**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**

**\_\_\_\_\_\_\_***Громова Никиты Сергеевича*

(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**

[Введение                                                                                                                              3](https://docs.google.com/document/d/1WxC7U7BJg-haKaleu3ikniprsEiU8tyy/edit#heading=h.gjdgxs)

Изучение предметной области                                                                                            5

[Изучение программных средств                                                                                          6](https://docs.google.com/document/d/1WxC7U7BJg-haKaleu3ikniprsEiU8tyy/edit#heading=h.3znysh7)

[OpenCV                                                                                                                                 8](https://docs.google.com/document/d/1WxC7U7BJg-haKaleu3ikniprsEiU8tyy/edit#heading=h.xiho1x7b05n6)

Основы языка Python                                                                                                     14

Приложение WPF                                                                                                               19

[Выводы                                                                                                                               31](https://docs.google.com/document/d/1WxC7U7BJg-haKaleu3ikniprsEiU8tyy/edit#heading=h.3dy6vkm)

[Список](https://docs.google.com/document/d/1WxC7U7BJg-haKaleu3ikniprsEiU8tyy/edit#heading=h.3dy6vkm) литературы 32

# Введение

Учебная практика – это неотъемлемая часть учебного процесса, в ходе которого студенты самостоятельно выполняют определенные задачи в условиях реально действующего предприятия и преследуют цель приобретения студентами навыков профессиональной работы по направлению.

Целью учебной практики является реализация компьютерного зрения с помощью библиотеки OpenCV и Основы языка Python.

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

**Предметная область** - множество понятий и объектов, рассматриваемых в пределах отдельного рассуждения, исследования или научной теории.

**Windows Presentation Foundation (WPF)** — это среда разработки, используемая для создания настольного приложения. Это часть .NET Framework. WPF имеет независимый от разрешения и рендеринг на основе векторов, который полезен для работы с современным графическим оборудованием[1].

**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.

# 

# Изучение программных средств

**Microsoft Visual Studio** - это интегрированная среда разработки (IDE) от компании Microsoft.

Преимущества:

1. Сервер для среды. С помощью VS и встроенного сервера ASP.NET можно не только работать элементами, но и запускать сайт прямо из среды. Этот подход значительно опережает аналогичные облачные сервисы: на стадии разработки никто не сможет получить доступ к продукту в оффлайне;
2. Универсальность языков. Некоторые IDE поддерживают часть языков, некоторые – только один. Microsoft разработали средство разработки, которое поддерживает превалирующее число PL. Но поддержкой не заканчивается универсальность, особенно это касается веб-разработки: в Visual Studio можно создавать страницы на разных языках, а после поместить их все в одном приложении;
3. Работая с другими средствами для разработки приходится писать много вещей, которые в VS добавляются благодаря интеллектуальной системе;
4. Код, читаемый с первых строк. Интеллектуальная система среды разработки умеет подстраивать код под необходимый формат и синтаксис. Таким образом, он становится более читаемым и доступным для редактирования.

**Pycharm** — это кросс-платформенная среда разработки для языка Python, которая совместима с Windows, macOS, Linux. PyCharm Community Edition (бесплатная версия) находится под лицензией Apache License, а PyCharm Professional Edition (платная версия) является проприетарным ПО.

Преимущества:

1. PyCharm имеет удобный редактор кода со всеми полезными функциями: подсветкой синтаксиса, автоматическим форматированием, дополнением и отступами. PyCharm позволяет проверять версии интерпретатора языка на совместимость, а также использовать шаблоны кода.
2. Утилита поддерживает все свежие версии Django, а также IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK и многие другие инструменты.
3. PyCharm имеет большую коллекцию плагинов, и его можно использовать в связке с разными трекерами вроде JIRA, Youtrack, Lighthouse, Redmine, Trac и так далее.
4. PyCharm кросс-платформенная среда разработки: можно использовать на Linux, Windows и Mac OS.

**SQL Server Management Studio (SSMS)** — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, Базы данных SQL Azure и Azure Synapse Analytics, а также управления ими.

Преимущества:

1. СУБД масштабируется, поэтому работать с ней можно на портативных ПК или мощной мультипроцессорной технике. Процессор может одновременно обрабатывать большой объем запросов.
2. Реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы.
3. Рутинные административные задачи автоматизированы: это управление блокировками, памятью, редактура размеров файлов. У системы продуманы настройки, можно создать профили пользователей.
4. Размер страниц – до 8 кб, поэтому данные извлекаются быстро, подробную и сложную информацию хранить удобнее. Система позволяет обрабатывать транзакции в интерактивном режиме, есть динамическая блокировка.
5. СУБД поддерживает работу с другими продуктами Microsoft: Access, MS Excel.

# 

# OpenCV

**OpenCV** - библиотека с открытым исходным кодом для машинного зрения и машинного обучения. Он имеет 2500+ оптимизированных алгоритмов - полный набор как классических, так и самых современных алгоритмов компьютерного зрения и машинного обучения. У него много реализаций, включая Python, Java, C ++ и Matlab.

Эта библиотека очень популярна за счёт своей открытости и возможности бесплатно использовать как в учебных, так и коммерческих целях.

Все работы за учебную практику по разделу Opencv были загружены на GitHub в папку Educational-Practice/Opencv.

**Описание кода.**

В самом начале нам понадобится создать проект в VS.

Открыв приложение видим меню, нажимаем кнопку «Создание проекта». Далее будет предложено создать проект нужного нам расширения, выбираем «Консольное приложение», нажимаем далее.

Вводим запрашиваемую приложением информацию(путь, название проекта), нажимаем «создать».

Проект создан, на данном этапе требуется подключить библиотеку Opencv, которую заранее надо скачать (полная инструкция по подключению библиотеки находится в презентации, которая выложена на GitHub в репозитории Educational-Practice в файле readme.md).

Для трансляции видео в VS было использовано приложение OBS studio.

Работа с кодом:

Для загрузки изображения мы используем функцию cv2.imread(), где первым аргументом указывается путь к изображению, а вторым аргументом, который является необязательным, мы указываем, в каком цветовом пространстве мы хотим считать наше изображение. Чтобы считать изображение в RGB — cv2.IMREAD\_COLOR, в оттенках серого — cv2.IMREAD\_GRAYSCALE. По умолчанию данный аргумент принимает значение cv2.IMREAD\_COLOR. Данная функция возвращает 2D (для изображения в оттенках серого) либо 3D (для цветного изображения) массив NumPy. Форма массива для цветного изображения: высота x ширина x 3, где 3 — это байты, по одному байту на каждую из компонент. В изображениях в оттенках серого всё немного проще: высота x ширина.

**def** **loading\_displaying\_saving**():

img = cv2.imread('girl.jpg', cv2.IMREAD\_GRAYSCALE)

cv2.imshow('girl', img)

cv2.waitKey(0)

cv2.imwrite('graygirl.jpg', img)

Для того, чтобы узнать высоту, ширину и количество каналов у изображения можно использовать атрибут shape:

print("Высота:"+str(img.shape[0]))

print("Ширина:" + str(img.shape[1]))

print("Количество каналов:" + str(img.shape[2]))

Чтобы получить доступ к значению пикселя, нам просто нужно указать координаты x и y пикселя, который нас интересует.

(b, g, r) = img[0, 0]

print("Красный: {}, Зелёный: {}, Синий: {}".format(r, g, b))

Cначала мы берём пиксель, который расположен в точке (0,0). Данный пиксель, да и любой другой пиксель, представлены в виде кортежа. Заметьте, что название переменных расположены в порядке b, g и r. В следующей строке выводим значение каждого канала на экран. Как можно увидеть, доступ к значениям пикселей довольно прост, также просто можно и манипулировать значениями пикселей:

img[0, 0] = (255, 0, 0)

(b, g, r) = img[0, 0]

print("Красный: {}, Зелёный: {}, Синий: {}".format(r, g, b))

Распознавание лиц

import cv2

image\_path = "./путь/к/фото.расширение"

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

image = cv2.imread(image\_path)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(

gray,

scaleFactor= 1.1,

minNeighbors= 5,

minSize=(10, 10)

)

faces\_detected = "Лиц обнаружено: " + format(len(faces))

print(faces\_detected)

# Рисуем квадраты вокруг лиц

for (x, y, w, h) in faces:

cv2.rectangle(image, (x, y), (x+w, y+h), (255, 255, 0), 2)

viewImage(image,faces\_detected)

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.
2. Параметр scaleFactor. Некоторые лица могут быть больше других, поскольку находятся ближе, чем остальные. Этот параметр компенсирует перспективу.
3. 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.
4. minSize — непосредственно размер этих областей.

## Contours — распознавание объектов

Распознавание объектов производится с помощью цветовой [сегментации изображения](https://www.fritz.ai/features/image-segmentation.html). Для этого есть две функции: cv2.findContours и cv2.drawContours

Разметка дороги

Цель данного проекта — построить простую линейную модель для покадрового распознавания полос движения.

Процесс построения детектора состоит из трех основных шагов:

* Предобработка данных, фильтрация от шума и векторизация изображения.
* Обновление состояния линий дорожной разметки по данным из первого шага.
* Рисование обновленных линий и других объектов на исходном изображении.

