**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ) Ликино-Дулевский политехнический колледж- филиал ГГТУ**

Наименование колледжа/техникума

**О Т Ч Ё Т**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

(код и наименование ПМ)

Обучающегося\_\_\_\_\_\_\_\_\_\_\_\_\_Толстова Тимура Артемьевича\_\_\_\_\_

Ф.И.О.

Курс \_\_\_\_3\_\_\_\_\_ группа \_\_\_ИСП.19А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Место практики Ликино-Дулевский политехнический колледж – филиал ГГТУ

Период практики с 24.11.2021 г. по 07.12.2021\_г., с 06.04.2022 г. по 19.04.2022\_г.\_

Руководители практики

от колледжа \_\_\_\_\_\_\_\_Гжегожевский Сергей Владимирович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кузьмина Елена Евгеньевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пронина Алла Юрьевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львов Олег Сергеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

**Содержание**

[1.Разработка программных модулей в среде MS Visual Studio 2019 4](#_Toc101314833)

[1.1 Изучение постановки задачи 4](#_Toc101314834)

[1.2. Описание области применения и пользователей приложения 4](#_Toc101314835)

[1.3. Проектирование интерфейса приложения 5](#_Toc101314836)

[1.4. Стандарты оформления кода языка C# 12](#_Toc101314837)

[1.5. Создание и использование стилей в технологии WPF 13](#_Toc101314838)

[1.6. Организация доступа к данным из приложения 15](#_Toc101314839)

[2. Разработка программных модулей в среде Python 20](#_Toc101314840)

[2.1. Установки и настройка среды PyCharm 20](#_Toc101314841)

[2.2. Особенности задания имен переменных и правила организации кода в Python 20](#_Toc101314842)

[2.3. Работа с циклами, строками и списками 21](#_Toc101314843)

[2.4. Именованные массивы – словари 24](#_Toc101314844)

[2.5. Картежи. Работа с файлами в Python 26](#_Toc101314845)

[3. Разработка мобильных приложений. 33](#_Toc101314846)

[3.1 Разработка дизайна мобильного приложения в среде Figma 33](#_Toc101314847)

[3.2 Работа в среде Android 36](#_Toc101314848)

[3.3 Структуру файла AndroidManifest.xml на основе файла вашего проекта. 47](#_Toc101314849)

[3.4 Структуру и назначение файла build.gradle 48](#_Toc101314850)

[3.5 Установка и запуск приложения 49](#_Toc101314851)

[3.6 Классы, используемые в проекте при разработке. 52](#_Toc101314852)

[3.7 Виды ошибок и предупреждений при разработке проекта. 52](#_Toc101314853)

[4. Тестирование программных модулей 53](#_Toc101314854)

[4.1. План тестирования 53](#_Toc101314855)

[4.2. Сценарии тестирования с результатами 53](#_Toc101314856)

[Test case #1 53](#_Toc101314857)

[Test case #2 53](#_Toc101314858)

[Test case #3: 54](#_Toc101314859)

[Test case #4: 54](#_Toc101314860)

[Test case #5: 55](#_Toc101314861)

[4.3. Предложения по улучшению функциональности 56](#_Toc101314862)

[5. Разработка технической документации 57](#_Toc101314863)

[5.1. Техническое задание 57](#_Toc101314864)

[5.1.1 Введение 57](#_Toc101314865)

[5.1.2 Назначение разработки 57](#_Toc101314866)

[5.1.3 Требования к программе или программному изделию 57](#_Toc101314867)

[5.1.4 Требования к программной документации 59](#_Toc101314868)

[5.2. Руководство программиста 59](#_Toc101314869)

[5.2.1 Назначения разработки 59](#_Toc101314870)

[5.2.2 Характеристики программы 59](#_Toc101314871)

[5.2.3 Обращения к программе 59](#_Toc101314872)

[5.2.4 Входные и выходные данные 60](#_Toc101314873)

[5.2.5 Разработка базы данных 60](#_Toc101314874)

[5.2.6 Разработка программного модуля 65](#_Toc101314875)

[5.3. Руководство пользователя 72](#_Toc101314876)

[5.3.1 Назначение программы 72](#_Toc101314877)

[5.3.2 Условия выполнения программы 72](#_Toc101314878)

[5.3.3 Выполнение программы 73](#_Toc101314879)

[5.3.4 Сообщения пользователю 78](#_Toc101314880)

[Заключение 80](#_Toc101314881)

[Список литературы 82](#_Toc101314882)

[Приложения 84](#_Toc101314883)

[Приложение1 84](#_Toc101314884)

[Приложение2 87](#_Toc101314885)

# 1.Разработка программных модулей в среде MS Visual Studio 2019

## **1.1 Изучение постановки задачи**

Программный продукт «Victorina» – программа, созданная для работы с базой данных по предметной области.

Автоматизация продукта «Victorina» - процесс внедрения программно-аппаратных комплексов для облегчения и автоматизации.

Приложение должно выполнять следующие функции:

• Вход в программу при запуске

• Просмотр информации из базы данных

• Обеспечить идентицифированным пользователям функционал по их ролям

• Корректно выводить данные из базы данных.

• Добавление новой информации

• Удаление информации

• Изменение информации

• Сортировка

• Переходы по страницам

## **1.2. Описание области применения и пользователей приложения**

**Пользователи:**

1. **Студент**

Роль «Студент» в приложении имеет доступ ко следующим функциям: «Прохождение тестов», «Просмотр своих результатов», «Управление личной информацией»

1. **Админ**

Роль «Админ» в приложении имеет доступ ко следующим функциям: «Просмотр чужих результатов», «Удаление/редактирование/добавление новых пользователе».

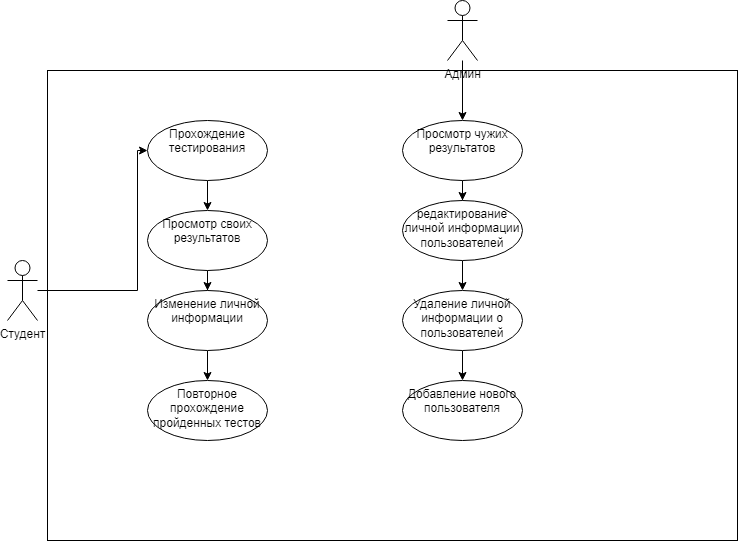
.

Рис №1. Сase-диаграмма

## **1.3. Проектирование интерфейса приложения**

**1.3.1. Правила и принципы разработки интерфейса**

**Главная задача — ясность**

Ясность — это первая и самая главная задача любого интерфейса. Чтобы интерфейс эффективно помогал людям добиваться своих целей, он должен обладать следующими характеристиками.

* Во-первых, он должен быть узнаваемым, а его назначение — очевидным для пользователя.
* Во-вторых, люди должны понимать, с чем они взаимодействуют через интерфейс.
* Наконец, процесс взаимодействия с интерфейсом должен быть предсказуемым. В интерфейс можно внести некую загадочность или элементы игры, но вот путаницы быть не должно.

Ясность рождает в пользователях уверенность и готовность продолжать работу с интерфейсом. Сто понятных экранов лучше, чем один беспорядочный.

**Ключевое назначение интерфейсов — взаимодействие**

Интерфейсы существуют, чтобы люди могли взаимодействовать с нашим миром. Через интерфейс мы можем прояснить, проиллюстрировать, дать возможность, показать взаимосвязь, объединить людей или разделить, управлять ожиданиями и давать доступ к услугам. Процесс проектирования интерфейсов — это не Искусство.

Интерфейсы — это не памятники самим себе. Интерфейсы выполняют конкретные задачи, их эффективность измерима.

Однако, они могут выходить за рамки чисто практического применения. Лучшие интерфейсы — те, которые вдохновляют, пробуждают чувства, удивляют и усиливают наш опыт общения с миром.

**Удержать внимание любой ценой**

Мы живем в мире отвлекающих факторов. В наше время даже почитать спокойно не получится — кто-то или что-то всегда будет отвлекать нас, перетягивать наше внимание. Внимание — это большая ценность.

**1.3.2. Разработка макета приложения**

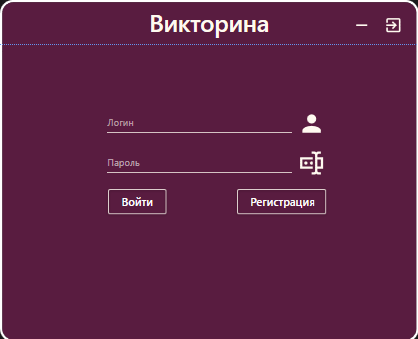


Рис №2 Страница авторизации



Рис №3 Страница пользовательского интерфейса



Рис №4 Страница админского интерфейса

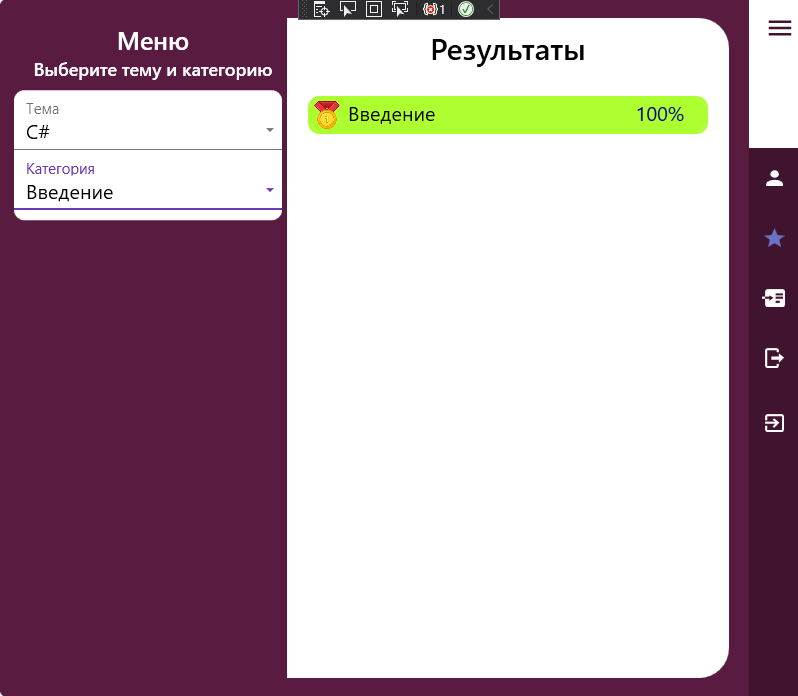


Рис №5 Страница результатов(Пользователь)

