МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**УТВЕРЖДАЮ:**

Зам.директора по УПР

Высшего колледжа ПГТУ «Политехник»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мамедов Б.Ф./

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20\_\_г.

**ОТЧЕТ**

**по учебной практике**

(вид практики)

**в**  Высший колледж ПГТУ «Политехник»

(наименование профильной организации/структурного подразделения ПГТУ)

**Тип практики** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Сроки практики** с 10 октября 2022г.по 23 октября 2022г.

Выполнил обучающийся гр.ИСиП-43

Федорцев Александр Дмитриевич

(Ф.И.О., подпись)

Руководитель практической подготовки от ПГТУ (кафедры/колледжа/филиала)

преподаватель с ученой степенью кандидата наук ВК «Политехник» Илюшов Г.С.

Должность подпись Ф.И.О

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Йошкар-Ола

2022

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc117332127)

[1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ 4](#_Toc117332128)

[2. ПРОГРАММА ПРАКТИКИ 6](#_Toc117332129)

[4. ТЕМА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 10](#_Toc117332130)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 11](#_Toc117332131)

[1.1. Понятие информационной системы, процессы, управленческие функции, состав, структура, классификация информационных систем 11](#_Toc117332132)

[ГЛАВА 2. АНАЛИТИЧЕСКАЯ ЧАСТЬ 14](#_Toc117332133)

[2.1. Анализ предметной области 14](#_Toc117332134)

[2.2. Выбор средств разработки информационной системы 15](#_Toc117332135)

[2.3. Выбор СУБД 15](#_Toc117332136)

[2.4 Описание среды Visual Studio и языка программирования C# 15](#_Toc117332137)

[ЗАКЛЮЧЕНИЕ 18](#_Toc117332138)

[ПРИЛОЖЕНИЕ 19](#_Toc117332139)

# ВВЕДЕНИЕ

Практика проходит в рамках профессионального модуля: ПМ.06 Сопровождение информационных систем

Место практики ВК ПГТУ «Политехник», кабинет 206a

наименование факультета/кафедры (номер кабинета, лаборатории, мастерской), предприятия, организации

Руководителем практики назначен Илюшов Г.С., преподаватель с ученой степенью кандидата наук ВК «Политехник»

(Ф.И.О. должность)

Тема индивидуального задания Информационная система учёта агентов.

Актуальность данной темы: Актуальность разработки программ обусловлена непрерывным совершенствованием информационных технологий и обновлением информационных систем

Сроки практики с «\_10\_» \_октября\_ 2022г. по «\_23\_» \_октября\_ 2022г.

# 1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ

В результате прохождения практики у обучающихся должны быть сформированы умения и приобретен первоначальный практический опыт по виду профессиональной деятельности Сопровождение информационных систем для последующего освоения ими общих и профессиональных компетенций по избранной специальности.

Результатом практики является формирование у обучающихся

- общих компетенций (ОК):

|  |  |
| --- | --- |
| ОК | Наименование результатов практики |
| ОК 1. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 2. | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 3. | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 4. | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 5. | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 6. | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. |
| ОК 7. | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |
| ОК 8. | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. |
| ОК 9. | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках. |
| ОК 11 | Использовать знания по финансовой грамотности |

- профессиональных компетенций (ПК):

|  |  |
| --- | --- |
| ПК | Наименование результатов практики |
| ПК 6.1. | Разрабатывать техническое задание на сопровождение информационной системы. |
| ПК 6.2. | Выполнять исправление ошибок в программном коде информационной системы. |
| ПК 6.3. | Разрабатывать обучающую документацию для пользователей информационной системы. |
| ПК 6.4 | Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания. |
| ПК 6.5 | Осуществлять техническое сопровождение, обновление и восстановление данных информационной системы в соответствии с техническим заданием. |

Результатом практики является развитие у обучающихся

- общих компетенций (ОК):

|  |  |
| --- | --- |
| ОК | Наименование результатов практики |
| ОК 1. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 2. | Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |
| ОК 3. | Планировать и реализовывать собственное профессиональное и личностное развитие. |
| ОК 4. | Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |
| ОК 5. | Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |
| ОК 6. | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. |
| ОК 7. | Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |
| ОК 8. | Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. |
| ОК 9. | Использовать информационные технологии в профессиональной деятельности. |
| ОК 10 | Пользоваться профессиональной документацией на государственном и иностранном языках. |
| ОК 11 | Использовать знания по финансовой грамотности |