Сначала на вход функции image\_pipeline подается 3-канальное изображение формата RGB, которое затем фильтруется, преобразовывается, а внутри функции обновляются объекты Line и Lane . Затем поверх самого изображения рисуются все необходимые элементы

Применим два преобразования:

1. [Детектор границ Кэнни](https://medium.com/@tempflip/lane-detection-with-numpy-56b923245fc9#.ek9oe9ucu): алгоритм оптимального определения границ, который рассчитывает градиенты интенсивности изображения, а затем с помощью двух порогов удаляет слабые границы, оставляя искомые (мы используем (280, 360) ) как пороговые значения в функции canny .
2. Преобразование Хафа: получив границы с помощью алгоритма Кэнни, мы можем соединить их с помощью линий. Я не хочу вдаваться в математику алгоритма — она достойна отдельного поста — эта ссылка или ссылка выше поможет вам, если вас заинтересовал метод. Главное, что, применив это преобразование, мы получаем набор линий, каждая из которых, после небольшой дополнительной обработки и фильтрации, становится экземпляром класса Line с известным углом наклона и свободным членом.

Оператор Собеля

**Оператор Собеля** — это дискретный дифференциальный оператор, вычисляющий приближение градиента яркости изображения.  
Оператор вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы.

В OpenCV оператор Собеля реализуется функцией cvSobel()  
Вычисление производной изображения (градиента), используя оператор Собеля (aperture\_size = 1,3,5,7) или Щарра (aperture\_size = -1):  
CVAPI(void) cvSobel( const CvArr\* src, CvArr\* dst,

int xorder, int yorder,

int aperture\_size CV\_DEFAULT(3));

src — исходное изображение  
dst — изображение для сохранения результа  
xorder — порядок производной по x (0,1 или 2)  
yorder — порядок производной по y (одновременно нулевой может быть только либо xorder, либо yorder)  
aperture\_size — размер ядра оператора Собеля (1,3,5,7)

Центр масс

**Преобразование Хафа (Hough Transform) —** это метод для поиска линий, кругов и других простых форм на изображении.

Преобразование Хафа основывается на представлении искомого объекта в виде параметрического уравнения. Параметры этого уравнения представляют фазовое пространство (т.н. аккумуляторный массив/пространство, пространство Хафа).  
Затем, берётся двоичное изображение (например, результат работы детектора границ Кенни). Перебираются все точки границ и делается предположение, что точка принадлежит линии искомого объекта — т.о. для каждой точки изображения рассчитывается нужное уравнение и получаются необходимые параметры, которые сохраняются в пространстве Хафа.

**CV\_HOUGH\_STANDARD** — классический вариант трансформации Хафа. Каждая линия представляется двумя числами типа float (rho, theta), где rho — дистанция между точкой (0,0) и линией, а theta — угол между осью x и нормалью к линии (т.е. матрица должна иметь тип of CV\_32FC2)  
**CV\_HOUGH\_PROBABILISTIC** — вероятностный метод трансформации Хафа (более эффективен с случае изображений с несколькими длинными линейными сегментами). Возвращает сегменты линии, которые представляются точками начала и конца (т.е. матрица должна иметь тип of CV\_32SC4)  
**CV\_HOUGH\_MULTI\_SCALE** — масштабный вариант классической трансформации Хафа. Линии представлены так же, как и в CV\_HOUGH\_STANDARD

vector<Moments>mu(contours.size()); //создается вектор моментов, куда передается количество контуров, которое зависит от изображения

drawContours(drawing, contours, i, color, 2, 8, hierarchy, 0, Point()); //передается картинка, hierarchy - возвращает иерархию внешних контуров и отверстий.

circle(drawing, mc[i], 4, color, -1, 5, 0); //выбирается изображение, центр масс, 4 - радиус, цвета, толщина

Нейросети

Чтобы создать полный проект по распознаванию лиц, мы должны работать на 3 очень разных этапах:

1. Обнаружение лиц и сбор данных
2. Обучение распознавателя
3. Распознавание лица

Самой основной задачей Распознавания Лица является, конечно же, «Определения Лица». Прежде всего, вы должны «захватить» лицо, чтобы распознать его, по сравнению с новым лицом, захваченным в будущем.

Наиболее распространенный способ обнаружения лица (или любых объектов), использует [«классификатор каскадов Хаар»](https://docs.opencv.org/3.3.0/d7/d8b/tutorial_py_face_detection.html).

Обнаружение объектов с использованием «классификатора каскадов Хаар» - эффективный метод обнаружения объектов, предложенный Полом Виолой и Майклом Джонсом в своей статье «Быстрое обнаружение объектов с использованием расширенного каскада простых функций» в 2001 году. Это подход, основанный на механизме обучения, каскадная функция обучается из множества положительных и отрицательных изображений. Затем он используется для обнаружения объектов на других изображениях.

Рассмотрим установку OpenCV только для Python. Мы можем установить ее при помощи менеджеров pip или conda (в случае, если у нас установлен пакет Anaconda).

Извлекаем имя человека из названия папки:

name = imagePath.split(os.path.sep)[-2]

Загружаем изображение и конвертируем его из BGR (OpenCV ordering) в dlib ordering (RGB):

image = cv2.imread(imagePath)

rgb = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

Используем библиотеку Face\_recognition для обнаружения лиц:

boxes = face\_recognition.face\_locations(rgb,model='hog')

Вычисляем эмбеддинги для каждого лица

encodings = face\_recognition.face\_encodings(rgb, boxes)

# Основы языка 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, а также на обеспечение переносимости написанных на нём программ.

Преимущества:

1. Язык характеризуется логичным синтаксисом, вследствие чего исходный код программ, написанных «на питоне», легко читается и воспринимается.
2. Еще одно из преимуществ данного языка программирования — его условная легкость. Он считается наиболее подходящим для начинающих специалистов: разрабатывать несложные программы можно научиться уже спустя пару-тройку дней изучения.
3. Большое интернет-сообщество. Если разработчик сталкивается с вопросами и трудностями, он всегда может спросить совета у коллег, что значительно ускоряет решение проблем.
4. Гибкость и масштабируемость. Python позволяет разработчикам адаптировать высокоуровневую логику приложения, что позволяет легко расширять сложные приложения по мере необходимости.
5. Разработка «на питоне» идет быстрее, чем на большинстве других языках.
6. Python является интерпретируемым языком программирования. Это значит, что до запуска он представляет собой обычный текстовый файл. Соответственно, программировать можно почти на всех платформах.

Для работы с этим языком было выбрано приложение Pycharm. Предварительно был установлен интерпретатор с официального сайта python.org.

Создание и запуск проекта в Pycharm (рисунок 1).

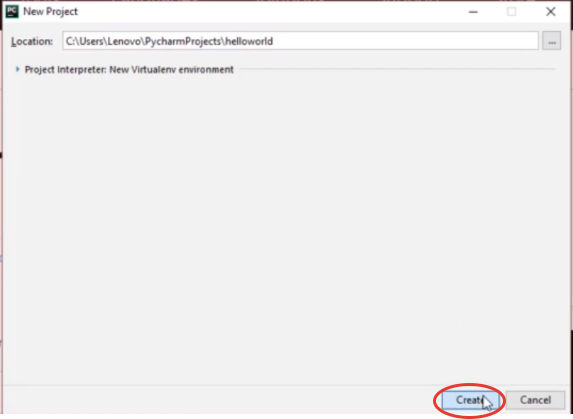


Рисунок 1.

Создадим Python File, чтобы мы могли приступить к коду (рисунок 2).

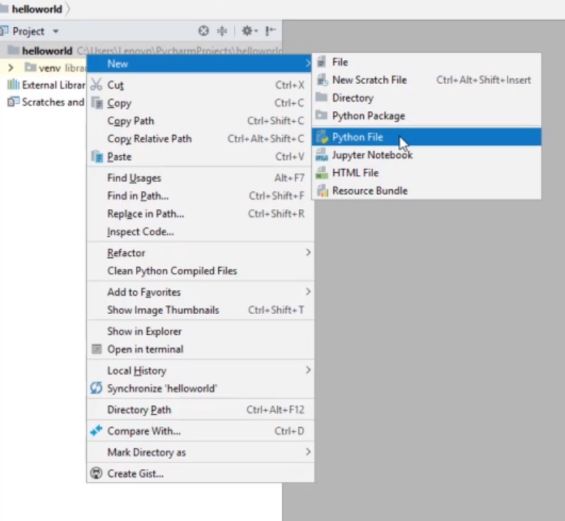


Рисунок 2,

После несложных манипуляций мы можем приступать к кодированию. Для этого нам понадобятся некоторые знания о переменных:

* Переменные используются для хранения данных.
* Переменные можно использовать несколько раз.
* Можно менять значение и тип переменной.

Объявление переменной:

имя\_переменной = значение переменной

name = «Кеша»

Простые типы переменных:

* Целое число – int
* Число с плавающей точкой - float
* Логический тип (истина/ложь) - bool
* Ничего (пустой тип) - none
* Строка – str

В конце второго урока курса нам выдается три практических задания, которые мы должны выполнить самостоятельно. Решение и условие этих заданий приводится ниже.

Практическое задание 1

Запросите от пользователя число, сохраните в переменную, прибавьте к числу 2 и выведите результат на экран. Если возникла ошибка, прочитайте ее, вспомните урок и постарайтесь устранить ошибку (рисунок 3).

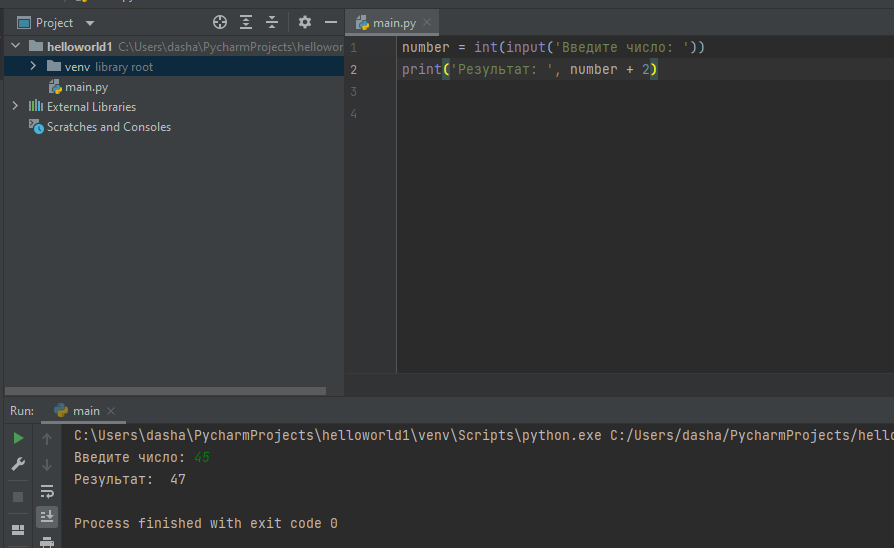


Рисунок 3.