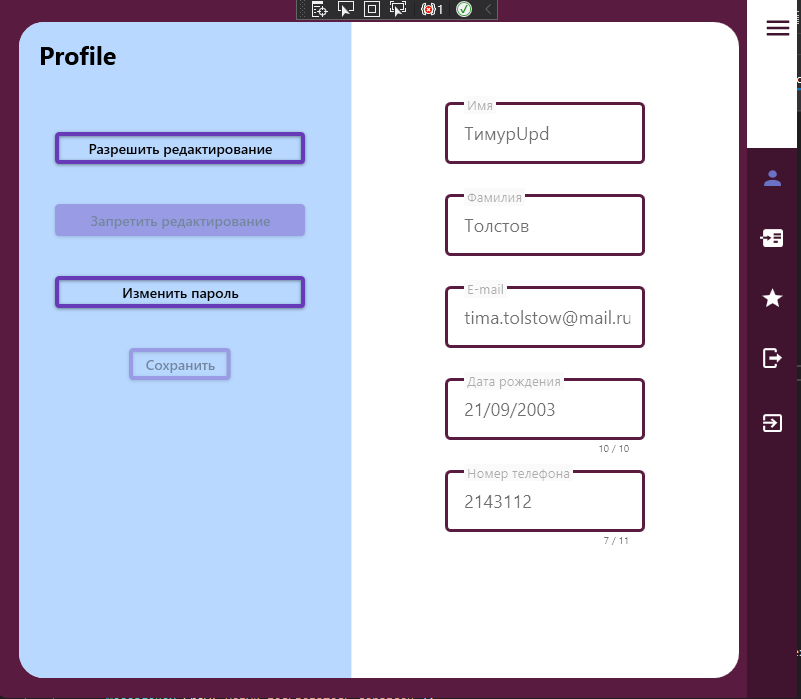


Рис №6 Страница профиля

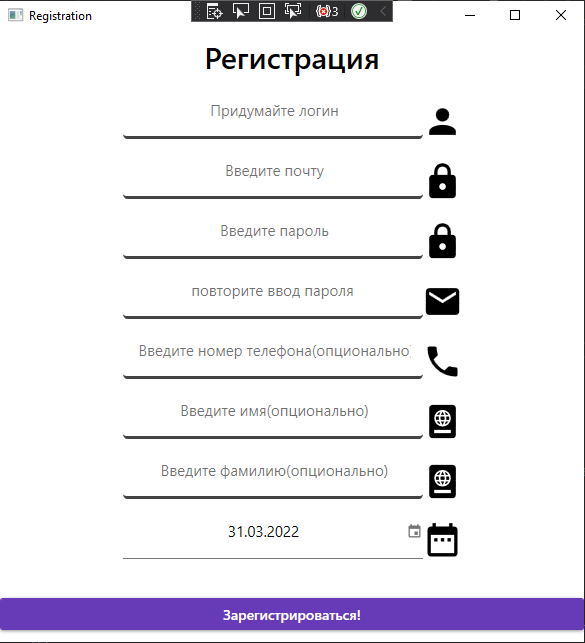


Рис №7 Страница регистрации

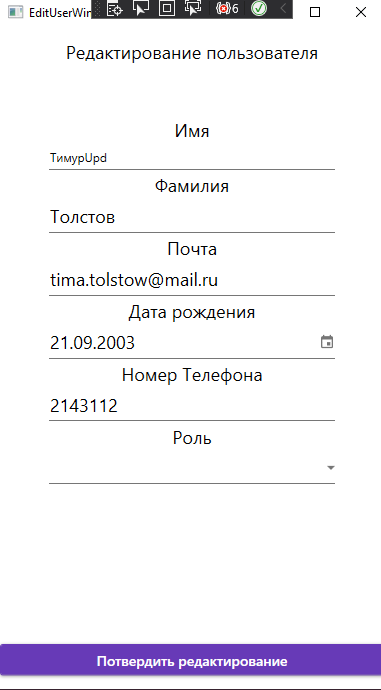


Рис №8 окно редактирование пользователя

**Окно добавления пользователя(Админ)**

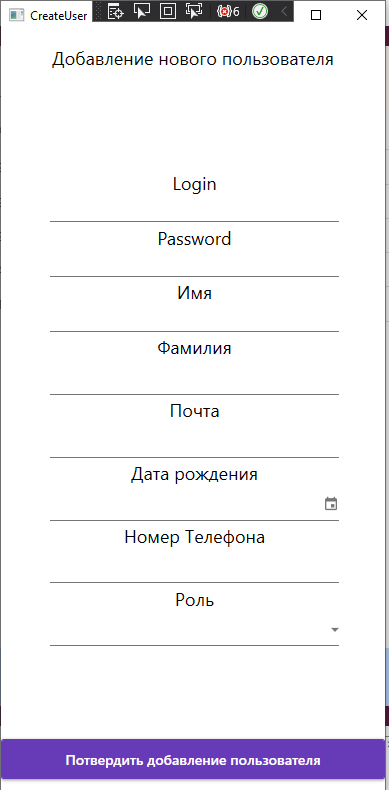


Рис №9 окно добавления пользователя

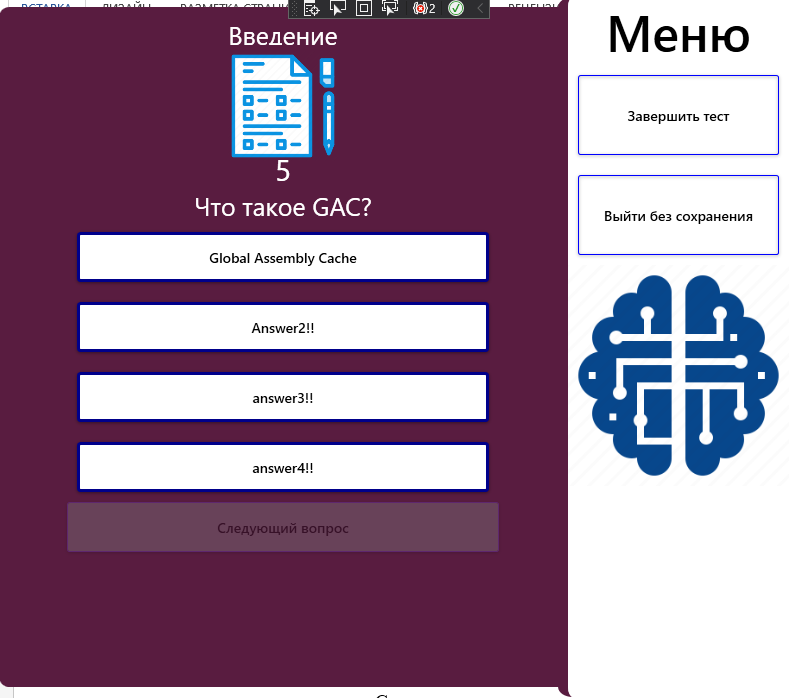


Рис №10 Страница прохождения теста

## **1.4. Стандарты оформления кода языка C#**

**1.4.1 Соглашения об именовании**

Camel case (иногда стилизованный под camelCase или camelCase, также известный как camel caps или более формально как медиальные столицы) - это практика написания фраз без пробелов или знаков препинания, указывающая на разделение слов с одной заглавной буквой и первое слово, начинающееся с любого случая. Распространенные примеры включают "iPhone" и "eBay". Он также иногда используется в онлайн-именах пользователей, таких как "JohnSmith", и для того, чтобы сделать многословные доменные имена более разборчивыми, например, при продвижении "EasyWidgetCompany.com.

Camel case часто используется в качестве соглашения об именах в компьютерном программировании, но является неоднозначным определением из-за необязательной капитализации первой буквы. Некоторые стили программирования предпочитают camel case с заглавной буквой, другие нет. Для ясности в этой статье называются две альтернативы верхний регистр верблюда (начальная заглавная буква, также известная как Начальные заглавные буквы, Начальные заглавные буквы, InitCaps или Pascal case) и нижний регистр верблюда (начальная строчная буква, также известная как падеж верблюда). Некоторые люди и организации, в частности Microsoft, используют этот термин camel case только для нижнего camel case, обозначая Pascal case для верхнего camel case.

**1.4.2. Таблица идентификаторов программных объектов**

|  |  |  |
| --- | --- | --- |
| Название обьекта | Тип обьекта | Назначение |
| Pages | | |
| AdminInterface | Страница | Интерфейс админа |
| AuthorizationPage | Страница | страница для авторизации |
| CategoryWnd | Окно | Окно категории |
| CreateNewUserWnd | Страница | Окно создание нового пользователя(админ) |
| EditUserWnd | Окно | Окно редактирования пользователя |
| ProfileUser | Страница | Страница профиля пользователя |
| RegistrationWnd | Окно | Окно регистрации |
| ResultPage | Страница | Страница результатов |
| TestForm | Страница | Страница прохождения тестов |
| UserInterface | Страница | Страница пользовательского интефейса |
| Services | | |
| СategoryServices | Класс | Запросы для коллекци категории |
| QuestionsServices | Класс | Запросы для коллекции вопросов |
| ResultsServices | Класс | Запросы для коллекции резульататов |
| TemaServices | Класс | Запросы для коллекции тем |
| TestItemServices | Класс | Запросы для коллекии тестов |
| UserService | Класс | Запросы для коллекции юзеров |

## **1.5. Создание и использование стилей в технологии WPF**

**1.5.1. Понятие и описание стиля**

Стили позволяют определить набор некоторых свойств и их значений, которые потом могут применяться к элементам в xaml. Стили хранятся в ресурсах и отделяют значения свойств элементов от пользовательского интерфейса. Также стили могут задавать некоторые аспекты поведения элементов с помощью триггеров. Аналогом стилей могут служить каскадные таблицы стилей (CSS), которые применяются в коде html на веб-страницах.

**1.5.2. Разработка стилей для элементов управления**

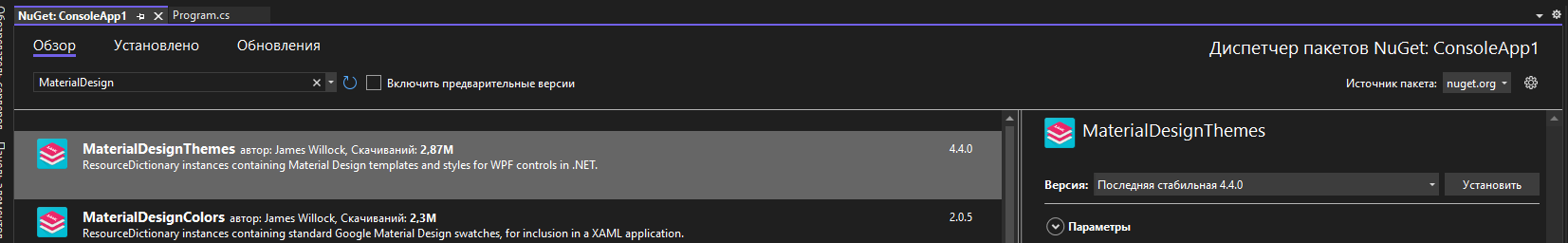


**1.5.3. Применение стилей для оформления интерфейса приложения**

Стили применяются автоматически при создании объектов в коде xaml. Вид и цвет объектов задается в app.xaml для всех объектов используя nu-get пакет MaterilDesign.

Подключение библиотеки MaterialdDesign

* 1. пкм, по названию проекта в обозревателе решений
  2. «Управление Nu-get пакетами»
  3. В поиске вводим MaterialDesign и справа нажимаем установить



* 1. Установка завершена

**1.5.4. Использование страниц Page в приложении**

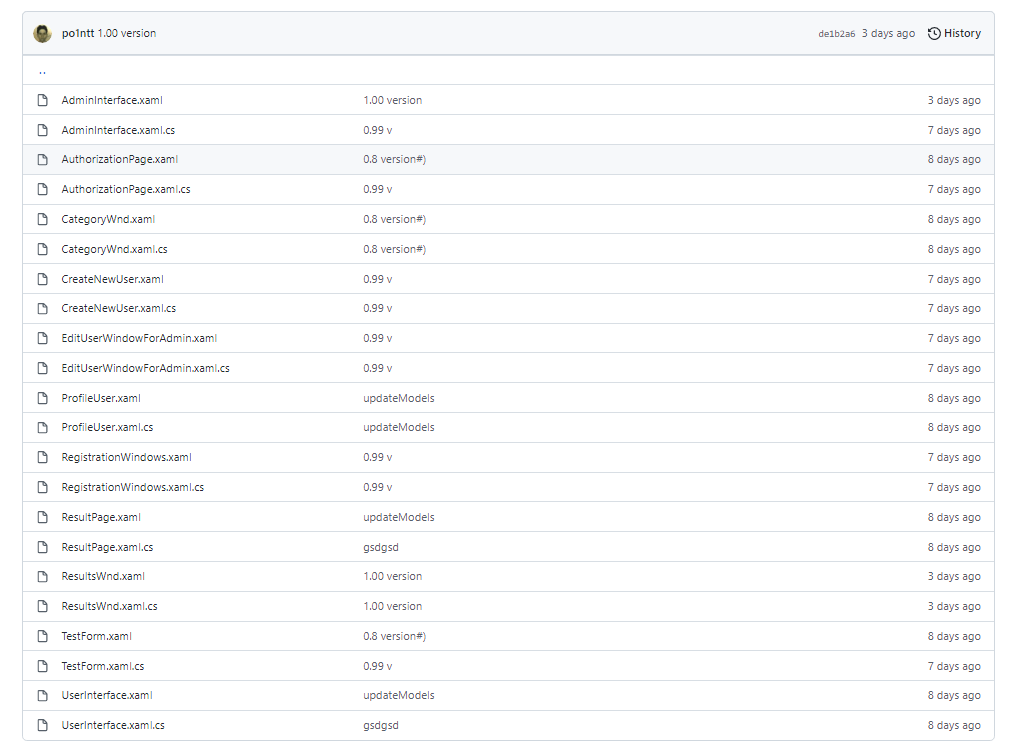


Рис №11. Папка Pages(В проекте)

**Класс навигации:**

class Frame

{

public static System.Windows.Controls.Frame FrameOBJ { get; set; }

}

**MainPage.cs:**

public MainWindow()

{

InitializeComponent();

ObjClas.Frame.FrameOBJ = MainFrame;

ObjClas.Frame.FrameOBJ.Navigate(new AuthorizationPage());

}

## **1.6. Организация доступа к данным из приложения**

**1.6.1. Использование Nu-get библиотеки FireBase.Database для связи с базой данных(теория)**

Для установки этой библиотеки нужно проделать следующие действия:

* 1. пкм, по названию проекта в обозревателе решений
  2. «Управление Nu-get пакетами»
  3. В поиске вводим firebase.database и справа нажимаем установить

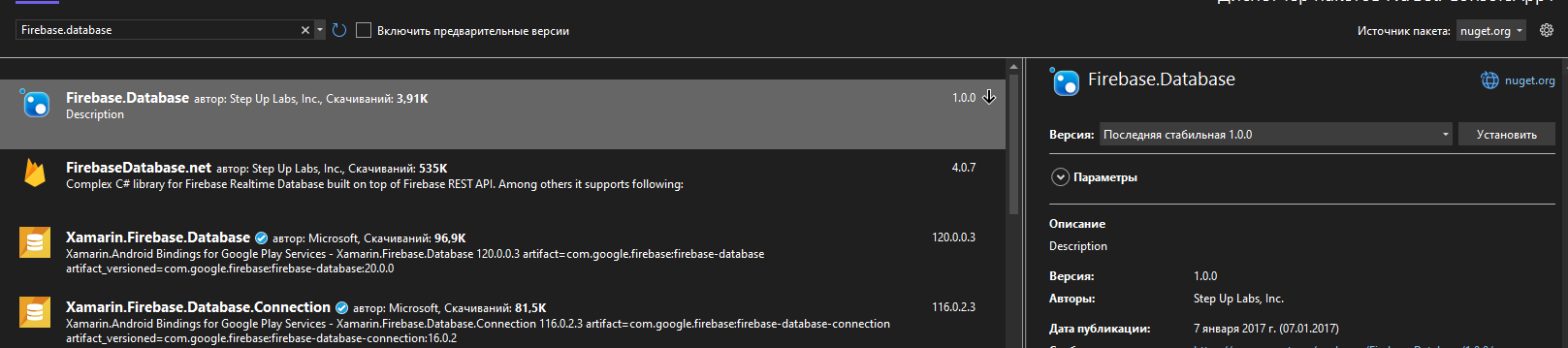


Рис №12. Завершение установки Firebase.Database

* 1. Установка завершена.

Как обращаться к базе данных через эту библиотеку описано ниже.

