**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**

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

(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 «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. по «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г.

Руководитель практики

от техникума Ревнивцев Максим Валерьевич

Жирнова Юлия Витальевна

Люберцы 2021

**Оглавление**

**1.Индивидуальное задание на практику**

**2.Дневник прохождения практики**

Студента 3 курса

Лидовской Иван Николаевич

Специальность: ИС и программирование

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

Руководитель практики

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

подпись Фио

# Введение

Целью учебной практики является: закрепление теоретических знаний, полученных при изучении базовых дисциплин; изучение организационной структуры предприятия и действующей на нем системы управления; ознакомление с содержанием основных работ и исследований, выполняемых на предприятии или в организации по месту прохождения практики.

Целью учебной практики является реализация компьютерного зрения с помощью библиотеки OpenCV, и разработка приложения с помощью WPF и 3DS Max

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

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 3D studio Max, 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. 3Ds Max это среда для создания 3D моделей и образов.

# 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/DeLarge666/Practiva-Lidovskogo/tree/main/OpenCV/python>

Конторы и центры масс

<https://github.com/DeLarge666/Practiva-Lidovskogo/tree/main/OpenCV/control>

Распознавание дорожной разметки

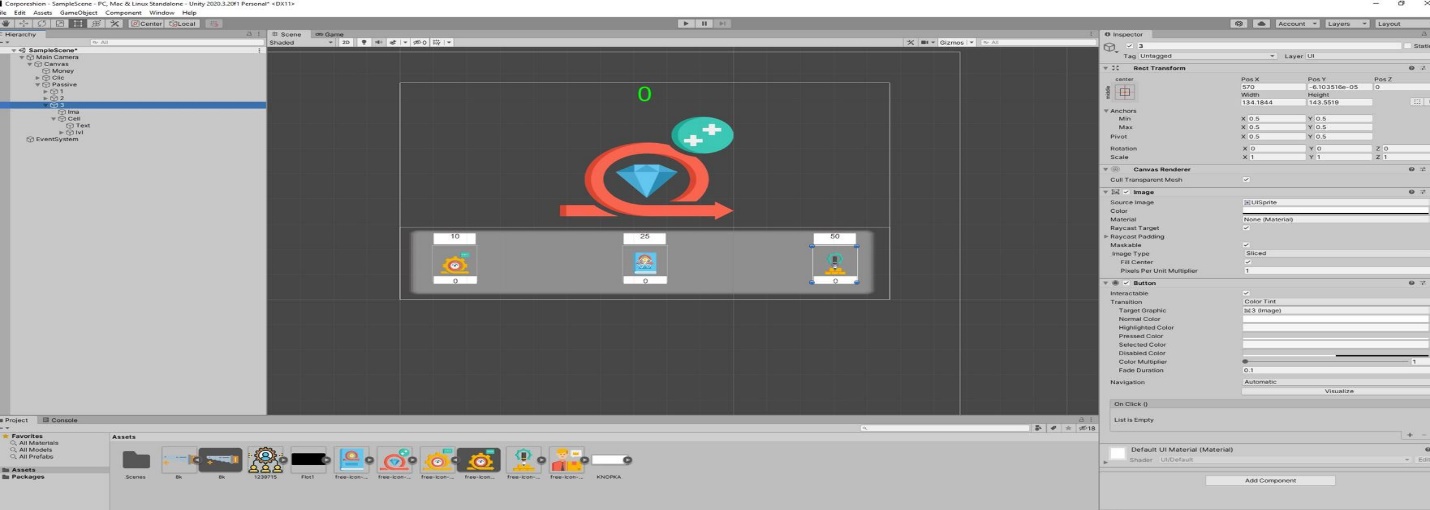
<https://github.com/DeLarge666/Practiva-Lidovskogo/tree/main/OpenCV/doroga>