Практическое задание 2

Используя цикл, запрашивайте у пользователя число, пока оно не станет больше 0, но меньше 10. После того, как пользователь введет корректное число, возведите его в степень 2 и выведите на экран. Например, пользователь вводит число 123, вы сообщаете ему, что число неверное, и говорите о диапазоне допустимых. И просите ввести заново. Допустим, пользователь ввел 2, оно подходит. Возводим его в степень 2 и выводим 4 (рисунок 4).

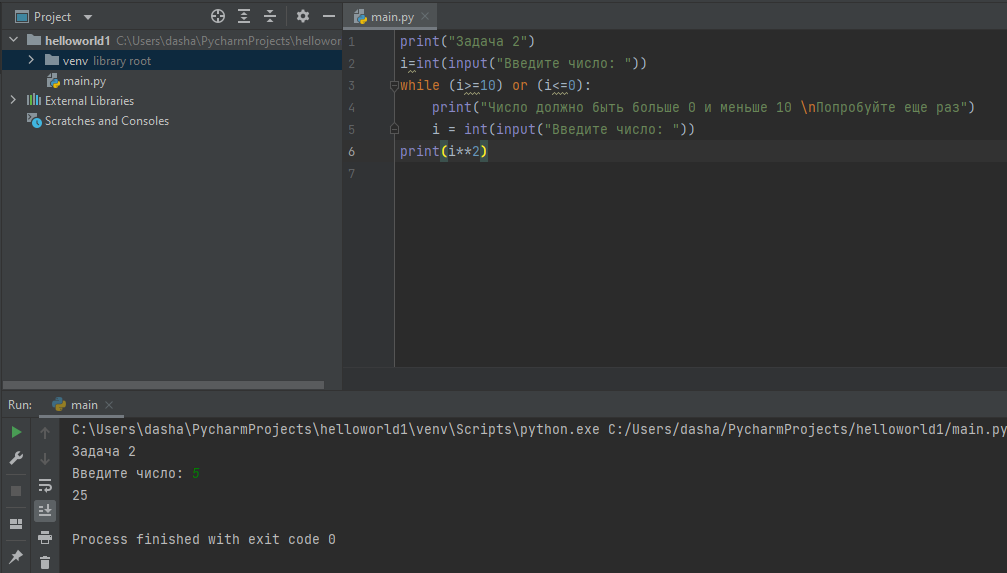


Рисунок 4.

Практическое задание 3

Создайте программу “Медицинская анкета”, где вы запросите у пользователя следующие данные: имя, фамилия, возраст и вес (рисунок 5).

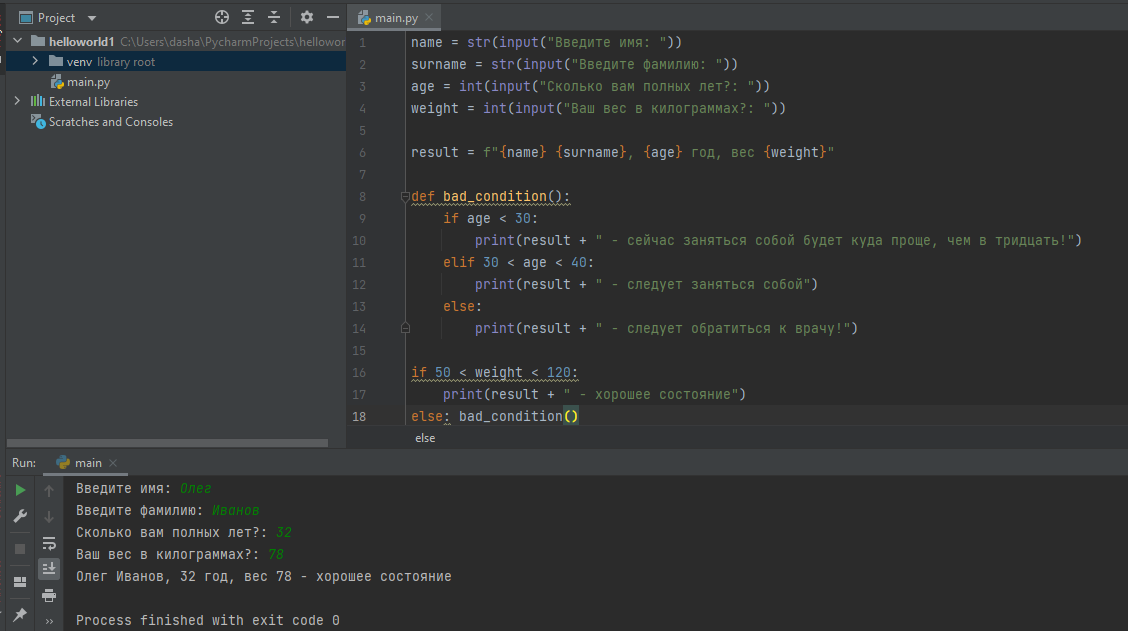


Рисунок 5.

**WPF**

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

В основе WPF лежит независимый от разрешения векторный модуль визуализации, использующий возможности современного графического оборудования. Возможности этого модуля расширяются с помощью комплексного набора функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двумерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF является частью .NET, поэтому вы можете создавать приложения, включающие другие элементы .NET API.

Чтобы начать работу 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/) (рисунок 6).

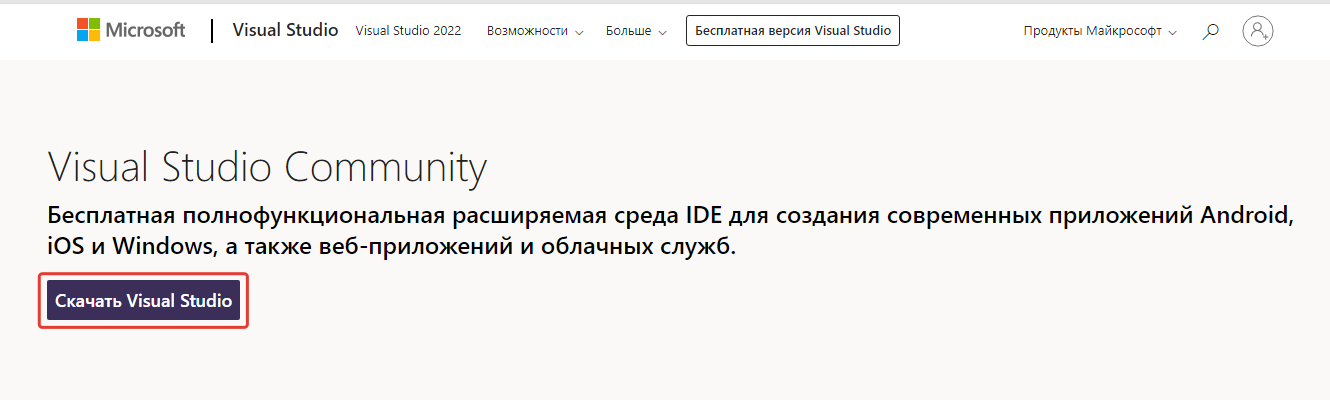


Рисунок 6.

После стандартной установки переходим к созданию проекта. Создаём проект и выбираем «Приложение WPF (NET.Framework)» (рисунок 7).

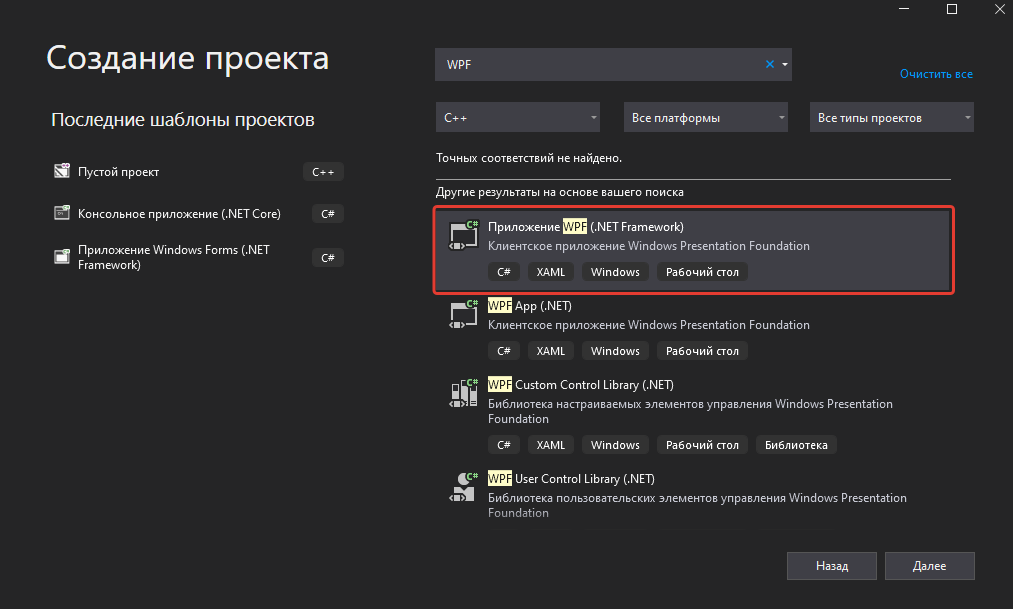


Рисунок 7.