**1.6.2. Создание приложения с применением Entity Framework**

(подключение к БД, модули отображения данных, добавления, редактирования, поиска, фильтрации)

Разработка базы данных проходила в веб-сервисе google. Firebase.Console.

База данных имеет следующие коллекции:

* **Сategories:**

1. Cathegoryid;
2. СategoryName
3. TemaId
4. ImageUrl

* **ItemLearnCategory**

1. ImageUrlItem
2. NameItemLearn
3. Url\_toLearningSite
4. Id\_LearnCategory
5. Id\_ItemLearn

* **LearningCategory**

1. ImageLearnCat
2. NameLearnCategory
3. TemaId
4. IdLearnCat

* **Questions**

1. TextQuest
2. Id\_quest
3. Id\_test
4. Quest\_Answer1
5. Quest\_answer2
6. Quest\_answer3
7. QuestAnaswer4
8. Quest\_rightanswer

* **Results**

1. CategoryId
2. MedalImage
3. NameTestDone
4. ScorePercennt
5. TestID
6. User\_Login

* **Role**

1. Id\_role
2. NameRole

* **Tema**

1. NameTema
2. TemaId

* **Tests**

1. CategoryId
2. Description
3. Name
4. RightAnswerId
5. TestId

* **Users**

1. BirthDay
2. Email
3. Login
4. Name
5. Phone
6. Password
7. RoleId
8. SurName

В веб-сервисе база данных выглядит следующим образом:



Рис №13. БД

Подключения в базе данных осуществляется следующим программным кодом:

Using Firebase.Database;

FirebaseClient client;

public СathegoryServices()

{

client = new FirebaseClient("https://victorinaproject-default-rtdb.firebaseio.com/");

}

Примеры программное реализации функции Добавления/Редактирование/Удаления/Просмотр на примере одной из модели данных(Questions)

**1. Удаление:**

public async Task<bool> DeleteQuest(int Id\_Quest)

{

var keytodelete = (await client.Child("Questions").OnceAsync<Questions>()).FirstOrDefault(a => a.Object.id\_quest == Id\_Quest);

await client.Child("Questions").Child(keytodelete.Key).DeleteAsync();

return true;

}

**2. Добавление**

public async Task<bool> AddQuest(string textquest, string questanswer1, string questanswer2, string questanswer3, string questanswer4, string questrightanswer, TestsModel tests)

{

var questions = await GetQuestionsAsync();

await client.Child("Questions").PostAsync(new Questions()

{

TextQuest = textquest,

id\_quest = questions.Count + 1,

quest\_answer1 = questanswer1,

quest\_answer2 = questanswer2,

quest\_answer3 = questanswer3,

quest\_answer4 = questanswer4,

quest\_rightanswer = questrightanswer,

id\_test = tests.TestId

});

return true;

}

**3. Редактирование**

public async Task<bool> UpdateQuestions(int id\_Quest,string textquest, string questanswer1, string questanswer2, string questanswer3, string questanswer4, string questrightanswer, TestsModel tests)

{

var keyquest = (await client.Child("Questions")

.OnceAsync<Questions>())

.FirstOrDefault

(a => a.Object.id\_quest == id\_Quest);

Questions quest = new Questions() {TextQuest = textquest, id\_quest = id\_Quest, id\_test = tests.TestId, quest\_answer1 = questanswer1, quest\_answer2 = questanswer2, quest\_answer3 = questanswer3, quest\_answer4 = questanswer4, quest\_rightanswer = questrightanswer };

await client.Child("Questions")

.Child(keyquest.Key)

.PutAsync(quest);

return true;

}

**4. Просмотр**

public async Task<List<Questions>> GetQuestionsAsync()

{

var questions = (await client.Child("Questions").OnceAsync<Questions>()).Select(f => new Questions

{

id\_quest= f.Object.id\_quest,

id\_test = f.Object.id\_test,

quest\_answer1 = f.Object.quest\_answer1,

quest\_answer2 = f.Object.quest\_answer2,

quest\_answer3 = f.Object.quest\_answer3,

quest\_answer4 = f.Object.quest\_answer4,

TextQuest = f.Object.TextQuest,

quest\_rightanswer =f.Object.quest\_rightanswer

}).ToList();

return questions;

}

public async Task<ObservableCollection<Questions>> GetQuestionsAsyncBYTest(int test\_id)

{

var QuestionsByTest = new ObservableCollection<Questions>();

var items = (await GetQuestionsAsync()).Where(p => p.id\_test == test\_id).ToList();

foreach (var item in items)

{

QuestionsByTest.Add(item);

}

return QuestionsByTest;

}

Работа с другими коллекциями данных происходит аналогичным образом.

# 2. Разработка программных модулей в среде Python

## **2.1. Установки и настройка среды PyCharm**

Для установки PyCharm сперва нужно установить интерпретатор Python.

После выполнения предыдущего пункта, перейдите на сайт JetBrains https://www.jetbrains.com/pycharm/. PyCharm – это проект JetBrains.

Нажмите «Download». Произойдет перенаправление на страницу загрузки PyCharm.

Дальше необходимо выбрать версию PyCharm: Community или Professional. Professional — это платная версия с полным набором функций. Она идеально подходит для профессиональной разработки. Community бесплатная. Ею можно пользоваться благодаря набору базовых возможностей. После нажатия на кнопку «Загрузить» скачивание начнется автоматически.

Теперь необходимо запустить установку. Нажмите «Next».

Оставьте папку установки по умолчанию. Это папка C:\Program files(x86) \JetBrains\PyCharm Community Edition 2019.2. Нажмите «Next».

Поставьте галочки напротив обоих пунктов, если нужен ярлык для рабочего стола. Нажмите «Next».

Далее установщик попросит указать имя для отображения в стартовом меню. По умолчанию это JetBrains. Нажмите «Install».

Установка начнется автоматически. Дождитесь завершения.

После завершения установки поставьте галочку напротив «Run PyCharm Community Edition» для запуска, а затем «Finish».

## **2.2. Особенности задания имен переменных и правила организации кода в Python**

PEP8 — это стилевые правила для кода на языке Python.

* используйте 4 пробела для отступа и не используйте Tab, не смешивайте их.
* максимальная длина строки 79 символов; для разрыва строки используйте отступы или “бэкслеш”
* функции верхнего уровня и определения классов отделяйте двумя пустыми строками
* определения методов внутри класса отделяйте одной пустой строкой
* дополнительные пустые строки используйте для логической группировки методов
* кодировка файлов должна быть Latin-1, а в Питоне 3 и выше - utf-
* каждый импортируемый модуль с новой строки
* порядок импортов: стандартные модули, third-party модули, локальные модули приложения
* пустая строка между каждой группой импортов
* избегайте лишних пробелов внутри скобок; перед запятыми, точкой с запятой и двоеточиями
* избегайте лишних пробелов перед скобками с аргументами функций и скобками с индексами
* избегайте больше, чем одного пробела между операторами (= - + отд.)
* используйте одиночный пробел для выделения операторов
* избегайте пробелов вокруг ‘=’, который используется для параметров по-умолчанию и keyword аргументов
* обновляйте комментарии вместе с кодом
* пишите комментарии по-английски
* избегайте имён из одиночных l, O, I
* для модулей и пакетов используйте короткие все строчные имена
* для классов используйте Pascal Case стандарт
* для функций используйте слова\_с\_подчёркиваниями (или смешанныйРегистр если так получилось исторически)
* для методов используйте слова\_с\_подчёркиваниями имена
* для внутренних методов и переменных \_подчёркивание\_перед\_словом
* для проверки на None используйте is или is not, не используйте операторы сравнения
* используйте исключения, основанные на классах – наследованные от Exception
* используйте методы класса string вместо строковых функций
* используйте ‘‘. startswith () и ‘‘. endswith () вместо вырезки из строк для проверки префиксов и суффиксов
* используйте isinstance () для проверки типа
* не сравнивайте булевы переменные с True и False

## **2.3. Работа с циклами, строками и списками**

**2.3.1 Выполнений задач на ветвление.**

Код:

price = float(input('Введите цену конфет: '))  
for i in range(1,11):  
 print(f'Цена за {i}кг конфет {i \* price}')

Результат:

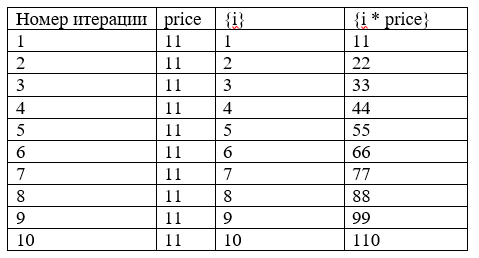


Рис №14. Ветвление(Python)

**2.3.2 Отладка программного кода встроенными методами IDE (теория)**

**PyCharm и отладка**

В то время как многие из нас в первую очередь боятся писать коды, некоторые другие находят отладку более сложной задачей. В Python это требует больше времени, поскольку вы не знаете, в чем ошибка. К счастью, с PyCharm и его значительными функциями отладки пользователи могут получить уникальный опыт отладки при запуске своих скриптов Python.

**Точки останова и как их разместить**

Точки останова - это маркеры, которые помогают вам приостановить выполнение вашей программы в любой конкретной точке, чтобы вы могли изучить ошибки и поведение этой конкретной строки. После отметки точка останова останется в вашем коде, если вы явно не удалите ее. Чтобы разместить их, щелкните левый желоб рядом со строкой, на которой вы хотите приостановить приложение.

**Как начать отладку?**

Когда вы закончите добавлять точки останова, ваш код готов к отладке. Отладчик запустит всю программу, за исключением строк, которые вы явно не отметили. Чтобы начать сеанс отладчика, щелкните значок ‘зеленыйигратьикона' слева. Когда появится всплывающее меню, выберите параметр ‘ОтлаживатьРешатель. (Решатель, потому что это имя файла.)

Когда запускается отладчик, ваша программа начинает выполняться на вкладке консоли окна отладки. Поскольку программе требуются входные значения, экран отладчика попросит вас сделать это ниже:

В первой точке останова отладчик приостанавливает выполнение программы, выделяя строку синим цветом:

Чтобы возобновить отладку, щелкните «зеленый значок воспроизведения» в верхней части панели инструментов вкладки «Отладчик».

**Концепция встроенной отладки**

Встроенная отладка позволяет просматривать значение каждой переменной, чтобы сделать процесс отладки более простым и понятным. Эта функция по умолчанию присутствует в PyCharm.

**Наблюдение за вашими переменными в PyCharm**

Для более глубокого понимания вашего кода PyCharm также позволяет вам наблюдать за переменными. Нажмите кнопку в верхней части панели инструментов вкладки «Переменные». Затем введите имя переменной, за которой вы хотите следить во время выполнения кода.

**2.3.3 Выполнение задач на строки**

Код:

str = input('Введите строку: ')  
str1 = ''  
if(len(str) > 10):  
 str1 += str[-1]  
 str1 += str[0]  
 str1 += str[-3]  
 str1 += str[2]  
 print(str1)  
else:  
 print('Длина ст роки меньше 10')

Результат:



Рис №15. Строка(Python)

**2.3.4 Выполнение задач на списки**

Код:

mas = []  
n = int(input('Введите размер массива: '))  
for i in range(n):  
 mas.append(int(input(f'Введите {i + 1}й элемент массива: ')))  
print(f'Максимальный элемент массива: {max(mas)}')  
print('Массив в обратном порядке')  
print(\*reversed(mas))

Результат:

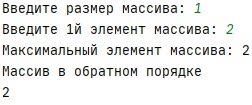


Рис №16. Cписки (Python)

## **2.4. Именованные массивы – словари**

**2.4.1 Понятие и назначение словарей**

Словарь — неупорядоченная структура данных, которая позволяет хранить пары «ключ — значение».

Вот пример словаря на Python:

dictionary = {'персона': 'человек',

'марафон': 'гонка бегунов длиной около 26 миль',

'противостоять': 'оставаться сильным, несмотря на давление',

'бежать': 'двигаться со скоростью'}

Данный словарь использует строки в качестве ключей, однако ключом может являться в принципе любой неизменяемый тип данных. Значением же конкретного ключа может быть что угодно. Вот ещё один пример словаря, где ключами являются числа, а значениями — строки:

gender\_dict = {0: 'муж',

1: 'жен'}

Важное уточнение: если вы попробуете использовать изменяемый тип данных в качестве ключа, то получите ошибку:

dictionary = {(1, 2.0): 'кортежи могут быть ключами',

1: 'целые числа могут быть ключами',

'бежать': 'строки тоже',

['носок', 1, 2.0]: 'а списки не могут'}

Прим. перев. На самом деле проблема не с изменяемыми, а с некэшируемыми типами данных, но обычно это одно и то же.

**2.4.2 Использование методов**

#Создаем словарь  
company = {'Ит-отдел': 12, 'Бухгалтерия': 5, 'Менеджеры': 20, 'Директора': 4, 'Уборщики': 10}  
print("-------------------------------------------")  
  
#Выводим число сотрудников по названию подразделения  
print("Выводим число сотрудников по названию подразделения")  
key = input('Введите название отдела: ')  
if key in company.keys():  
 print(f'Кол-во сотрудников: {company[key]}')  
else:  
 print('Такого отдела нет!')  
print("-------------------------------------------")  
  
#Добавляем новые подразделения  
#Изменяем число  
company['Директора'] = 1  
company['Инженеры'] = 9  
company['Чертежники'] = 6  
  
#Выводим данные о подразделениях  
print("Выводим данные о подразделениях")  
print(company)  
print("-------------------------------------------")  
  
#Преобразовываем словарь в список  
print("Преобразовываем словарь в список")  
company\_list = list(company.items())  
for M in company\_list:  
 print(\*M)

**2.4.2 Разработка программы по обработке коллекций**

Необходимо написать программу, которая будет хранить информацию о товарах на складе.

Ключом словаря должен являться id записи.

Реализуйте следующий функционал:

• Добавление новой записи (id должно быть уникальным).

• Удаление записи по её id (учитывать, что такой номер может отсутствовать в словаре).

• Вывести все записи о товаре на указанном складе.