Просмотр видео с веб камеры оператор собеля

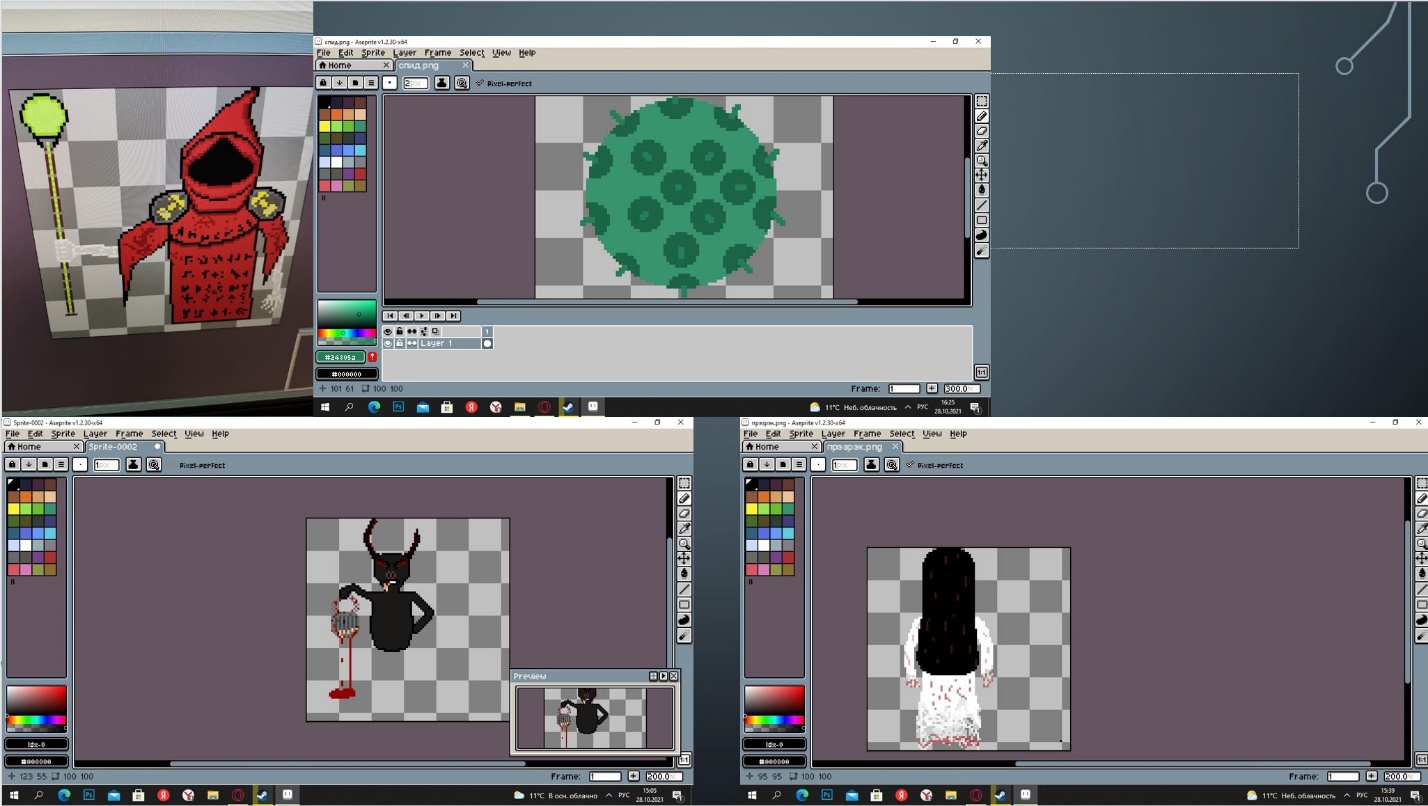
<https://github.com/DeLarge666/Practiva-Lidovskogo/tree/main/OpenCV/1>

# Ваш курс

* **Файлы** с расширением “**.un”** открываются в программе “Unity” стандартным способом
* Audio Source и Trigger auto компоненты Unity позволяющие добавлять звуки и музыку
* Trigger - **Триггер** — **это** зона срабатывания определенных действий, когда объект входит в эту зону, находится в ней или выходит за ее пределы.
* Спрайт – Sprites – Это изображения, которые используются как графика в игре
* Создаём проект и начинаем добавлять разные компоненты, которые и будут элементами игры
* Рассмотрим примеры компонентов:
* RigBody (добавление “веса” и физики)
* Audio Source (добавление постоянной музыки)
* Image Effects (добавление эффектов для спрайтов)
* Box Collider (элемент которые позволяет добавлять тригер и текстуру объекту )
* Element (элемент)







**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.

**4. Список литературы**

1. [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.
2. 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.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2017. – 461 с.
4. 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
5. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. – СПб.: БХВ– Петербург, 2017. – 512 с.
6. Прайс Марк. C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов / Прайс Марк – Москва; Вильямс, 2018. – 604.
7. Скит Джон. C# для профессионалов. Тонкости программирования / Скит Джон – Москва; Вильямс, 2019. – 608.
8. Мюллер Д. П. C# для чайников / Мюллер Д. П. – Москва; Москва, 2019, – 608 с.
9. [Главная | GeekBrains - образовательный портал (gb.ru)](https://gb.ru/)
10. https://opencv.org/courses/
11. <https://robocraft.ru/page/opencv/>
12. Skillbox: <https://live.skillbox.ru/playlists/code/python/>

**5. ПРИЛОЖЕНИЯ (при наличии)**