Называем наш проект. В окне MainWindow.xaml с помощью кода прописываем дизайн нашего калькулятора (рисунок 8).

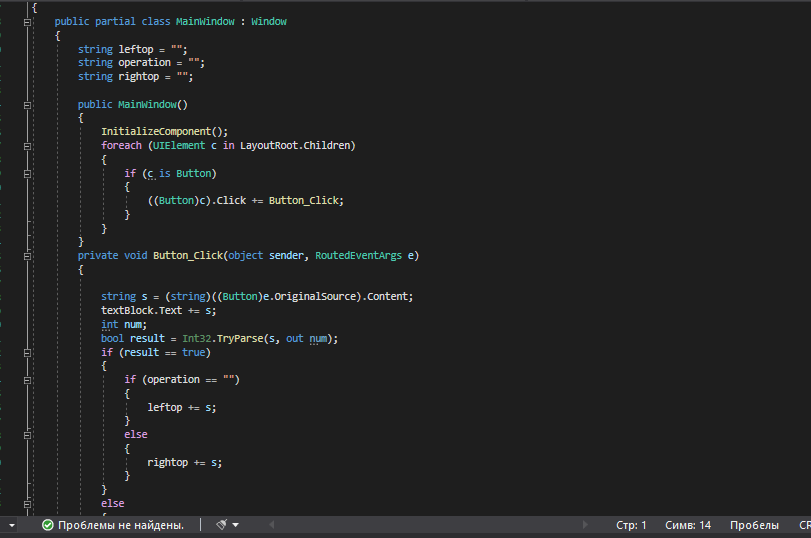


Рисунок 8.

В файле MainWindow.xaml.cs напишем функции для калькулятора (рисунок 9).

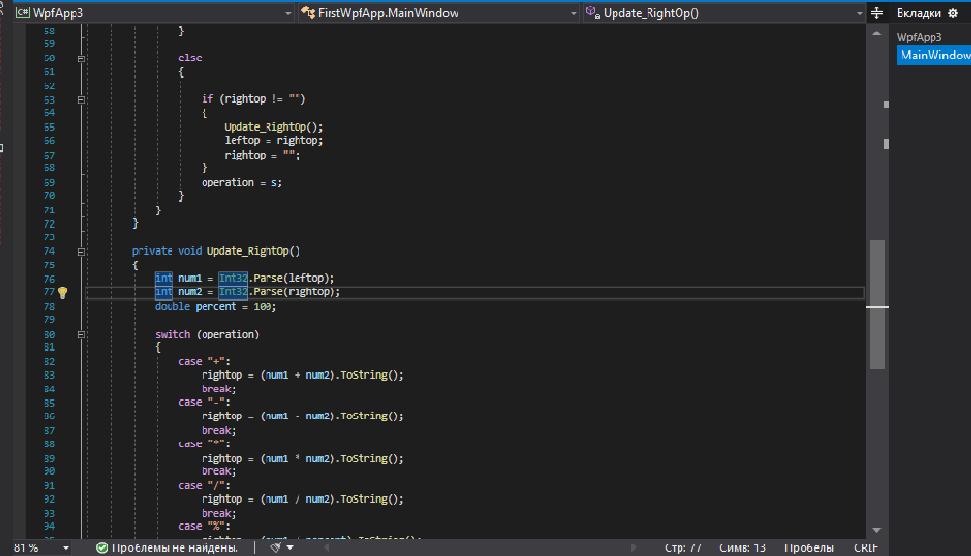


Рисунок 9.

После нам понадобится создать новый проект. Добавим в него библиотеку MaterialDesignThemes c помощью «Управления пакетами NuGet» (рисунок 10).

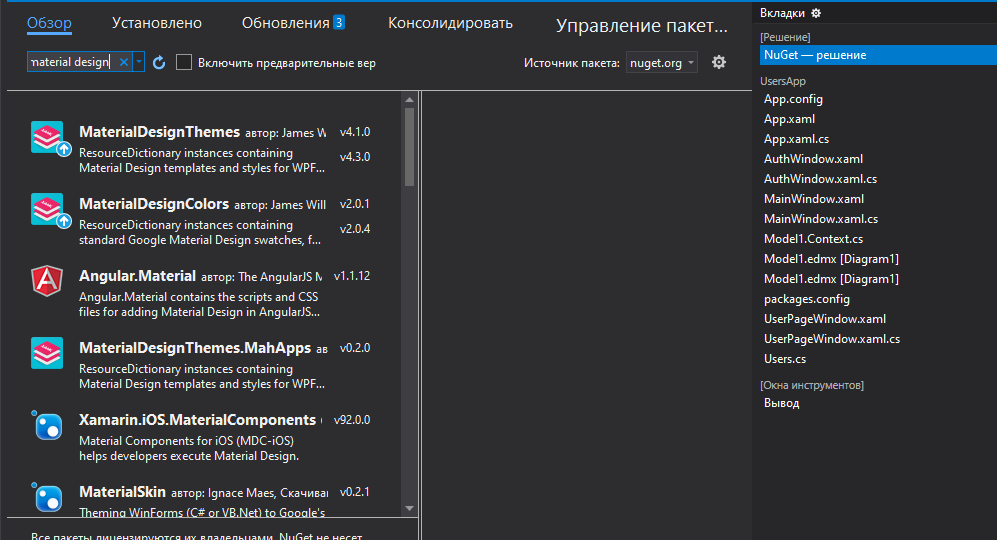


Рисунок 10.

Чтобы подключить библиотеку к проекту нужно перейти на официальный сайт этой библиотеки, и, скопировав нужный код, вставить его в App.xaml (рисунок 11).

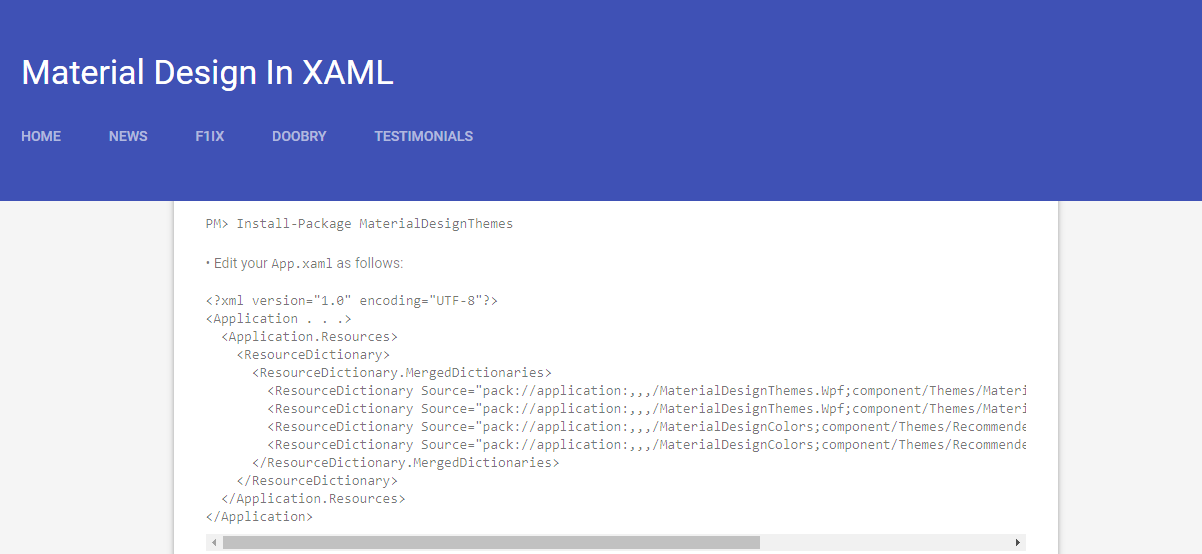


Рисунок 11.

В файле MainWindow.xaml мы можем прописать дизайн будущей программы (рисунок 12).