• Вывести всю информацию о товаре с ценой меньше указанной.

• Вывести всю информацию о товаре с указанной ед. изм.

• Вывести среднюю цену товары по складам в формате: «название склада: сред. Цена»

• Вывести всю информацию об указанном товаре.

• Вывести все данные записи и сумму по каждой записи. Сумма = Цена\*кол-во

• Любая ваша функция.

При реализации данного функционала необходимо реализовать диалог с пользователем. (например, каждой функции можно присвоить номер, когда пользователь его введет, функция будет выполнена).

Код: [См.Приложение1](#_Приложение1)

**2.4.3 Разработка интерфейса программы**

По указанной предметной области был разработан интерфейс приложения, позволяющего делать совершать учет склада. Рисунок 46 демонстрирует возможный интерфейс приложения, а именно страницу добавления товара на определенный склад. Слева можно видеть меню приложения, а справа рабочую область, которая изменяется в зависимости от выбора пункта меню. На данном рисунке изображена форма добавления, на которой пользователю будет нужно заполнить поля, назначение которых указано в блоке слева.

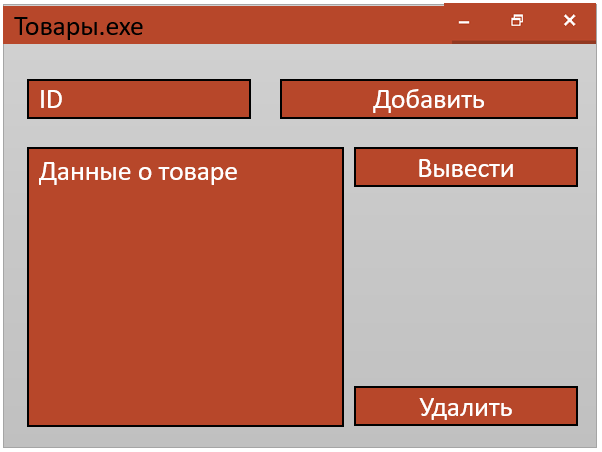


Рис №17. Интерфейс (Python)

## **2.5. Картежи. Работа с файлами в Python**

**2.5.1 Понятие картежей и их назначение**

Отличие кортежа от списка заключается в том, что кортеж защищен от изменений, как намеренных, так и случайных.

**Меньший размер. К примеру:**

>>> a = (1, 2, 3, 4, 5, 6)

>>> b = [1, 2, 3, 4, 5, 6]

>>> a.\_\_sizeof\_\_()

36

>>> b.\_\_sizeof\_\_()

44

**Возможность использовать кортежи в качестве ключей словаря:**

>>> d = {(1, 1, 1) : 1}

>>> d

{(1, 1, 1): 1}

>>> d = {[1, 1, 1] : 1}

Traceback (most recent call last):

File "", line 1, in

d = {[1, 1, 1] : 1}

TypeError: unhashable type: 'list'

**Как работать с кортежами?**

Работа с кортежами не сильно отличается от работы со списками.

Создаем пустой кортеж:

>>>

>>> a = tuple() # С помощью встроенной функции tuple()

>>> a

()

>>> a = () # С помощью литерала кортежа

>>> a

()

>>>

Создаем кортеж из одного элемента:

>>>

>>> a = ('s')

>>> a

's'

Получилась строка. Получить кортеж можно следующим образом:

>>>

>>> a = ('s', )

>>> a

('s',)

Все дело - в запятой. Сами по себе скобки ничего не значат, точнее, значат то, что внутри них находится одна инструкция, которая может быть отделена пробелами, переносом строк и т.д. Также, кортеж можно создать и так:

>>> a = 's',

>>> a

('s',)

Создать кортеж из итерируемого объекта можно с помощью все той же пресловутой функции tuple()

>>> a = tuple('hello, world!')

>>> a

('h', 'e', 'l', 'l', 'o', ',', ' ', 'w', 'o', 'r', 'l', 'd', '!')

**Операции с кортежами**

Все операции над списками, не изменяющие список (сложение, умножение на число, методы index () и count () и некоторые другие операции). Можно также по-разному менять элементы местами и так далее.

**2.5.2 Основы работы с внешними файлами в Python**

CSV (comma-separated value) — это формат представления табличных данных (например, это могут быть данные из таблицы или данные из БД).

В этом формате каждая строка файла — это строка таблицы. Несмотря на название формата, разделителем может быть не только запятая.

И хотя у форматов с другим разделителем может быть и собственное название, например, TSV (tab, separated, values), тем не менее, под форматом CSV понимают, как правило, любые разделители.

Пример файла в формате CSV (sw\_data.csv):

hostname, vendor, model, location

sw1, Cisco,3750, London

sw2, Cisco,3850, Liverpool

sw3, Cisco,3650, Liverpool

sw4, Cisco,3650, London

В стандартной библиотеке Python есть модуль csv, который позволяет работать с файлами в CSV формате.

**Чтение**

Пример чтения файла в формате CSV (файл csv\_read.py):

Вывод будет таким:

$ python csv\_read.py

['hostname', 'vendor', 'model', 'location']

['sw1', 'Cisco', '3750', 'London']

['sw2', 'Cisco', '3850', 'Liverpool']

['sw3', 'Cisco', '3650', 'Liverpool']

['sw4', 'Cisco', '3650', 'London']

В первом списке находятся названия столбцов, а в остальных соответствующие значения.

Обратите внимание, что сам csv.reader возвращает итератор:

In [1]: import csv

In [2]: with open('sw\_data.csv') as f:

...: reader = csv.reader(f)

...: print(reader)

...:

<\_csv.reader object at 0x10385b050>

При необходимости его можно превратить в список таким образом:

In [3]: with open('sw\_data.csv') as f:

...: reader = csv.reader(f)

...: print(list(reader))

...:

[['hostname', 'vendor', 'model', 'location'], ['sw1', 'Cisco', '3750', 'London'], ['sw2', 'Cisco', '3850', 'Liverpool'], ['sw3', 'Cisco', '3650', 'Liverpool'], ['sw4', 'Cisco', '3650', 'London']]

Чаще всего заголовки столбцов удобней получить отдельным объектом. Это можно сделать таким образом (файл csv\_read\_headers.py):

Иногда в результате обработки гораздо удобней получить словари, в которых ключи — это названия столбцов, а значения - значения столбцов.

Для этого в модуле есть DictReader (файл csv\_read\_dict.py):

Вывод будет таким:

$ python csv\_read\_dict.py

OrderedDict([('hostname', 'sw1'), ('vendor', 'Cisco'), ('model', '3750'), ('location', 'London')])

sw1 3750

OrderedDict([('hostname', 'sw2'), ('vendor', 'Cisco'), ('model', '3850'), ('location', 'Liverpool')])

sw2 3850

OrderedDict([('hostname', 'sw3'), ('vendor', 'Cisco'), ('model', '3650'), ('location', 'Liverpool')])

sw3 3650

OrderedDict([('hostname', 'sw4'), ('vendor', 'Cisco'), ('model', '3650'), ('location', 'London')])

sw4 3650

DictReader создает не стандартные словари Python, а упорядоченные словари. За счет этого порядок элементов соответствует порядку столбцов в CSV-файле.

Примечание

До Python 3.6 возвращались обычные словари, а не упорядоченные.

В остальном с упорядоченными словарями можно работать, используя те же методы, что и в обычных словарях.

Запись Аналогичным образом с помощью модуля csv можно и записать файл в формате CSV (файл csv\_write.py):

В примере выше строки из списка сначала записываются в файл, а затем содержимое файла выводится на стандартный поток вывода.

Вывод будет таким:

$ python csv\_write.py

hostname,vendor,model,location

sw1,Cisco,3750,"London, Best str"

sw2,Cisco,3850,"Liverpool, Better str"

sw3,Cisco,3650,"Liverpool, Better str"

sw4,Cisco,3650,"London, Best str"

Обратите внимание на интересную особенность: строки в последнем столбце взяты в кавычки, а остальные значения - нет.

Так получилось из-за того, что во всех строках последнего столбца есть запятая. И кавычки указывают на то, что именно является целой строкой. Когда запятая находится в кавычках, модуль csv не воспринимает её как разделитель.

Иногда лучше, чтобы все строки были в кавычках. Конечно, в данном случае достаточно простой пример, но когда в строках больше значений, то кавычки позволяют указать, где начинается и заканчивается значение.

Модуль csv позволяет управлять этим. Для того, чтобы все строки записывались в CSV-файл с кавычками, надо изменить скрипт таким образом (файл csv\_write\_quoting.py):

Теперь вывод будет таким:

$ python csv\_write\_quoting.py

"hostname","vendor","model","location"

"sw1","Cisco","3750","London, Best str"

"sw2","Cisco","3850","Liverpool, Better str"

"sw3","Cisco","3650","Liverpool, Better str"

"sw4","Cisco","3650","London, Best str"

Теперь все значения с кавычками. И поскольку номер модели задан как строка в изначальном списке, тут он тоже в кавычках.

Кроме метода writerow, поддерживается метод writerows. Ему можно передать любой итерируемый объект.

Например, предыдущий пример можно записать таким образом (файл csv\_writerows.py):

DictWriter

С помощью DictWriter можно записать словари в формат CSV.

В целом DictWriter работает так же, как writer, но так как словари не упорядочены, надо указывать явно в каком порядке будут идти столбцы в файле. Для этого используется параметр fieldnames (файл csv\_write\_dict.py):

Указание разделителя Иногда в качестве разделителя используются другие значения. В таком случае должна быть возможность подсказать модулю, какой именно разделитель использовать.

Например, если в файле используется разделитель; (файл sw\_data2.csv):

hostname; vendor; model; location

sw1; Cisco;3750; London

sw2; Cisco;3850; Liverpool

sw3; Cisco;3650; Liverpool

sw4; Cisco;3650; London

Достаточно просто указать, какой разделитель используется в reader (файл csv\_read\_delimiter.py)

**2.5.2 Добавление в программу функционала по работе с csv файлами**

Код: [См.Приложения2](#_Приложение2)

Файл csv:

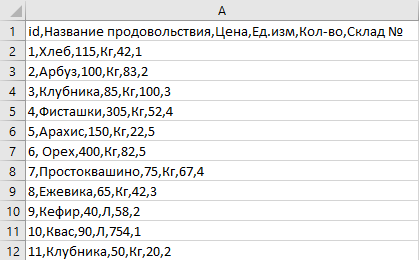


Рис №18. Файл сsv (Python)

# 3. Разработка мобильных приложений.

## **3.1 Разработка дизайна мобильного приложения в среде Figma**

**«Стартовая страница»:**

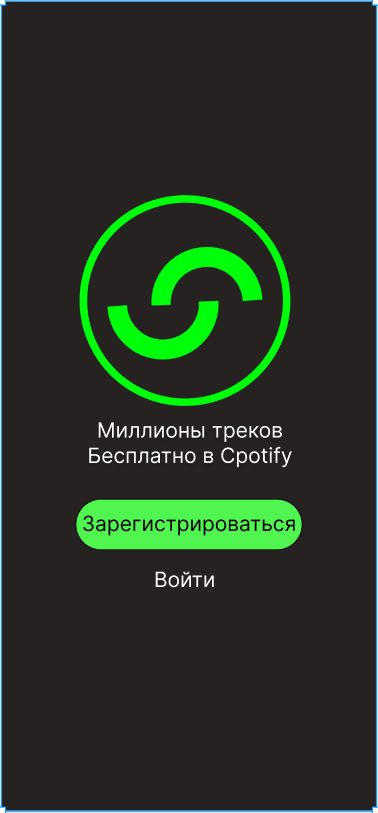
****

Рис №19. Стартовая страница

**Объекты «Стартовая страница»:**

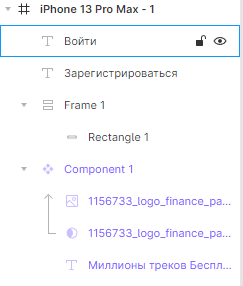
****

Рис №20. Объекты стартовой страницы

**«Страница авторизации»**

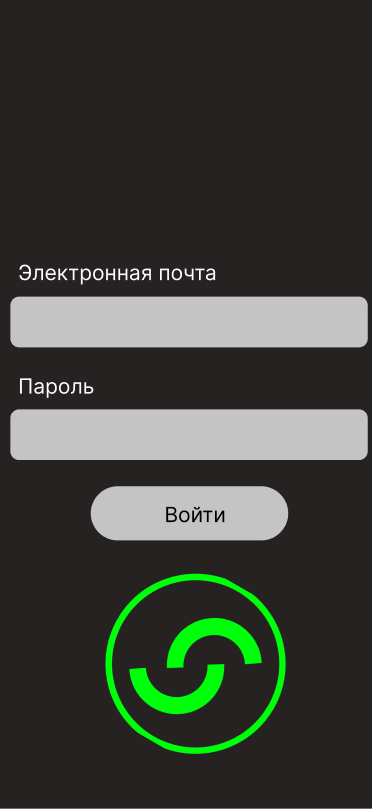
****

Рис №21. Страница авторизации

**Объекты «Страница авторизации»**

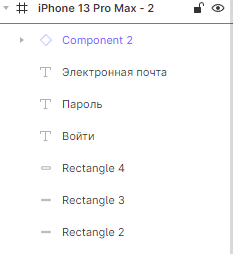
****

Рис №22. Объекты страницы авторизации

**«Главная страница»:**

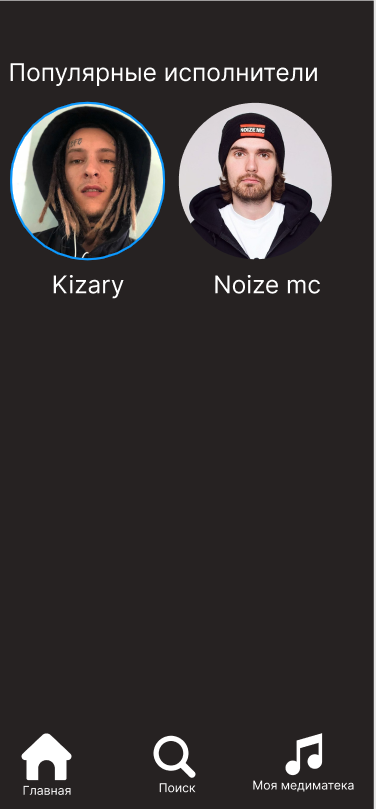
****

Рис №23. Главная страницы

**Объекты «Главная страницы»:**

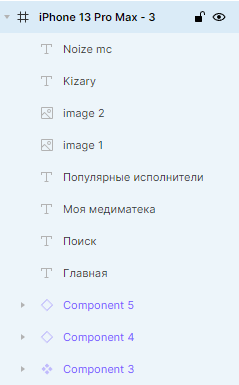
****

Рис №24. Объекты главной страницы

**«Страница исполнитель»**

****

Рис №25. Страница исполнителя

**Объекты «Страница исполнитель»**

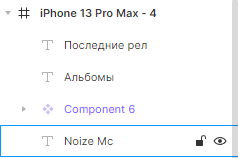
****

Рис №26. Объекты страницы исполнителя

## **3.2 Работа в среде Android**

**Xaml**

Стартовая страница:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#262222"  
 tools:context=".MainActivity" >  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="80dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logomain" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:width="280dp"  
 android:height="60dp"  
 android:background="@drawable/buttonshape"  
 android:text="Зарегистрироваться"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="100dp"  
 android:text="Войти"  
 android:textColor="#FFFFFF"  
 android:textSize="28dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button"  
 app:layout\_constraintVertical\_bias="0.336"  
 android:onClick="ClickToLogin"/>  
</androidx.constraintlayout.widget.ConstraintLayout>

