — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework, использующая язык XAML.

* Для начала работы нам потребуется IDE Visual Studio 2022. Переходим на официальный сайт Microsoft и скачиваем [Visual Studio 2022 Community Edition](https://visualstudio.microsoft.com/ru/vs/community/), она бесплатна
* Затем запускаем exe файл, выбираем удобное место для установки и выбираем нужную среду разработки
* Затем, после установки, переходим к созданию проекта. Создаём проект и выбираем «Приложение WPF (NET.Framework)»
* Даём название нашему проекту и переходим к окну с кодом. В окне MainWindow.xaml с помощью кода прописываем дизайн нашего калькулятора
* Затем, в файле MainWindow.xaml.cs пишем функции для нашего калькулятора. Прописываем к каждой кнопке и полю
* После этого создаём новый проект, добавляем библиотеку MaterialDesignThemes c помощью «Управления пакетами NuGet»
* Затем, необходимо дописать код для подключения библиотеки к проекту, для этого переходим на [официальный сайт библиотеки](http://materialdesigninxaml.net/) и копируем следующий код. Вставляем его в файл App.xaml
* Далее, создадим дизайн программы с помощью кода, писать его нужно в файле MainWindow.xaml
* После создания дизайна, переходим к получению данных от пользователя. Нам необходимо получить данные от пользователя и проверить их. Для этого пишем проверки вводимых данных
* Далее, переходим к тестированию нашей программы. Вводим некорректные данные и видим, что наши поля для ввода «горят красным цветом». Это значит, что наши проверки работают и можно переходить к следующему этапу разработки.
* Следующим этапом разработки является создание и подключение Базы данных к нашей программе. Для этого нам понадобится программа Microsoft SQL Server Management Studio 18. Скачать эту программу можно на [официальном сайте Microsoft](https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15#download-ssms)
* Следуя инструкциям по установке, производим установки программы и запускаем её. Перед нами открывается окно программы. Вводим требуемые данные и переходим к созданию нашей базы данных
* Затем, в обозревателе объектов выбираем «Базы данных» и создаём базу данных
* После создания базы данных создаём таблицу «Users»
* После создания базы данных и таблицы, переходим к подключению базы к нашему приложению. Для этого нужно нажать правой кнопкой мыши по файлу нашего проекта и нажать кнопку «Добавить» 🡪 «Создать элемент». В окне «Добавление нового элемента» в поле «Поиск» вводим «edm» и добавляем Модель ADO.NET.EDM
* Далее, следуя инструкция, вводим имя нашего сервера, выбираем нашу БД, выбираем таблицу «Users» и затем, происходит подключение
* После этого, создаём класс-модель для работы с нашей таблицей
* Затем, создаём класс для работы с базой данных
* Следующим шагом мы будем добавлять объект в БД. Для этого прописываем следующий код
* После того, как мы добавили данные в нашу таблицу, нам необходимо отобразить наши данные. Для этого мы прописываем следующий код
* Далее, проверяем нашу программу
* После проверки нашей программы, переходим к окну с авторизацией. Создаём новое окно AuthWindow.xaml, в файле AuthWindow.xaml.cs прописываем код стиля, после чего получается следующее окно
* Затем, добавим проверку введённых пользователем данных и сверение их с данными из нашей Базы данных
* После написания кода, проверяем работу нашей программы
* После создания окна авторизации, можно перейти к созданию «Кабинета пользователя». Для этого нам необходимо настроить переход между страницами. Далее, создадим само окно «Кабинет пользователя». Прописываем код вывода всех зарегистрированных пользователей и проверяем работу программы.
* Вот мы и закончили создание нашего приложения WPF. Во время разработки этого приложения, мы научились работать с базой данных, создавать переход между окнами, добавлять новых пользователей в базу данных и т.д.Разработка данного приложения нужна была для закрепления навыков создания подобных приложений и работы с базой данных.

# Выводы

По окончании учебной практики были решены следующие задачи:

1. Знакомство с литературой
2. Изучить библиотеку OpenCV
3. Реализация приложения на WPF

В ходе этой практики были сформированы следующие профессиональные компетенции:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

1. **Список литературы**
2. [Орлов С.А.](http://library.sgu.ru/cgi-bin/irbis64r_15/cgiirbis_64.exe?LNG&amp;Z21ID&amp;I21DBN=NIKA&amp;P21DBN=NIKA&amp;S21STN=1&amp;S21REF=1&amp;S21FMT=fullwebr&amp;C21COM=S&amp;S21CNR=20&amp;S21P01=0&amp;S21P02=1&amp;S21P03=A%3D&amp;S21STR=%D0%9E%D1%80%D0%BB%D0%BE%D0%B2%2C%20%D0%A1%D0%B5%D1%80%D0%B3%D0%B5%D0%B9%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%D0%B8%D1%87) Программная инженерия. Технологии разработки программного обеспечения: учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Москва ; Санкт- Петербург:Питер, 2016. – 640 с.
3. Джепикс Филипп, Троелсен Эндрю. Язык программирования C# 7 и платформы .NET и .NET Core / Джепикс Филипп, Троелсен Эндрю – М.; Вильямс, 2018. – 1328. «Наука», 2017 – 87с.
4. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2017. – 461 с.
5. Семакин И. Г. Основы алгоритмизации и программирования : учебник для студ. учреждений сред. проф. образования / И. Г. Се- макин, А. П. Шестаков. — М. : Издательский центр «Академия», 2017. — 304 с
6. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. – СПб.: БХВ– Петербург, 2017. – 512 с.
7. Прайс Марк. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / Прайс Марк – Москва; Вильямс, 2018. – 604.
8. Скит Джон. C# для профессионалов. Тонкости программирования / Скит Джон – Москва; Вильямс, 2019. – 608.
9. Мюллер Д. П. C# для чайников / Мюллер Д. П. – Москва; Москва, 2019, – 608 с.

**5. ПРИЛОЖЕНИЯ (при наличии)**