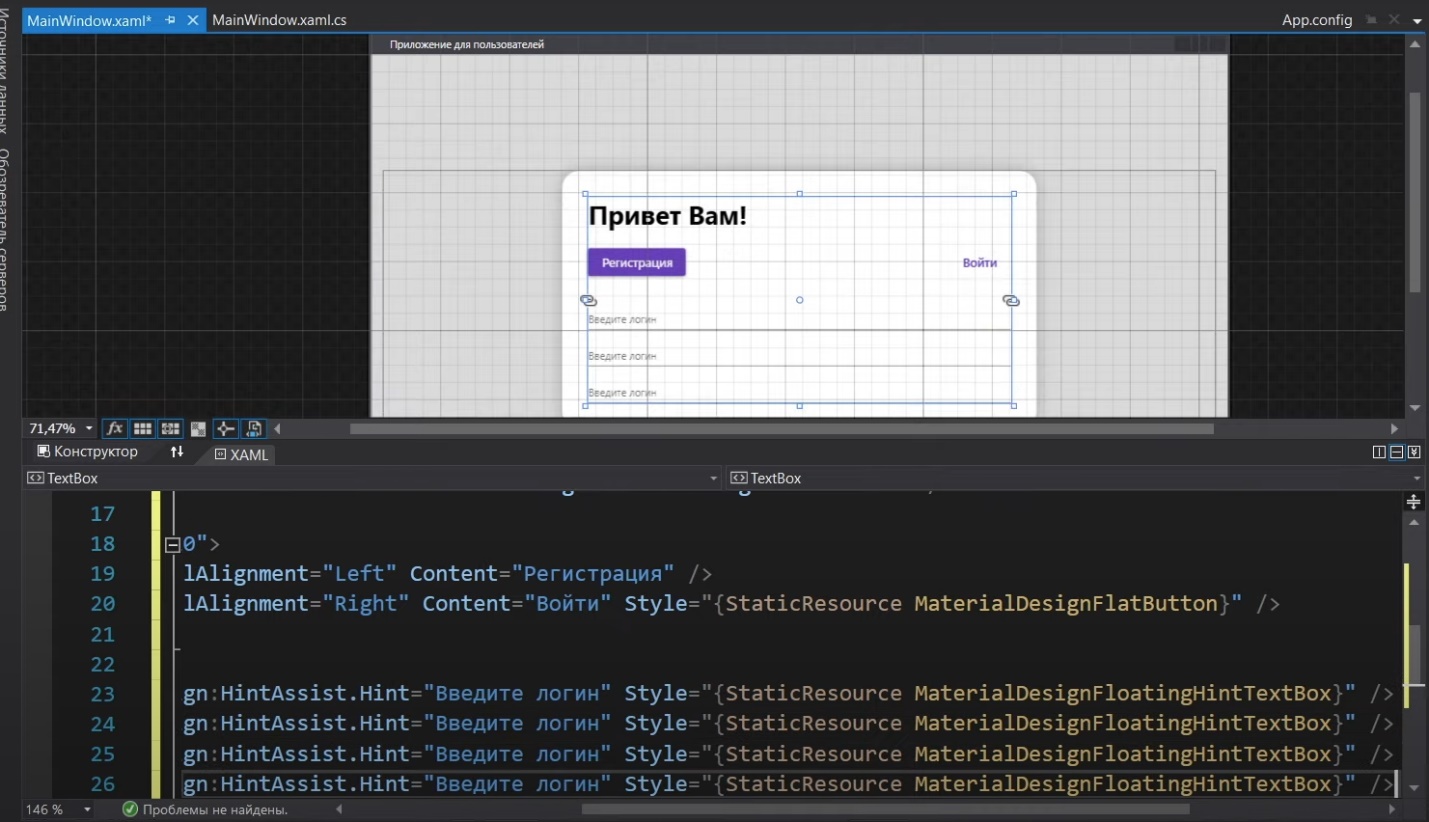


Рисунок 12.

Теперь приступим к созданию и подключению базы данных. Заранее скачанная с официального сайта программа Microsoft SQL Server Management Studio поможет нам в этом (рисунок 13).

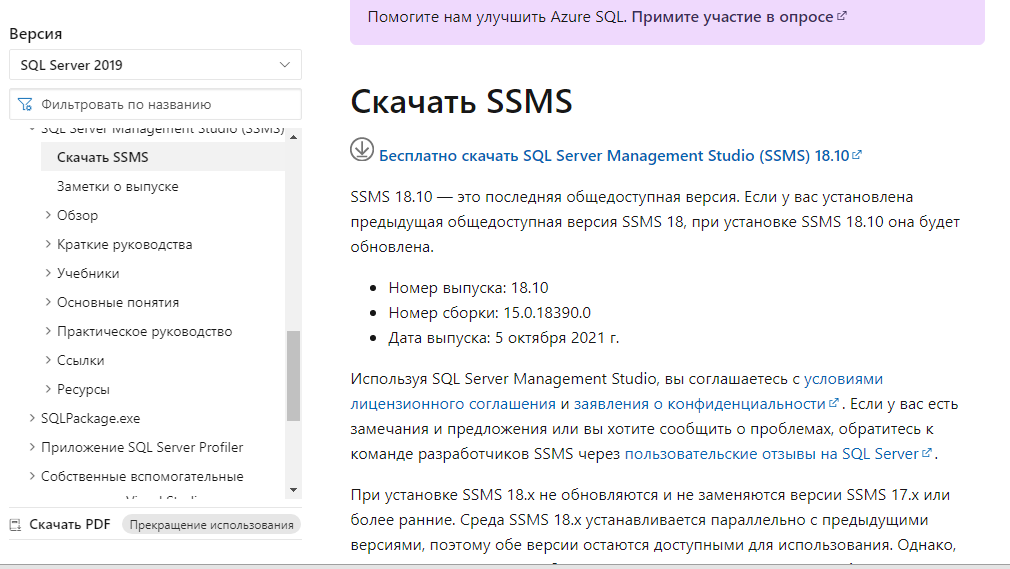


Рисунок 13.

После установки вводим запрашиваемые программой данные и создаем базу данных.

В созданной базе данных нужно создать таблицу «Users» (рисунок 14).

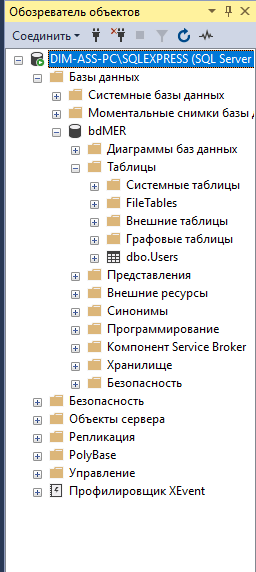


Рисунок 14.

Чтобы подключить базу к проекту нажимаем правой кнопкой мыши по файлу нашего проекта и нажимаем кнопку «Добавить», «Создать элемент». В окне «Добавление нового элемента» в поле «Поиск» вводим «edm» и добавляем Модель ADO.NET.EDM (рисунок 15.)

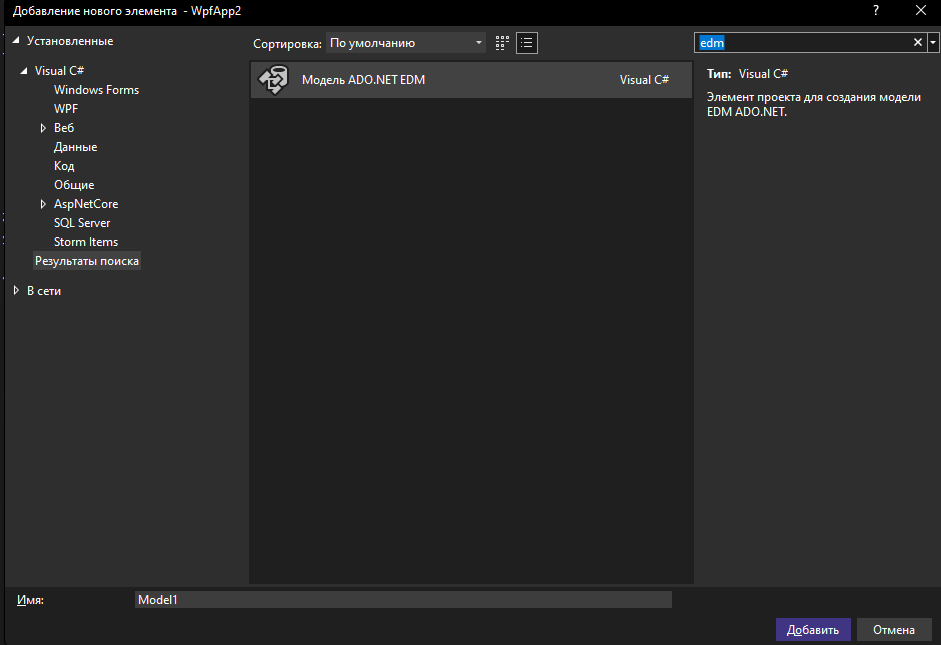


Рисунок 15.

Нам требуется ввести имя нашего сервера, выбрать нашу базу данных, далее выбрать таблицу «Users», после чего произойдет подключение (рисунок 16).

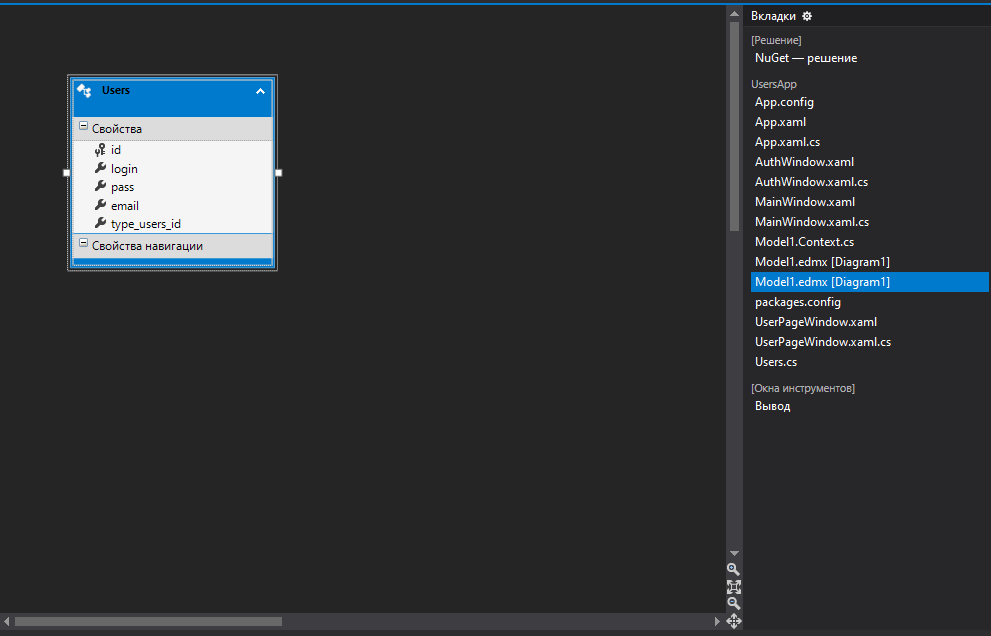


Рисунок 16.