Страница авторизации:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#262222"  
 tools:context=".authorize"  
 >  
  
 <TextView  
 android:id="@+id/LaberForEmail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="96dp"  
 android:text="@string/email"  
 android:textColor="#FFFFFF"  
 android:textSize="28sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.104"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/LabelForPassword"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/password"  
 android:textColor="#FFFFFF"  
 android:textSize="28sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.067"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txbEmail" />  
  
 <EditText  
 android:id="@+id/txbPassword"  
 android:layout\_width="380dp"  
 android:layout\_height="60dp"  
 android:layout\_marginTop="4dp"  
 android:height="80dp"  
 android:background="@drawable/inputshape"  
 android:ems="10"  
 android:inputType="textPassword"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.529"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/LabelForPassword"  
 tools:ignore="SpeakableTextPresentCheck"  
 android:importantForAutofill="no" />  
  
 <EditText  
 android:id="@+id/txbEmail"  
 android:layout\_width="380dp"  
 android:layout\_height="57dp"  
 android:height="80dp"  
 android:background="@drawable/inputshape"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.516"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/LaberForEmail"  
 tools:ignore="SpeakableTextPresentCheck"  
 android:importantForAutofill="no" />  
  
 <androidx.appcompat.widget.AppCompatButton  
 android:id="@+id/button4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:width="140dp"  
 android:background="@drawable/buttonshape"  
 android:backgroundTint="#C4C4C4"  
 android:text="@string/login"  
 android:textColor="#000000"  
 android:onClick="Authorize"  
  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txbPassword"  
 />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="32dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.496"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/button4"  
 app:srcCompat="@drawable/logologin"  
 android:contentDescription="TODO" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Главная страница:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".mainpage"  
 android:background="@color/back">  
  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="291dp"  
 android:layout\_height="46dp"  
  
 android:layout\_marginTop="16dp"  
 android:text="Главная страница"  
 android:textColor="@color/white"  
 android:textSize="30dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.133"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="361dp"  
 android:layout\_height="40dp"  
 android:layout\_marginTop="24dp"  
 android:text="Исполнители"  
 android:textColor="@color/white"  
 android:textSize="30sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView3" />  
  
 <ImageView  
 android:id="@+id/KizaryClick"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_weight="1"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/NoizeMcClick"  
 app:layout\_constraintHorizontal\_bias="0.421"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@drawable/kizary" />  
  
 <ImageView  
 android:id="@+id/NoizeMcClick"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="32dp"  
 android:onClick="noizemcClick"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView4"  
 app:layout\_constraintVertical\_bias="0.0"  
 app:srcCompat="@drawable/noizemc" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Страница исполнителя:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#262222"  
  
 tools:context=".IspKizary">  
  
 <TextView  
 android:id="@+id/textView17"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="264dp"  
 android:text="Грабли"  
 android:textColor="#FFFFFF"  
 android:textSize="22sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView19"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="244dp"  
 android:layout\_marginEnd="184dp"  
 android:text="NoizeMc"  
 android:textColor="#F1CFCF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/textView7"  
 app:layout\_constraintHorizontal\_bias="0.32"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5" />  
  
 <TextView  
 android:id="@+id/textView18"  
 android:layout\_width="81dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="280dp"  
 android:text="4:37"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.789"  
 app:layout\_constraintStart\_toEndOf="@+id/textView5"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView14"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="184dp"  
 android:text="Выдыхай"  
 android:textColor="#FFFFFF"  
 android:textSize="22sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView16"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="164dp"  
 android:layout\_marginEnd="184dp"  
 android:text="NoizeMc"  
 android:textColor="#F1CFCF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/textView7"  
 app:layout\_constraintHorizontal\_bias="0.32"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5" />  
  
 <TextView  
 android:id="@+id/textView15"  
 android:layout\_width="81dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="192dp"  
 android:text="3:08"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.789"  
 app:layout\_constraintStart\_toEndOf="@+id/textView5"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="104dp"  
 android:text="Чайлдфри"  
 android:textColor="#FFFFFF"  
 android:textSize="22sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView8"  
 android:layout\_width="81dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="112dp"  
 android:text="4:04"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.789"  
 app:layout\_constraintStart\_toEndOf="@+id/textView5"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView10"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="92dp"  
 android:layout\_marginEnd="184dp"  
 android:text="NoizeMc"  
 android:textColor="#F1CFCF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/textView7"  
 app:layout\_constraintHorizontal\_bias="0.32"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5" />  
  
 <ImageView  
 android:id="@+id/imageView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="BackClick"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.044"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.023"  
 app:srcCompat="@drawable/backbutton" />  
  
 <TextView  
 android:id="@+id/ispTxb"  
 android:layout\_width="181dp"  
 android:layout\_height="54dp"  
 android:text="NoizeMc"  
 android:textAlignment="center"  
 android:textColor="#FFFFFF"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.417"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.028" />  
  
 <TextView  
 android:id="@+id/ispTxb2"  
 android:layout\_width="181dp"  
 android:layout\_height="54dp"  
 android:text="Альбомы"  
 android:textAlignment="viewStart"  
 android:textColor="#FFFFFF"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.069"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.134" />  
  
 <TextView  
 android:id="@+id/ispTxb3"  
 android:layout\_width="311dp"  
 android:layout\_height="48dp"  
 android:text="Последние релизы"  
 android:textAlignment="viewStart"  
 android:textColor="#FFFFFF"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.03"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView5"  
 app:layout\_constraintVertical\_bias="0.022" />  
  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="24dp"  
 android:text="Вселенная бесконечна"  
 android:textColor="#FFFFFF"  
 android:textSize="22sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView7"  
 android:layout\_width="81dp"  
 android:layout\_height="31dp"  
 android:layout\_marginTop="32dp"  
 android:text="4:20"  
 android:textColor="#FFFFFF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.789"  
 app:layout\_constraintStart\_toEndOf="@+id/textView5"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb3" />  
  
 <TextView  
 android:id="@+id/textView6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="12dp"  
 android:layout\_marginEnd="184dp"  
 android:text="NoizeMc"  
 android:textColor="#F1CFCF"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/textView7"  
 app:layout\_constraintHorizontal\_bias="0.32"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textView5" />  
  
 <ImageView  
 android:id="@+id/imageView4"  
 android:layout\_width="245dp"  
 android:layout\_height="177dp"  
 android:layout\_marginTop="4dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.054"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb2"  
 app:srcCompat="@drawable/vihodvgorod" />  
  
 <ImageView  
 android:id="@+id/imageView5"  
 android:layout\_width="270dp"  
 android:layout\_height="184dp"  
 android:layout\_marginStart="68dp"  
 android:layout\_marginTop="4dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="@+id/imageView4"  
 app:layout\_constraintTop\_toBottomOf="@+id/ispTxb2"  
 app:srcCompat="@drawable/vo9djer" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Java**

**Стартовая страница:**

public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 public void ClickToLogin(View view) {  
 Intent intent = new Intent(this, authorize.class);  
 startActivity(intent);  
 }  
}

**Главная страница:**

public class mainpage extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_mainpage*);  
 }  
  
  
  
 public void noizemcClick(View view) {  
 Intent intent = new Intent(this, IspKizary.class);  
 startActivity(intent);  
 }  
}

**Страница авторизации:**

public class authorize extends AppCompatActivity {  
 private FirebaseAuth mDatabase;  
 private String User\_Key ="Users";  
 private EditText loginEd, passwordEd;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_authorize*);  
 init();  
  
 }  
 private void init(){  
 loginEd = findViewById(R.id.*txbEmail*);  
 passwordEd = findViewById(R.id.*txbPassword*);  
 mDatabase = FirebaseAuth.*getInstance*();  
   
 }  
 public void Authorize(View view) {  
 String email = loginEd.getText().toString();  
 String pass = passwordEd.getText().toString();  
 Resources resources = getResources();  
 mDatabase.signInWithEmailAndPassword(email,pass).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()){  
 Intent i = new Intent( authorize.this, mainpage.class);  
 startActivity(i);  
 finish();  
 }else {  
 loginEd.setTextColor(resources.getColor(R.color.*green*));  
 }  
 }  
 });  
  
  
 }

## **3.3 Структуру файла AndroidManifest.xml на основе файла вашего проекта.**

AndroidManifest.xml — это основной файл манифеста конфигурации всего приложения, который содержит такую информацию, как имя пакета, номер версии, компоненты и разрешения. Файл манифеста используется для записи информации о конфигурации, связанной с приложением.

Файл манифеста может содержать только один узел приложения, Приложение может установить значок (android: icon), заголовок (android: label), стиль темы (android: theme).

Тег приложения может содержать множество компонентов, таких как четыре основных компонента: android, activity, service, contentprovider и broadcastreceiver, чтобы указать компоненты приложения.

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.kd.crapps">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Crapps">  
 <activity  
 android:name=".MainActivity3"  
 android:exported="false" />  
 <activity  
 android:name=".MainActivity2"  
 android:exported="false" />  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>

## **3.4 Структуру и назначение файла build.gradle**

Gradle — система автоматической сборки, построенная на принципах Apache Ant и Apache Maven. В Eclipse использовалась система Ant, но большинство разработчиков даже не замечало её работы. В основном возможности системы использовались в конторах для автоматизации различных задач. В Android Studio такой номер не пройдёт. Gradle сопровождает вас во время разработки постоянно. Поначалу, если вы перешли с Eclipse, Gradle сильно раздражает своими действиями. Но позже вы оцените её удобство и может даже полюбите её.

Также есть файл build.gradle, который относится к проекту. Но с ним работают реже. Так находятся настройки для репозиториев и самого Gradle.

// Top-level build file where you can add configuration options common to all sub-projects/modules.  
buildscript {  
 repositories {  
 google()  
 mavenCentral()  
 }  
 dependencies {  
classpath "com.android.tools.build:gradle:7.0.4"  
  
 // NOTE: Do not place your application dependencies here; they belong  
 // in the individual module build.gradle files  
 **}  
}**task clean(type: Delete) **{** delete rootProject.buildDir  
**}**

plugins **{** id 'com.android.application'  
**}**android **{** compileSdk 32  
  
 defaultConfig **{** applicationId "com.kd.crapps"  
 minSdk 28  
 targetSdk 32  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}** buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}  
}**dependencies **{** implementation 'androidx.appcompat:appcompat:1.4.1'  
 implementation 'com.google.android.material:material:1.5.0'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.3'  
 testImplementation 'junit:junit:4.+'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
**}**

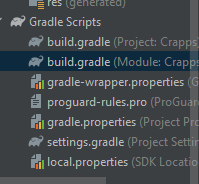


Рис №27. Gradle

## **3.5 Установка и запуск приложения**

**3.5.1 Процесс создания эмулятора устройства и запуска проекта на нём.**

AVD расши ровывается как Android Virtual Device, а именно виртуальное Андроид устройство. На вашем компьютере вы открываете заранее выбранную и сконфигурированную систему. Это удобно даже когда у вас есть реальный телефон. В эмуляторе можно изменить версию ОС, размер экрана и другие параметры. Это позволяет протестировать то, как работает ваше приложение на разных версиях Android.

В Android Studio всеми виртуальными устройствами управляет AVD manager. Его можно открыть через меню Tools в Android studio**.**

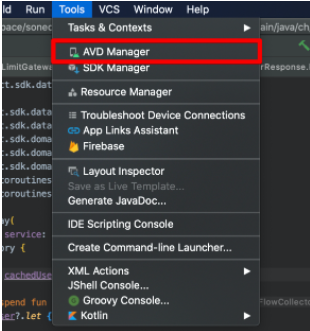
****

Рис №27. Tools

**Создание нового Андроид эмулятора:**

В AVD Manager нужно нажать на кнопку создания нового виртуального устройства и выбрать предустановленный шаблон или же создать свой новый. Давайте выберем Pixel XL.

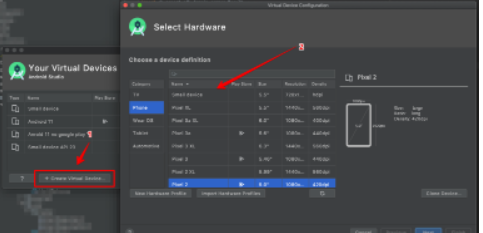
****

Рис №28. AVD Manager

Следующим пунктом необходимо выбрать образ операционной системы. Это важный шаг, т.к. именно от него будет зависеть как версия Android будет установлена на новом виртуальном устройстве.

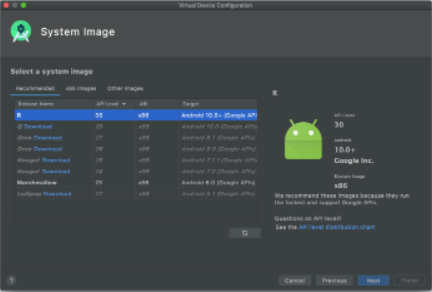
****

Рис №29. AVD Manager

**Управление настройками и запуск эмулятора:**

Когда все будет готово останется лишь запустить созданный эмулятор Android устройства. Делается это в уже знакомом AVD Manager. Также в нем вы можете найти возможные настройки и пункты управления девайсом.

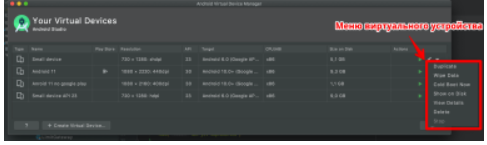
****

Рис №30. AVD Manager

**3.5.2 Процесс установки проекта на реальное устройство.**

Если у вас есть реальное Android устройство, вот как вы можете установить и запустить приложение:

Подключите ваш смартфон к компьютеру с помощью кабеля USB. Если вы разрабатываете на Windows, вам может потребоваться установить соответствующий драйвер USB для вашего устройства. Включите Отладка через USB на вашем устройстве.

На большинстве устройств под управлением Android 3.2 или старше, вы можете найти опцию вНастройки> Приложения> Разработка.

На Android 4.0 и новее, это находится в Настройки> Для разработчиков.

Примечание: на Android 4.2 и новее, для разработчиков по умолчанию скрыт. Чтобы сделать его доступным, перейдите к Настройки 🡪 О телефоне и нажмите Номер сборки семь раз. Вернитесь к предыдущему экрану, чтобы найти для разработчиков.

## **3.6 Классы, используемые в проекте при разработке.**

Для взаимодействия между различными объектами activity ключевым классом является android.content.Intent. С помощью данного класса был реализован переход между экранами приложения. Так же по средствам данного класса можно передать значения между activity.

С помощью TextView можно выводить текст.

Класс View - представляет базовый стандартный блок для компонентов пользовательского интерфейса.

Класс Bundle - сопоставление строковых ключей с различными значениями.

## **3.7 Виды ошибок и предупреждений при разработке проекта.**

Слишком контрастный цвет у текста:

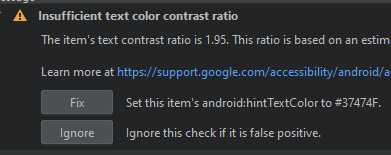
****

Рис №31. Ошибка№1

Ошибка с кнопкой:

****

Рис №32. Ошибка№2

# 4. Тестирование программных модулей

## **4.1. План тестирования**

* Тестирование авторизации, и правильности переходов на страницу после авторизации (в зависимости от роли пользователя).
* Тестирование функционала «Тестирования» получение результатов тестирования
* Изменение личных данных самим пользователем на странице «Профиль»
* Авторизация с неверными данными.

## **4.2. Сценарии тестирования с результатами**

## **Test case #1**

|  |  |
| --- | --- |
| **Тестовый пример #** | 1 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Авторизация обычного пользователя |
| **Краткое изложение теста** | После ввода данных пользователь должен оказаться на странице пользовательского интерфейса |
| **Этапы теста** | 1. Запуск приложения 2. Ввод данных в textbox-ы(Пароль,Логин) 3. Потверждение данных |
| **Тестовые данные** | Login – TestLog, Password-TestPas |
| **Ожидаемый результат** | Пользователь успешно оказался на странице пользовательского интерфейса |
| **Фактический результат** | Пользователь успешно оказался на странице пользовательского интерфейса |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | Тест пройден успешно |

## **Test case #2**

|  |  |
| --- | --- |
| **Тестовый пример #** | 2 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Авторизация пользователя с ролью администратора |
| **Краткое изложение теста** | После ввода данных пользователь должен оказаться на странице админского интерфейса. |
| **Этапы теста** | 1. Запуск приложения 2. Ввод данных для авторизации(Аккаунт админа) 3. Подтверждение данных |
| **Тестовые данные** | Login – LogAdmin, Password - Testgg |
| **Ожидаемый результат** | Пользователь успешно оказался на странице админского интерфейса |
| **Фактический результат** | Пользователь успешно оказался на странице админского интерфейса |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | Тест пройден успешно |

## **Test case #3:**

|  |  |
| --- | --- |
| Тестовый пример **#** | 3 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Получение результатов после прохождения тестирования |
| **Краткое изложение теста** | После прохождения тестирования пользователю должна быть предоставлена информация о результатах прохождения(процент правильных ответов) |
| **Этапы теста** | 1. Запуск приложения 2. Авторизация 3. Выбор категории 4. Переход на определенный тест 5. Прохождение теста 6. Получение результата(в messagebox) |
| **Тестовые данные** | Login –TestLog, Password - TestPas |
| **Ожидаемый результат** | После прохождения теста, высвечивается всплывающее окно с результатом тестирования. |
| **Фактический результат** | После прохождения теста, высвечивается всплывающее окно с результатом тестирования. |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | Тест пройден успешно. |

## **Test case #4:**

|  |  |
| --- | --- |
| Тестовый пример **#** | 4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Личное изменение данных пользователем |
| **Краткое изложение теста** | На странице «Профиль» пользователь используя интерфейс может изменить личные данные введенные при регистрации. |
| **Этапы теста** | 1. Запуск приложения 2. Авторизация 3. Переход на страницу профиля 4. Клик на кнопку «Разрешить редактирование» 5. Изменить данные Email и Name 6. Нажать «сохранить» |
| **Тестовые данные** | Login –TestLog, Password – TestPas, новый email – [Shahanov@mail.ru](mailto:Shahanov@mail.ru), Name- Никита |
| **Ожидаемый результат** | Пользователь успешно изменил свои данные и высветился всплывающее окно с уведомление о изменение данных |
| **Фактический результат** | Пользователь успешно изменил свои данные и высветился всплывающее окно с уведомление о изменение данных |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | Тест пройден успешно |

## **Test case #5:**

|  |  |
| --- | --- |
| **Test Case #** | 5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Ввод неверных данных при авторизациия |
| **Краткое изложение теста** | После ввода неверных данных при авторизации пользователю должен высветится попап, сообщающий о вводе неверных данных |
| **Этапы теста** | 1. Запуск приложения 2. На окне авторизации ввести неверные данные 3. Подтвердить |
| **Тестовые данные** | Login – 12341 Password -153 |
| **Ожидаемый результат** | Пользователь не смог авторизоваться, пользователя высветился всплывающее окно, сообщающий о вводе неверных данных |
| **Фактический результат** | Пользователь не смог авторизоваться, пользователя высветился всплывающее окно, сообщающий о вводе неверных данных |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | - |
| **Примечания/комментарии** | Тест пройден успешно. |

## **4.3. Предложения по улучшению функциональности**

1. Добавление функционала получения наград за завершения категорий на 100%(Достижения)

2. возможность отправки репортов/рекомендаций/комментариев пользователем

# 5. Разработка технической документации

## **5.1. Техническое задание**

# Введение

ООО “Заявки” функционирует с 2003 года в сегменте сети, подключая людей к интернету.

# Назначение разработки

Автоматизированная информационная система “Заявки” предназначена для сбора и обработки информации о поступивших заявках на подключение интернета.

# 4 Требования к программе или программному изделию

## **4.1 Требования к функциональным характеристикам**

## **4.1.1. Требования к составу выполняемых функций:**

* Добавление/просмотр/удаление заявок
* Системы авторизации пользователя
* Экспорт данных в excel
* Добавление/просмотр/удаление городов
* Регистрация пользователя
* Добавление исполнителей/просмотр/удаление исполнителей
  + 1. **Требования к организации входных данных:**

Входные данные программы организованы в виде ввода в текстовые поля или выбором: Email, ФИО, Название города, услуга, исполнитель, описание, дата, улица, номер телефона, «ВК».

**Требования к организации выходных данных:**

Выходные данные организованы в виде экспорта отчета в виде Excel документа.

## **4.2 Требования к надежности.**

Программа должна быть в достаточной степени надёжна от сбоев. На крайний случай предусмотрено сохранение или восстановление данных в случае завершения работы, в программе предусмотрена парольная защита.

## **4.3 Условия эксплуатации**

Программа не требует специального обслуживания. Для ознакомления с полным функционалом пользователь должен прочесть Руководство пользователя. Для работы с программой требуются хотя бы малейшие навыки работы с 1С и похожими продуктами по автоматизации склада.

**Климатические условия эксплуатации:**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

## **4.4 Требования к составу и параметрам технических средств**

Процессор - Intel Celeron J4025

Процессор, частота - 2 ГГц (2.9 ГГц, в режиме Turbo)

Оперативная память - 4 ГБ, DDR4, SO-DIMM, 2133 МГц

Графика - Intel UHD Graphics 600

Минимальный объём свободной памяти - 500 МБ

## **4.5 Требования к информационной и программной совместимости**

Для корректной работы программы необходимо: ОС Windows 7/8/8.1/10, Microsoft Excel 2010 (и выше).

## **4.6 Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса. Программа должна обеспечивать удобный и быстрый экспорт данных в виде Excel-файла.

# Требования к программной документации

В ходе разработки программы должны быть подготовлены следующие программные документы: текст программы, описание программы, программа и методика испытаний, руководство пользователя, руководство программиста, технико-экономическое обоснование.

## **5.3. Руководство пользователя**

**1.Назначение программы:**

Автоматизированная информационная система “Заявки” предназначена для сбора и обработки информации о поступивших заявках на подключение интернета.

**2.Условия выполнения программы:**

Процессор - Intel Celeron J4025

Процессор, частота - 2 ГГц (2.9 ГГц, в режиме Turbo)

Оперативная память - 4 ГБ, DDR4, SO-DIMM, 2133 МГц

Графика - Intel UHD Graphics 600

Минимальный объём свободной памяти - 500 МБ

Требования к программным средствам:

## Для корректной работы программы необходимо: ОС Windows 7/8/8.1/10, Microsoft Excel 2010 (и выше).

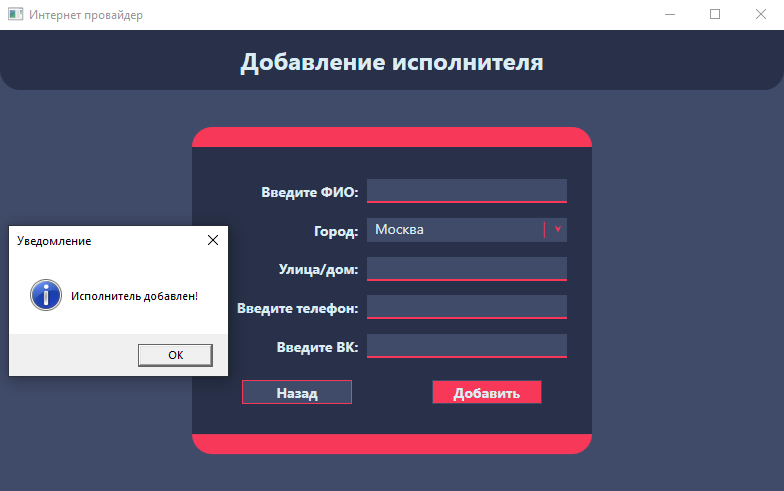
**3.Выполнение программы:**

При открытии программы (исполняемый «.exe» файл) появится окно авторизации, где необходимо ввести пароль. Если введённый пароль не правильный – пользователя уведомит об этом диалоговое окно с ошибкой.

Если пароль введен верно – откроется окно программы(Главная страница). В ней возможно добавить заявки и просмотреть их.

**Добавление исполнителя:**

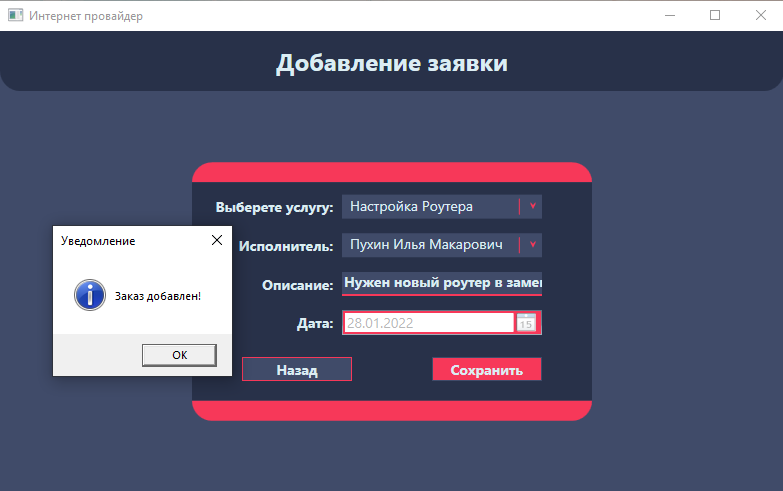
Для добавления исполнителя необходимо перейти в соответствующие окно. В окне необходимо ввести следующие данные: «Фио», «Город», «Улица/дом», «телефон», «ВК» и нажать на кнопку «Добавить»;



**Добавление заявки:**

Для добавление заявки необходимо перейти в соответствующие окно.

В окне необходимо ввести следующие данные: Выбрать «Услугу», «Исполнителя», ввести «Описание», «Дата».