Создаём класс-модель для работы с таблицей (рисунок 17).

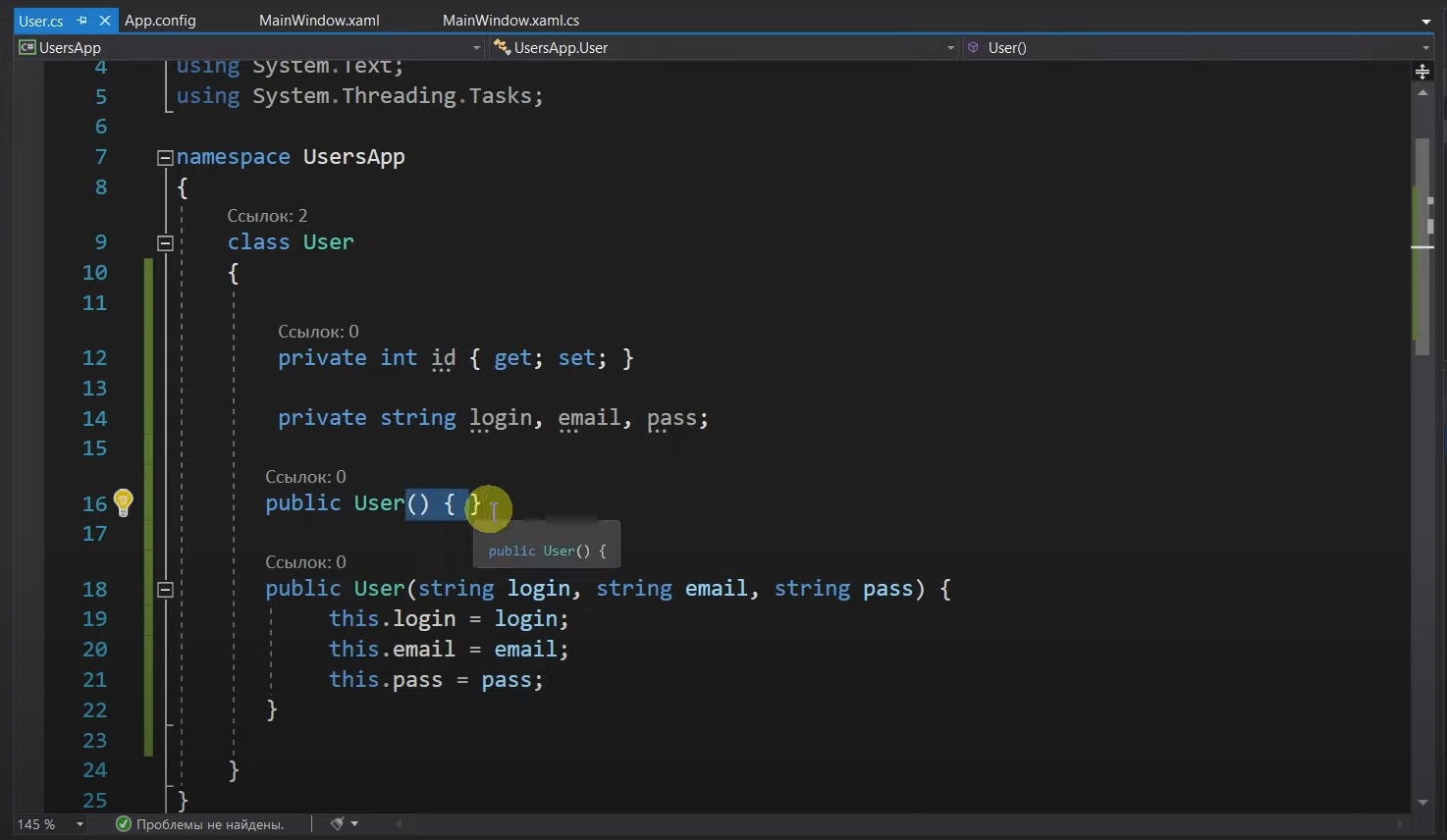


Рисунок 17.

Создаём класс для работы с базой данных (рисунок 18).

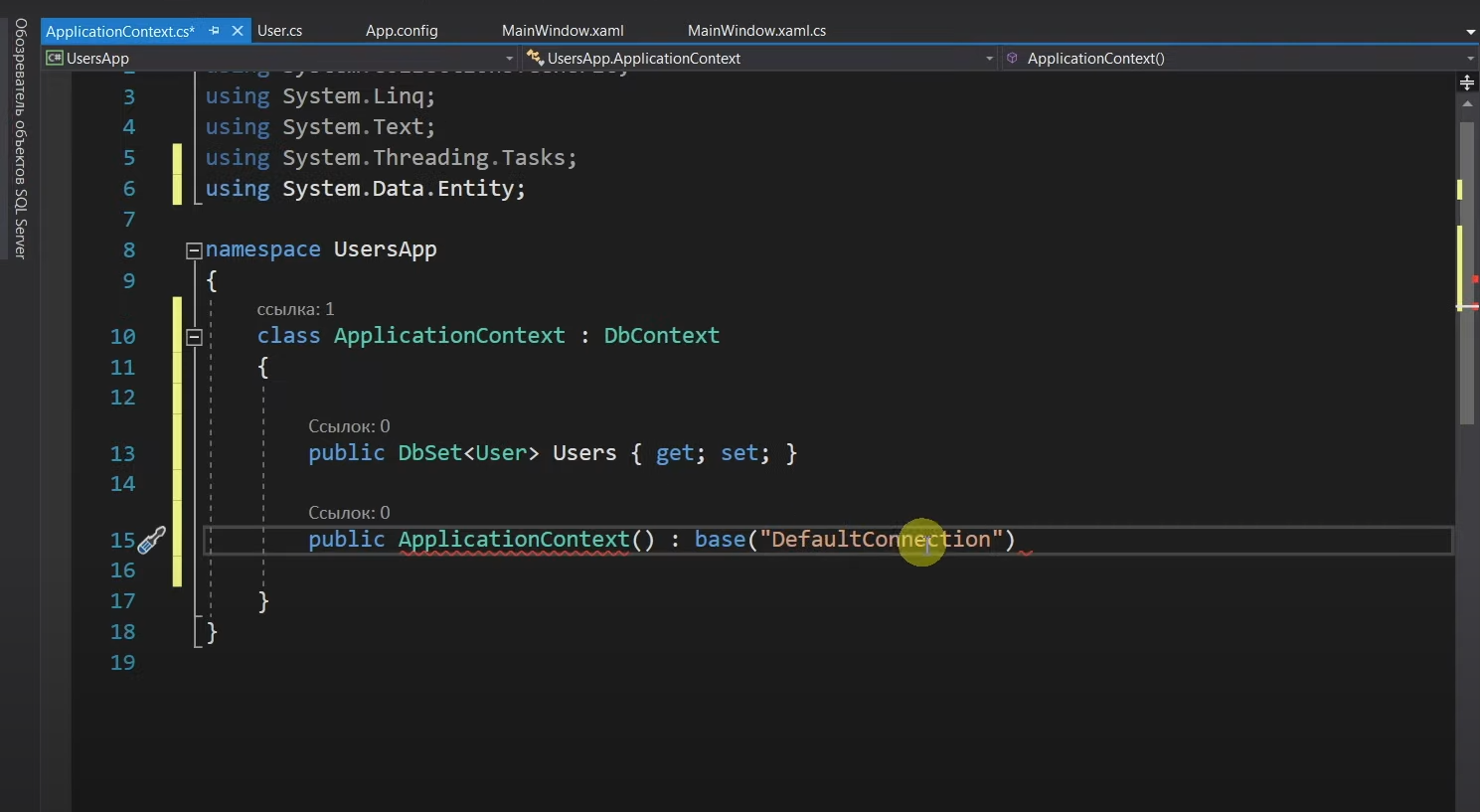


Рисунок 18.

Теперь добавим объект в БД (рисунок 19).