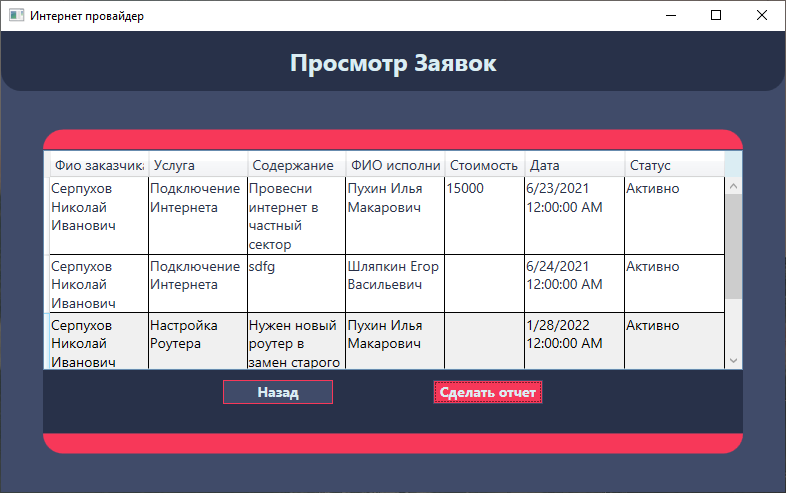
**Добавление/удаление города**

Для добавления города необходимо перейти в соответствующие окно. В окне необходимо ввести «Название города» и нажать добавить город. Если же нужно наоборот удалить, нажмите на кнопку «Удалить город», введенный город удалиться.



**Экспорт отчетов в Excel:**

Кнопка “Сделать отчет” располагается в окне просмотра заявок в нижней панели инструментов. После нажатия программа экспортирует данные в excel.



**4.Сообщение оператору:**

Попытка входа с в неверными введенными данными.

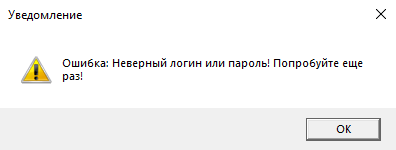


Рис №1 Неверные данные

При добавление заказа:

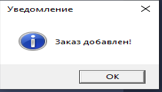


Рис №2 Добавление

## **5.4. Руководство программисту**

1. **Назначение и условия применения программы**

Автоматизированная информационная система “Заявки” предназначена для сбора и обработки информации о поступивших заявках на подключение интернета.

**Характеристика программы**

Программа должна обеспечивать следующие функции:

* Добавление/просмотр/удаление заявок
* Системы авторизации пользователя
* Экспорт данных в excel
* Добавление/просмотр/удаление городов
* Регистрация пользователя
* Добавление исполнителей/просмотр/удаление исполнителей

1. **Обращение к программе**

С# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270

Xaml код окна добавления города:

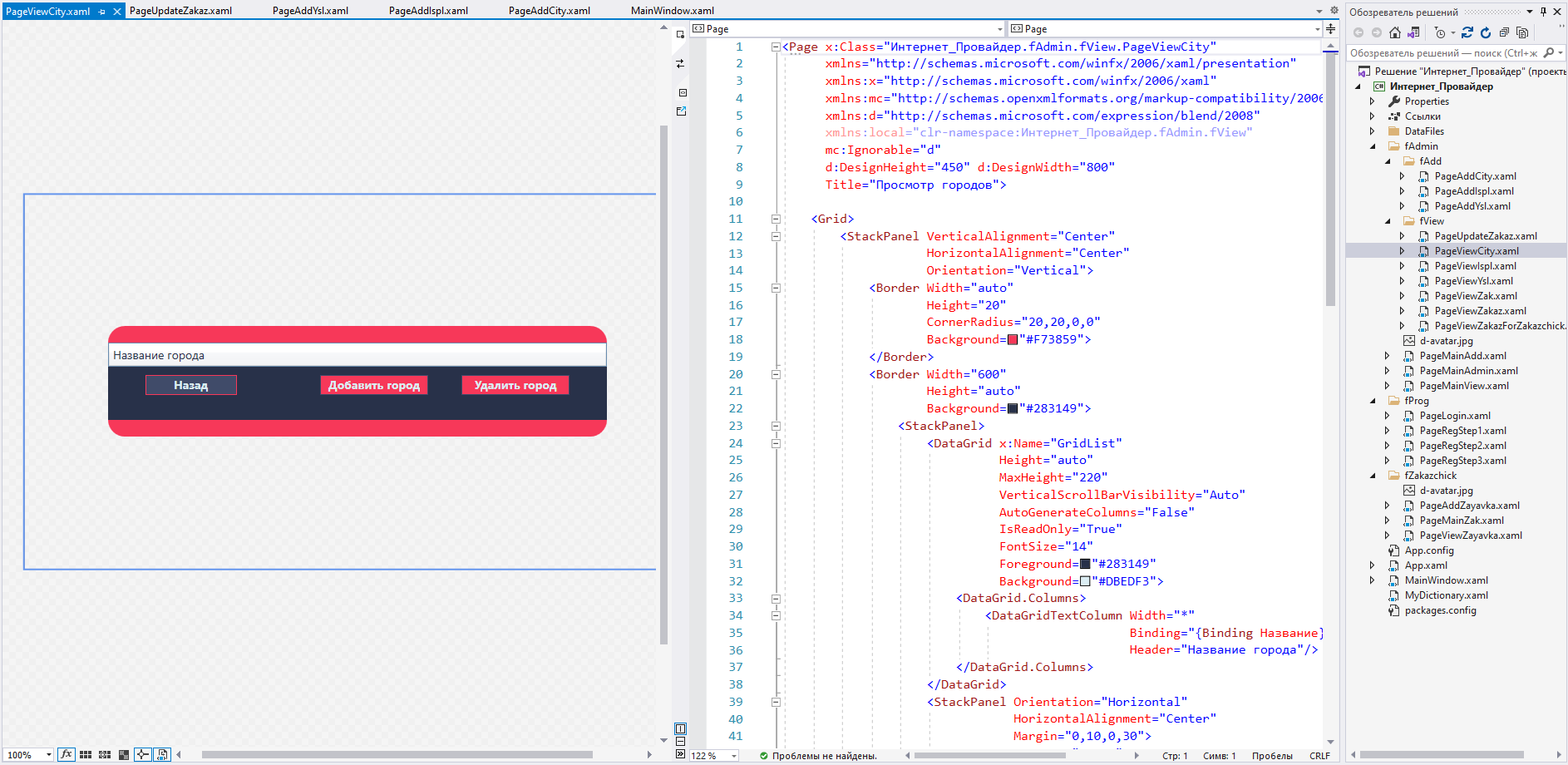


Рис №1 «окно добавления города»

Xaml код окна добавления исполнителя

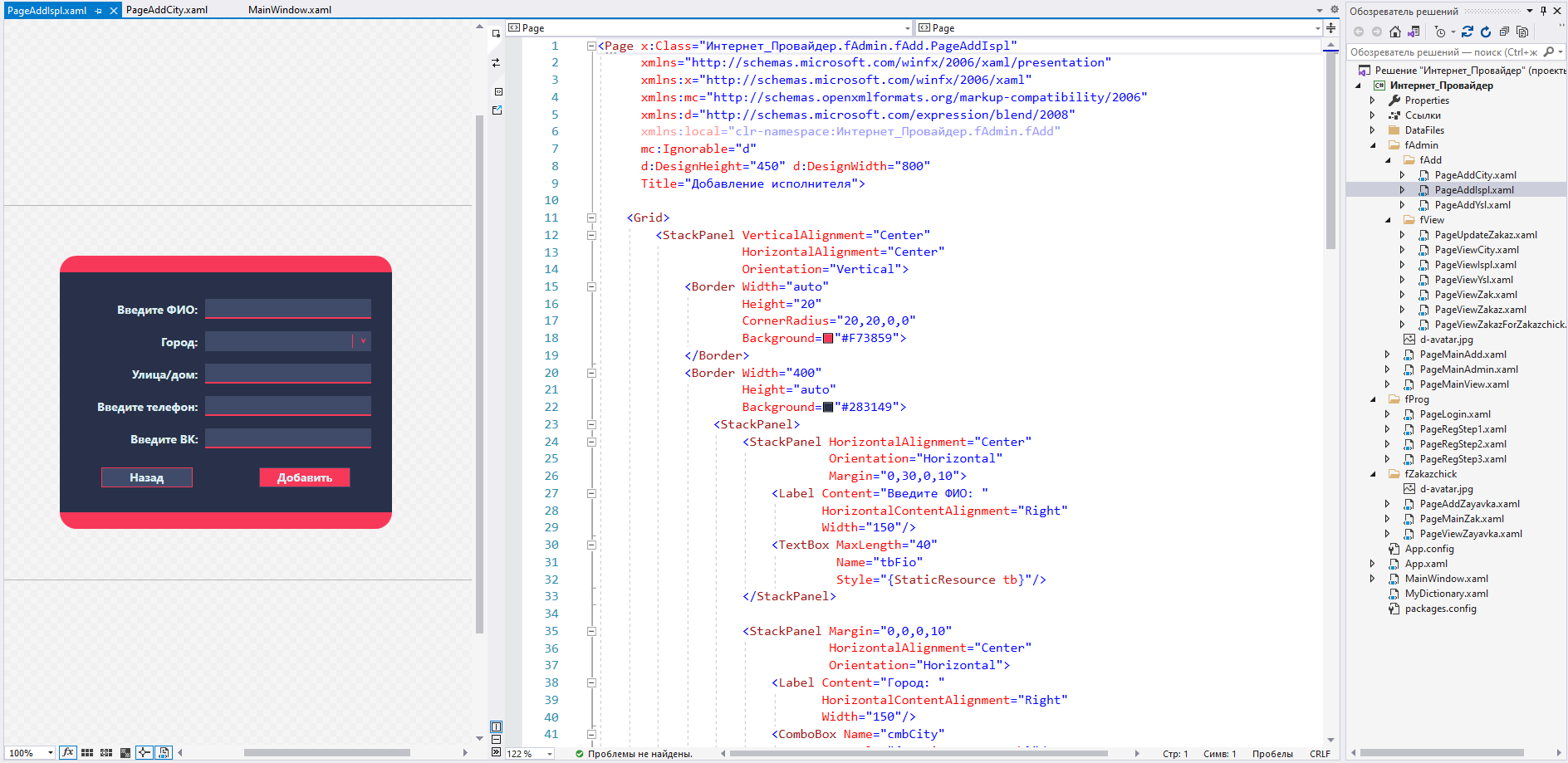


Рис №2 «окно добавления исполниеля»

# Входные и выходные данные

Входные данные:

|  |  |
| --- | --- |
| **Поля** | **Тип данных** |
| Пароль | Текстовой |
| Фио | Текстовой |
| Логин | Текстовой |
| Город | Текстовой |
| Номер телефона | Цифровой |
| Вк | Текстовой |
| Email | Текстовой |
| Услуга | Текстовой |
| Исполнитель | Текстовой |
| Дата | Дата |
| Улица | Текстовой |

**Выходная информация:**

Выходной информацией является экспорт «заявок» виде нового Excel-файла.

# Сообщения

Попытка входа с в неверными введенными данными.

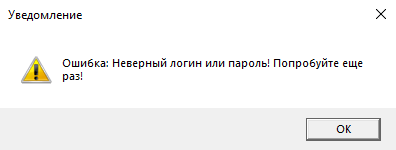


Рис №1 Неверные данные

При добавление заказа:

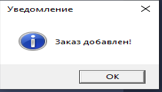


Рис №2 Добавление

# Заключение

Учебную практику по ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» проходил в учебном учреждении Ликино-Дулевский политехнический колледж – филиал ГГТУ.

Во время учебной практики выполнялись следующие виды работ:

1. Анализ и проектирование программных решений.
2. Разработка программных модулей для информационной системы.
3. Проведение тестирования, отладки и оптимизации программных модулей.
4. Разработки мобильных приложений.

В соответствии с постановкой задачи был разработан программный модуль со следующей структурой данных:

* **Сategories:**

Cathegoryid;

СategoryName

TemaId

ImageUrl

* **ItemLearnCategory**

ImageUrlItem

NameItemLearn

Url\_toLearningSite

Id\_LearnCategory

Id\_ItemLearn

* **LearningCategory**

ImageLearnCat

NameLearnCategory

TemaId

IdLearnCat

* **Questions**

TextQuest

Id\_quest

Id\_test

Quest\_Answer1

Quest\_answer2

Quest\_answer3

QuestAnaswer4

Quest\_rightanswer

* **Results**

CategoryId

MedalImage

NameTestDone

ScorePercennt

TestID

User\_Login

* **Role**

Id\_role

NameRole

* **Tema**

NameTema

TemaId

* **Tests**

CategoryId

Description

Name

RightAnswerId

TestId

* **Users**

BirthDay

Email

Login

Name

Phone

Password

RoleId

SurName

Хранение данных должно быть организованно в облачной базе данных firebase. Для разработки программного модуля использовалась среда программирования СИ#.

В техническом задании отражены требования к составу технических средств, информационно-программной совместимости и требования, к надёжности, которые должны быть соблюдены во время разработки, внедрении и эксплуатации приложения.

В соответствии с постановкой задачи разработан алгоритм работы программного модуля.

Произведена отладка, тестирование и оптимизация программного модуля с использованием встроенных средств отладки приложений в среде программирования.

На основании созданного программного модуля и в соответствии с ГОСТ разработана техническая документация.