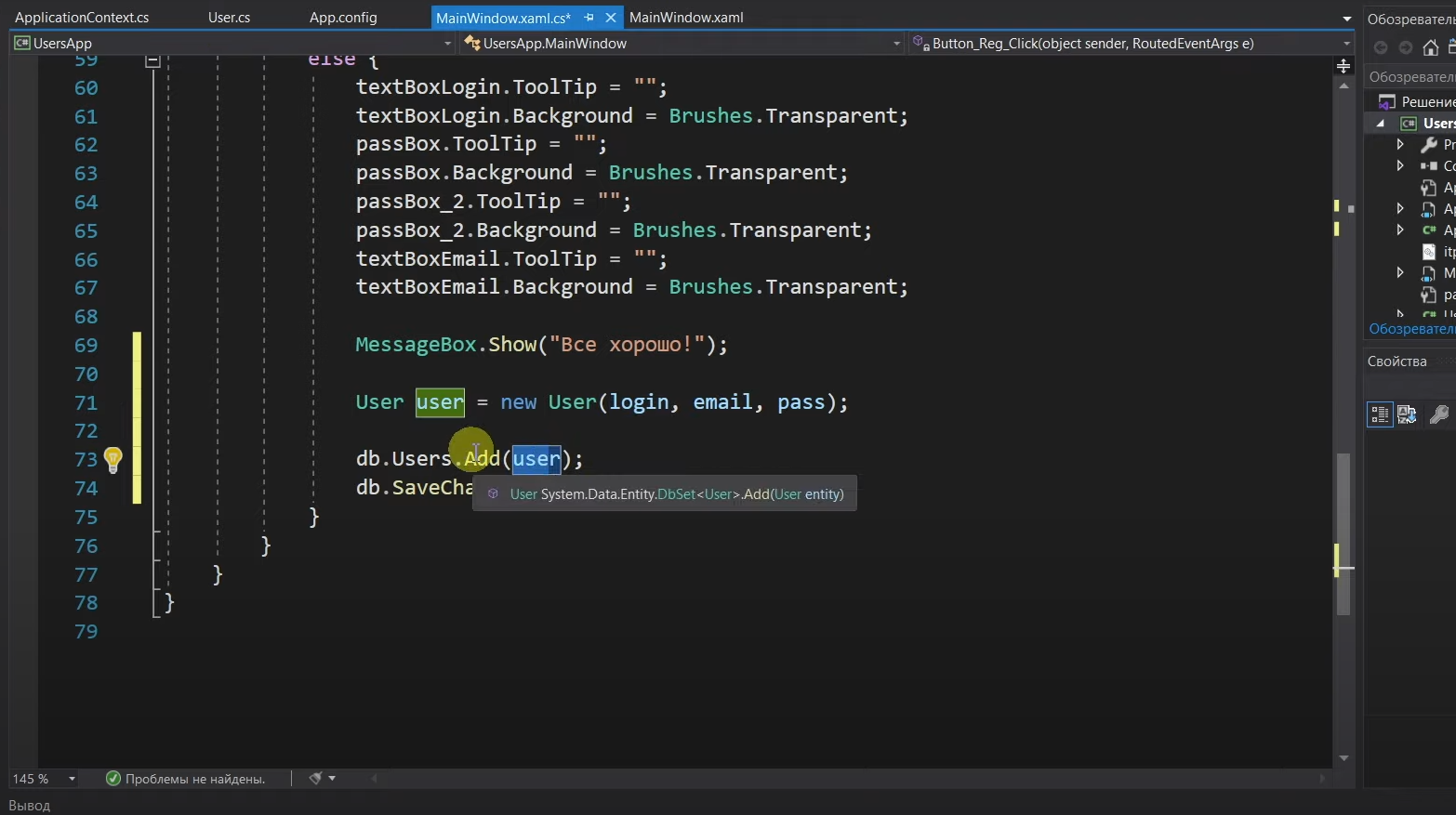


Рисунок 19.

Данные, которые мы добавили в таблицу нужно отобразить (рисунок 20).

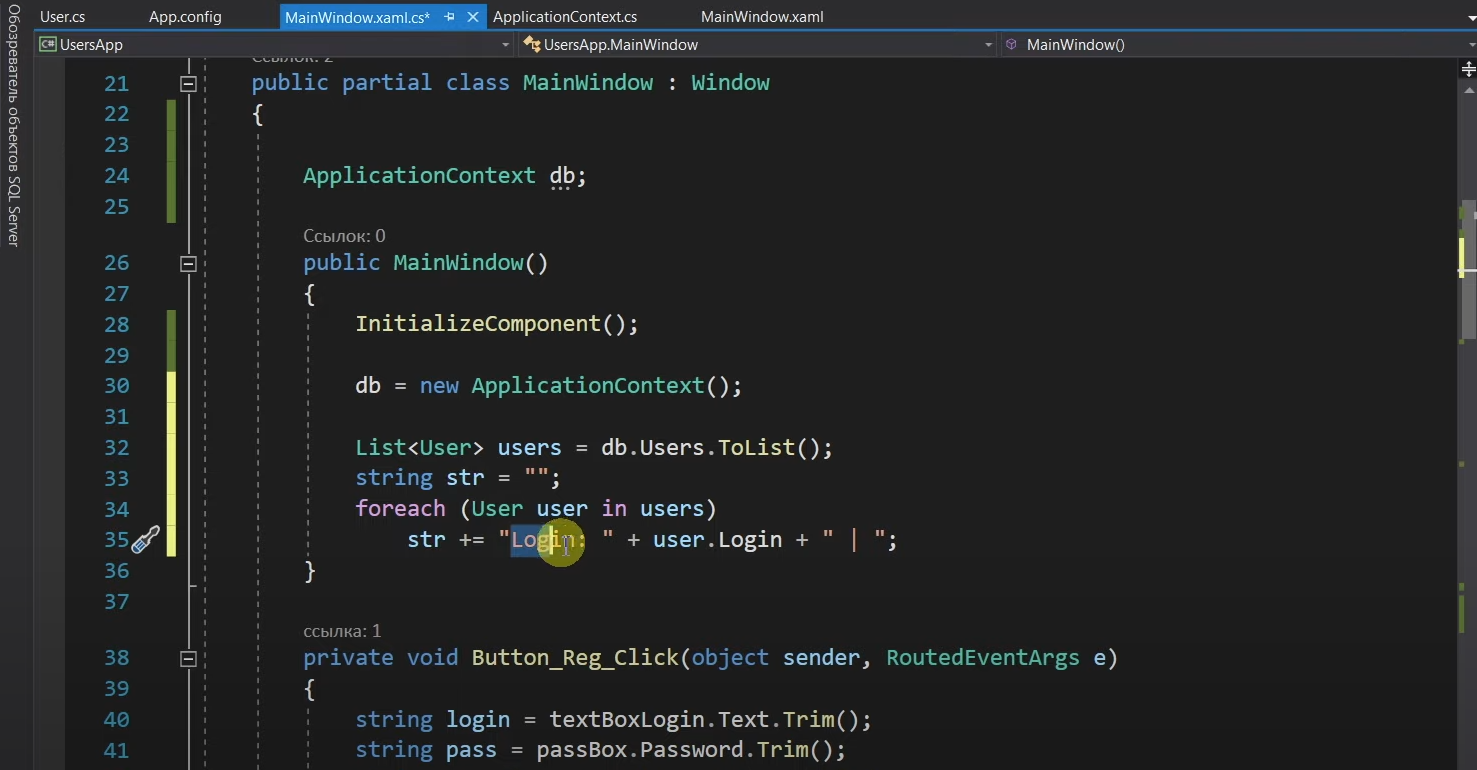


Рисунок 20.

Проверяем нашу программу (рисунок 21).

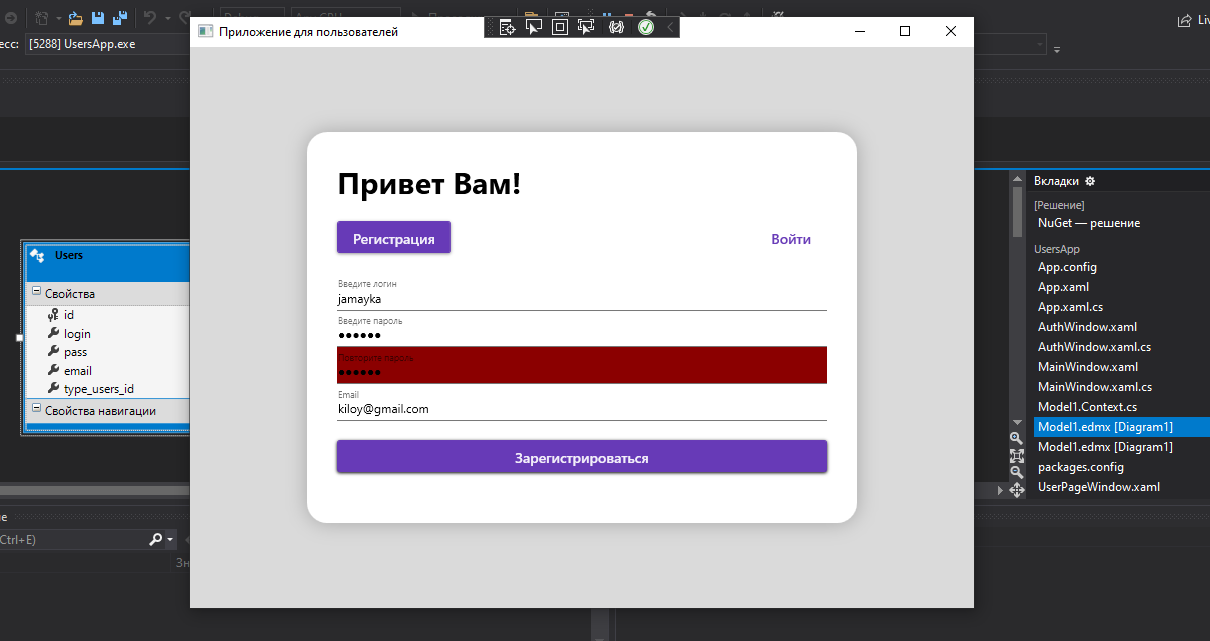


Рисунок 21.

После проверки нашей программы, переходим к окну с авторизацией. Создаём новое окно AuthWindow.xaml, в файле AuthWindow.xaml.cs прописываем код стиля, после чего выводится окно (рисунок 22).

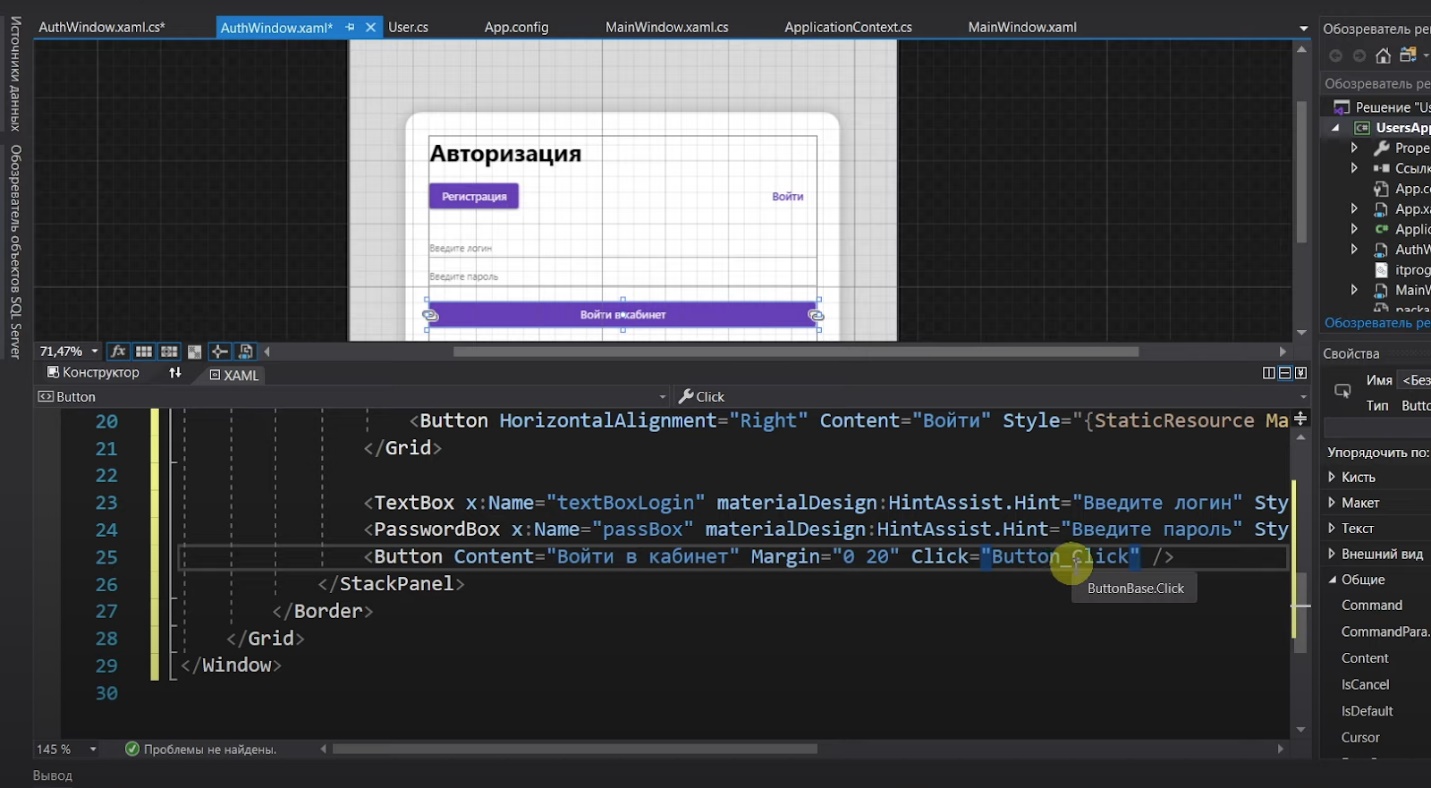


Рисунок 22.

Добавим проверку на введенные пользователем данные и данные из нашей Базы данных (рисунок 23).

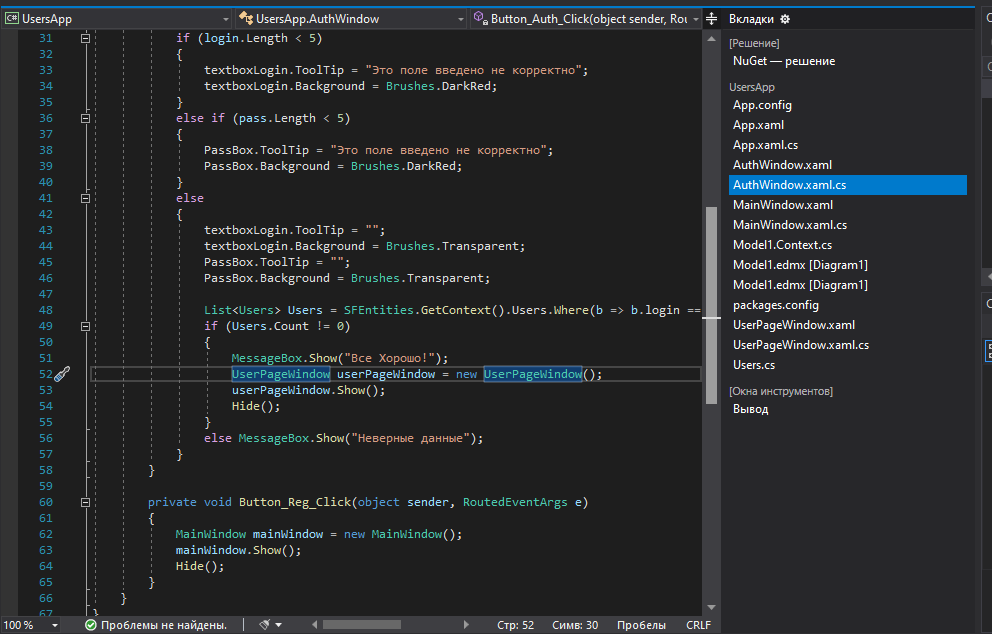


Рисунок 23.

Снова проверяем работу нашей программы (рисунок 24).

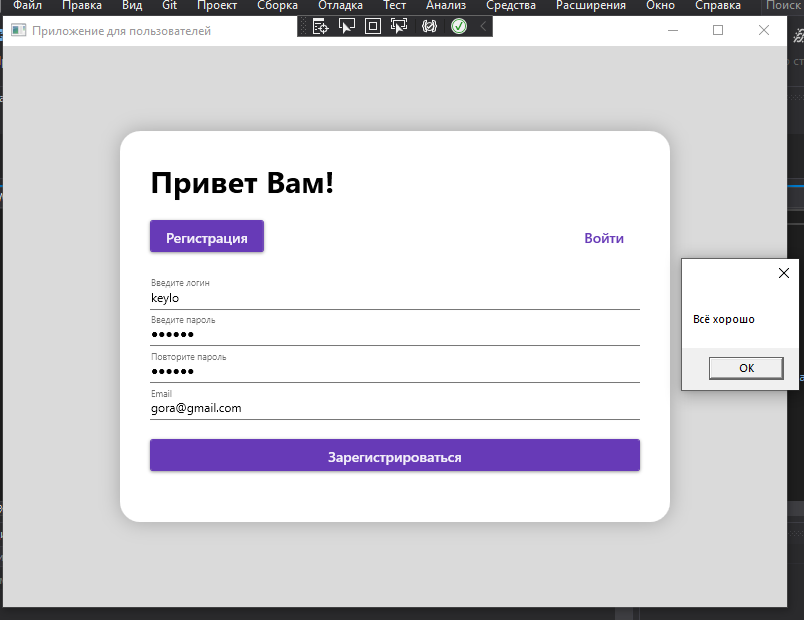


Рисунок 24.

На данном этапе переходим к созданию «Кабинета пользователя». Для этого нам понадобится настроить переход между страницами. Создаём само окно «Кабинет пользователя». Прописываем код вывода всех зарегистрированных пользователей и проверяем работу программы (рисунок 25).

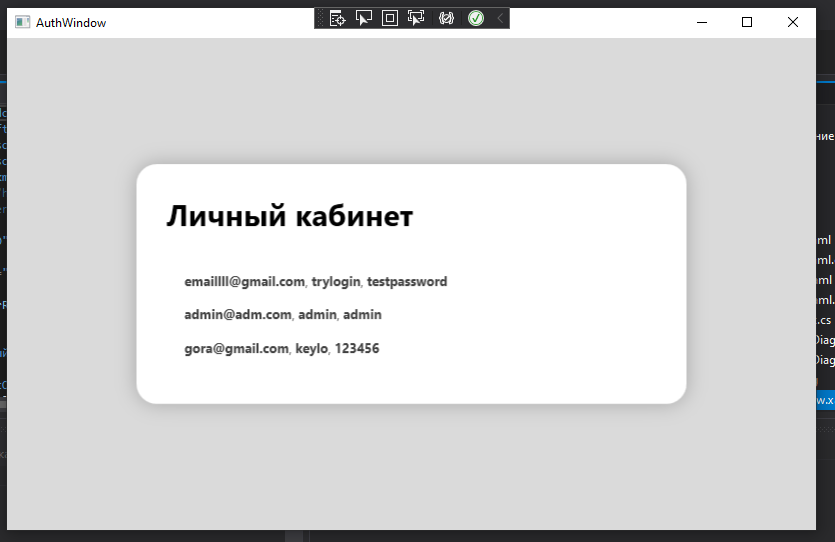


Рисунок 25.

Создание нашего WPF проекта закончено. Весь процесс был изначально затеян для получения навыков работы с базами данных, создания переходов между окнами, добавления новых пользователей в базу данных и тому подобное.

# 

# Выводы

По окончании учебной практики были решены следующие задачи:

1. Знакомство с литературой
2. Изучить библиотеку OpenCV
3. Реализация приложения на WPF
4. Прохождение курса «Основы языка Python»

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.

**Список литературы**

* 1. AcademiaThemes. Портал информатики для гиков. – URL: <https://tinyurl.com/mzz9k3ys> Дата обращения: 23.11.2021
  2. Мюллер Д. П. C# для чайников / Мюллер Д. П. – Москва; Москва, 2019, – 608 с
  3. 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
  4. 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
  5. <https://docs.google.com/presentation/d/17YdUQJrXn2dPckjSeDarp3v6mYDyaHhsLKEFe0GpqEk/edit#slide=id.g101651fe5da_0_13>
  6. <https://github.com/Sokolovafm?tab=repositories>
  7. <https://github.com/Sokolovafm/Educational-practice/tree/main/Opencv>
  8. <https://github.com/Sokolovafm/Educational-practice/tree/main/%D0%BA%D1%83%D1%80%D1%81%D1%8B>
  9. https://clck.ru/Z3caJ