# Список литературы

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студентов СПО /Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 384с.
2. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студентов СПО / Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 285 с.
3. Черпаков, И. В.Основы программирования: учебник и практикум для среднего профессионального образования /И. В. Черпаков. — Москва : Издательство Юрайт, 2019. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст: электронный //ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/436557>
4. Хахаев, И.А. Практикум по алгоритмизации и программированию на Python: курс /И.А. Хахаев. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 179 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429256>
5. Царёв, Р.Ю. Алгоритмы и структуры данных (CDIO): учебник /Р.Ю. Царёв, А.В. Прокопенко; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: СФУ, 2016. - 204 с. : ил. - Библиогр. в кн. - ISBN 978-5-7638-3388-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=497016>
6. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 2-е изд. — Москва: Издательство Юрайт, 2020. — 161 с.
7. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с.
8. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2020. — 175 с.
9. C# 7.0. Справочник. Полное описание языка.: Пер. с англ. – СпБ.: ООО “Альфакнига”, 2018. – 1024 с. : ил. – Парал. тит. англ.
10. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017. – 848 с.: ил. – (Серия «Библиотека программиста»).
11. Программирование на C# для начинающих. Основные сведения / Алексей Васильев. – Москва: Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
12. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2018 – 1328 с. : ил. – Парал. тит. англ.
13. Оппель, Эндрю Дж. SQL. Полное руководство / Оппель Эндрю Дж.. - М.: Диалектика / Вильямс, 2016. - 902 c.
14. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 161 с. — (Профессиональное образование). — ISBN 978-5-534-11961-9.
15. Гниденко, И. Г.  Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9.
16. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2020. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0

**Интернет-ресурсы**

ЭБС «Университетская библиотека online» - <http://biblioclub.ru/>

ЭБС «Юрайт» - <http://www.urait.ru/>

# Приложения

## **Приложение1**

menu = {  
 '1': 'Добавить запись',  
 '2': 'Удалить запись',  
 '3': 'Вывести все записи о товаре на указанном складе: ',  
 '4': 'Вывести всю информацию о товаре с ценой меньше указанной: ',  
 '5': 'Вывести всю информацию о товаре с указанной ед. изм.: ',  
 '6': 'Вывести среднюю цену товара по складам в формате: «название склада: сред. Цена»: ',  
 '7': 'Вывести всю информацию об указанном товаре: ',  
 '8': 'Вывести все данные записи и сумму по каждой записи. Сумма = Цена\*кол-во: ',  
 '9': 'Ввывести все данные',  
 '0': ' <-Меню',  
 'exit': 'Выйти'  
}  
  
staff = [{'id': 1, 'Название продовольствия': 'Хлеб','Цена': 115, 'Ед.изм': 'Кг', 'Кол-во': 42, 'Склад №': 1},  
{'id': 2, 'Название продовольствия': 'Арбуз', 'Цена': 100, 'Ед.изм': 'Кг', 'Кол-во': 83, 'Склад №': 2},  
{'id': 3, 'Название продовольствия': 'Клубника', 'Цена': 85, 'Ед.изм': 'Кг', 'Кол-во': 100, 'Склад №': 3},  
{'id': 4, 'Название продовольствия': 'Фисташки', 'Цена': 305, 'Ед.изм': 'Кг', 'Кол-во': 52, 'Склад №': 4},  
{'id': 5, 'Название продовольствия': 'Арахис', 'Цена': 150, 'Ед.изм': 'Кг', 'Кол-во': 22, 'Склад №': 5},  
{'id': 6, 'Название продовольствия': 'Орех', 'Цена': 400, 'Ед.изм': 'Кг', 'Кол-во': 82, 'Склад №': 5},  
{'id': 7, 'Название продовольствия': 'Простоквашино', 'Цена': 75, 'Ед.изм': 'Кг', 'Кол-во': 67, 'Склад №': 4},  
{'id': 8, 'Название продовольствия': 'Ежевика', 'Цена': 65, 'Ед.изм': 'Кг', 'Кол-во': 42, 'Склад №': 3},  
{'id': 9, 'Название продовольствия': 'Кефир', 'Цена': 40, 'Ед.изм': 'Л', 'Кол-во': 58, 'Склад №': 2},  
{'id': 10,'Название продовольствия': 'Квас', 'Цена': 90, 'Ед.изм': 'Л', 'Кол-во': 754, 'Склад №': 1}  
 ]  
  
for key, val in menu.items():  
 print(key, '-', val)  
  
while True:  
 action = input('>\_ ')  
  
 if action == '1':  
 NazProd = input('Название продовольствия:')  
 price = int(input('Цена:'))  
 Edizmer = input('Ед.изм:')  
 kolvo = int(input('Кол-во:'))  
 sklad = input('Cклад №:')  
 lid = staff[-1]['id']  
 staff.append({'id': lid + 1, 'Название продовольствия': NazProd, 'Цена': price, 'Ед.изм': Edizmer, 'Кол-во': kolvo, 'Cклад №': sklad})  
 print('Данные добавлены!')  
  
 elif action == '2':  
 key\_del = input('Код продукции: ')  
 fl = False  
 for i in range(len(staff)):  
 if staff[i]['id'] == key\_del:  
 del staff[i]  
 print('Данные удалены!')  
 fl = True  
 break  
 if fl == False:  
 print('Такого кода нет в списке!')  
  
 elif action == '3':  
 sklad = int(input('Введите номер склада:'))  
 fl = False  
 for i in range(len(staff)):  
 if sklad == staff[i]['Склад №']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Склад №", sklad, "не найден!")  
  
 elif action == '4':  
 price = int(input('Введите цену: '))  
 fl = False  
 for i in range(len(staff)):  
 if price > staff[i]['Цена']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Ниже такой цены ничего нету")  
  
 elif action == '5':  
 Edizmer = input('Введите ед.измерения: ')  
 fl = False  
 for i in range(len(staff)):  
 if Edizmer == staff[i]['Ед.изм']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Такого единицы", Edizmer, "Нету")  
  
 elif action == '6':  
  
 src\_sklad5 = 0  
 src\_sklad4 = 0  
 src\_sklad3 = 0  
 src\_sklad2 = 0  
 src\_sklad1 = 0  
  
 delitel1 = 0  
 delitel2 = 0  
 delitel3 = 0  
 delitel4 = 0  
 delitel5 = 0  
  
 for i in range(len(staff)):  
  
 if 4 == staff[i]['Склад №']:  
 src\_sklad4 += staff[i]['Цена']  
 delitel4 += 1  
  
 elif 3 == staff[i]['Склад №']:  
 src\_sklad3 += staff[i]['Цена']  
 delitel3 += 1  
  
 elif 2 == staff[i]['Склад №']:  
 src\_sklad2 += staff[i]['Цена']  
 delitel2 += 1  
  
 elif 1 == staff[i]['Склад №']:  
 src\_sklad1 += staff[i]['Цена']  
 delitel1 += 1  
  
 elif 5 == staff[i]['Склад №']:  
 src\_sklad5 += staff[i]['Цена']  
 delitel5 += 1  
 src\_sklad5 /= delitel5  
 src\_sklad4 /= delitel4  
 src\_sklad3 /= delitel3  
 src\_sklad2 /= delitel2  
 src\_sklad1 /= delitel1  
 print("{}\t{}\t".format(1, round(src\_sklad1,3)))  
 print("{}\t{}\t".format(2, round(src\_sklad2,3)))  
 print("{}\t{}\t".format(3, round(src\_sklad3,3)))  
 print("{}\t{}\t".format(4, round(src\_sklad4,3)))  
 print("{}\t{}\t".format(5, round(src\_sklad5,3)))  
  
 elif action == '7':  
 NazProd = input('Введите продукцию:')  
 fl = False  
 for i in range(len(staff)):  
 if NazProd == staff[i]['Название продовольствия']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Продукция", NazProd, "не найден!")  
  
 elif action == '8':  
 sum = 0  
 for i in range(len(staff)):  
 sum = staff[i]['Цена'] \* staff[i]['Кол-во']  
 print("{}\t{}\t{}\t{}\t{}\tСумма: {}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во'], sum))  
  
 elif action == '9':  
 for i in range(len(staff)):  
 print("{}\tНазвание продовольствия: {:13}\tЦена: {:4}\tЕд.изм: {:3}\tКол-во: {:4}\tСклад №{:3}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во'],staff[i]['Склад №']))  
  
 elif action == '0':  
 for key, val in menu.items():  
 print(key, '-', val)  
 elif action == 'exit':  
 break

## **Приложение2**

import csv  
from builtins import print  
  
FILENAME = "users2.csv"  
  
menu = {  
 '1': 'Добавить запись',  
 '2': 'Удалить запись',  
 '3': 'Вывести все записи о товаре на указанном складе: ',  
 '4': 'Вывести всю информацию о товаре с ценой меньше указанной: ',  
 '5': 'Вывести всю информацию о товаре с указанной ед. изм.: ',  
 '6': 'Вывести среднюю цену товара по складам в формате: «название склада: сред. Цена»: ',  
 '7': 'Вывести всю информацию об указанном товаре: ',  
 '8': 'Вывести все данные записи и сумму по каждой записи. Сумма = Цена\*кол-во: ',  
 '9': 'Ввывести все данные',  
 '10':'Открыть файл',  
 '11':'Сохранить в файл',  
 '0': ' <-Меню',  
 'exit': 'Выйти'  
}  
  
staff = []  
  
for key, val in menu.items():  
 print(key, '-', val)  
  
while True:  
 action = input('>\_ ')  
  
 if action == '1':  
 lid = int(input('Введите код: '))  
 NazProd = input('Название продовольствия:')  
 price = int(input('Цена:'))  
 Edizmer = input('Ед.изм:')  
 kolvo = int(input('Кол-во:'))  
 sklad = input('Cклад №:')  
 staff.append({'id': lid, 'Название продовольствия': NazProd, 'Цена': price, 'Ед.изм': Edizmer, 'Кол-во': kolvo, 'Склад №': sklad})  
 print('Данные добавлены!')  
  
 elif action == '2':  
 key\_del = input('Код продукции: ')  
 fl = False  
 for i in range(len(staff)):  
 if staff[i]['id'] == key\_del:  
 del staff[i]  
 print('Данные удалены!')  
 fl = True  
 break  
 if fl == False:  
 print('Такого кода нет в списке!')  
  
 elif action == '3':  
 sklad = input('Введите номер склада:')  
 fl = False  
 for i in range(len(staff)):  
 if sklad == staff[i]['Склад №']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Склад №", sklad, "не найден!")  
  
 elif action == '4':  
 price = int(input('Введите цену: '))  
 fl = False  
 for i in range(len(staff)):  
 if price > int(staff[i]['Цена']):  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Ниже такой цены ничего нету")  
  
 elif action == '5':  
 Edizmer = input('Введите ед.измерения: ')  
 fl = False  
 for i in range(len(staff)):  
 if Edizmer == staff[i]['Ед.изм']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Такого единицы", Edizmer, "Нету")  
  
 elif action == '6':  
  
 src\_sklad5 = 0  
 src\_sklad4 = 0  
 src\_sklad3 = 0  
 src\_sklad2 = 0  
 src\_sklad1 = 0  
  
 delitel1 = 0  
 delitel2 = 0  
 delitel3 = 0  
 delitel4 = 0  
 delitel5 = 0  
  
 for i in range(len(staff)):  
  
 if 4 == int(staff[i]['Склад №']):  
 src\_sklad4 += int(staff[i]['Цена'])  
 delitel4 += 1  
  
 elif 3 == int(staff[i]['Склад №']):  
 src\_sklad3 += int(staff[i]['Цена'])  
 delitel3 += 1  
  
 elif 2 == int(staff[i]['Склад №']):  
 src\_sklad2 += int(staff[i]['Цена'])  
 delitel2 += 1  
  
 elif 1 == int(staff[i]['Склад №']):  
 src\_sklad1 += int(staff[i]['Цена'])  
 delitel1 += 1  
  
 elif 5 == int(staff[i]['Склад №']):  
 src\_sklad5 += int(staff[i]['Цена'])  
 delitel5 += 1  
  
 src\_sklad5 /= delitel5  
 src\_sklad4 /= delitel4  
 src\_sklad3 /= delitel3  
 src\_sklad2 /= delitel2  
 src\_sklad1 /= delitel1  
 print("{}\t{}\t".format(1, round(src\_sklad1,3)))  
 print("{}\t{}\t".format(2, round(src\_sklad2,3)))  
 print("{}\t{}\t".format(3, round(src\_sklad3,3)))  
 print("{}\t{}\t".format(4, round(src\_sklad4,3)))  
 print("{}\t{}\t".format(5, round(src\_sklad5,3)))  
  
  
 elif action == '7':  
 NazProd = input('Введите продукцию:')  
 fl = False  
 for i in range(len(staff)):  
 if NazProd == staff[i]['Название продовольствия']:  
 print("{}\t{}\t{}\t{}\t{}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во']))  
 fl = True  
 if fl == False:  
 print("Продукция", NazProd, "не найден!")  
  
 elif action == '8':  
 sum = 0  
 for i in range(len(staff)):  
 sum = int(staff[i]['Цена']) \* int(staff[i]['Кол-во'])  
 print("{}\t{}\t{}\t{}\t{}\tСумма: {}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во'], sum))  
  
 elif action == '9':  
 for i in range(len(staff)):  
 print("{}\tНазвание продовольствия: {:13}\tЦена: {:4}\tЕд.изм: {:3}\tКол-во: {:4}\tСклад №{:3}".format(staff[i]['id'], staff[i]['Название продовольствия'],  
 staff[i]['Цена'], staff[i]['Ед.изм'], staff[i]['Кол-во'],staff[i]['Склад №']))  
  
 elif action == '10':  
 k = 0  
 with open(FILENAME, "r", newline="") as file:  
 reader = csv.DictReader(file)  
 for row in reader:  
 staff.append({"id": row["id"].strip(), "Название продовольствия": row["Название продовольствия"].strip(),  
 "Цена": row["Цена"].strip(), "Ед.изм": row["Ед.изм"].strip(),  
 "Кол-во": row["Кол-во"].strip(),"Склад №": row["Склад №"].strip()})  
 k += 1  
 print('Прочитано:', k, 'строк')  
  
 elif action == '11':  
 with open(FILENAME, "w", newline="") as file:  
 columns = ['id', 'Название продовольствия', 'Цена', 'Ед.изм', 'Кол-во', 'Склад №']  
 writer = csv.DictWriter(file, fieldnames=columns)  
 writer.writeheader()  
 # запись нескольких строк  
 writer.writerows(staff)  
 print("Данные добавлены!")  
 elif action == '0':  
 for key, val in menu.items():  
 print(key, '-', val)  
 elif action == 'exit':  
 break