- профессиональных компетенций (ПК):

|  |  |
| --- | --- |
| ПК | Наименование результатов практики |
| ПК 6.1. | Разрабатывать техническое задание на сопровождение информационной системы. |
| ПК 6.2. | Выполнять исправление ошибок в программном коде информационной системы. |
| ПК 6.3. | Разрабатывать обучающую документацию для пользователей информационной системы. |
| ПК 6.4 | Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания. |
| ПК 6.5 | Осуществлять техническое сопровождение, обновление и восстановление данных информационной системы в соответствии с техническим заданием. |

# 2. ПРОГРАММА ПРАКТИКИ

|  |  |  |  |
| --- | --- | --- | --- |
| Формируемые компетенции (код, наименование) | Виды работ | Объем часов | Наименование междисциплинарных курсов, дисциплин, входящих в состав профессионального модуля, с указанием конкретных разделов (тем), обеспечивающих выполнение видов работ |
| ПК 6.1. – ПК 6.5  ОК 1– ОК 11 | 1. Разрабатывать техническое задание на сопровождение информационной системы. 2. Выполнять исправление ошибок в программном коде информационной системы. 3. Разрабатывать обучающую документацию для пользователей информационной системы. 4. Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания. 5. Осуществлять техническое сопровождение, обновление и восстановление данных информационной системы в соответствии с техническим заданием. | 72 | МДК.06.01 Внедрение ИС  МДК.06.02 Инженерно-техническая поддержка сопровождения информационных систем  МДК.06.03 Устройство и функционирование информационной системы  МДК.06.04 Интеллектуальные системы и технологии |

**3. СОДЕРЖАНИЕ ИНСТРУКТАЖА ПО ТЕХНИКЕ БЕЗОПАСНОСТИ**

1. Общие требования охраны труда

1.1. Инструкция разработана для работников, занятых эксплуатацией ПЭВМ и ВДТ (далее операторов), работа которых связана с приемом и вводом информации.

1.2. При работе на ПЭВМ могут оказываться следующие опасные и вредные факторы:

— повышенные уровни электромагнитного излучения;

— повышенные уровни ультрафиолетового излучения;

— повышенный уровень инфракрасного излучения;

— повышенный уровень статического электричества;

— повышенный уровень прямой блесткости;

— неравномерность распределения яркости в поле зрения;

— напряжение зрения;

— напряжение внимания;

— эмоциональные нагрузки;

— длительные статические нагрузки;

— монотонность труда;

— большой объем информации;

— интеллектуальные нагрузки.

1.3. К работе оператором, программистом, инженером на персональном компьютере и ВДТ допускаются лица не моложе 18 лет, прошедшие инструктаж на рабочем месте с обязательной регистрацией в журнале и медицинский осмотр, не имеющие противопоказаний.

1.4. На основании требований санитарных правил и норм — СанПиН 2.2.2/2.4.2732-10 – «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». Изменение N 3 к СанПиН 2.2.2/2.4.1340-03 утвержденных Главным государственным санитарным врачом Российской Федерации 03.09.2010, пункт 13.2. Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену) при условии соблюдения гигиенических требований, установленных настоящими Санитарными правилами. Трудоустройство беременных женщин следует осуществлять в соответствии с законодательством Российской Федерации.

1.5. СИЗ оператора являются: белый халат, экранный защитный фильтр класса «полная защита» (если это предусмотрено техническими характеристиками), специальные спектральные очки.

1.6. Работник, допустивший нарушение инструкции по охране труда, может быть привлечен к дисциплинарной ответственности. Если нарушение правил охраны труда связано с причинением имущественного ущерба предприятию, работник несет и материальную ответственность в установленном законом порядке.

1.7. В помещениях, где проводятся работы на ПК, ВДП, необходимо создать оптимальные условия зрительной работы. Освещенность рабочего места при смешанном освещении (в горизонтальной плоскости в зоне размещения клавиатуры и рабочих документов) должна быть в пределах от 300 до 500 Лк. Основной поток естественного света должен быть слева, солнечные лучи и блики не должны попадать в поле зрения работающего и на экраны видеомониторов.

1.8. Монитор ПК должен находиться на расстоянии 50-70 см от глаз оператора и иметь антибликовое покрытие. Освещенность поверхности экрана не должна быть более 300 лк Покрытие должно также обеспечивать снятие электростатического заряда с поверхности экрана, исключать искрение и накопление пыли.

1.9. Нельзя загораживать заднюю стенку системного блока или ставить ПК вплотную к стене, это приводит к нарушению охлаждения системного блока и его перегреву.

1.10. Для повышения влажности воздуха в помещениях с персональным компьютером следует применять увлажнители воздуха, заправляемые ежедневно дистиллированной водой.

1.11. Запрещается проводить ремонт персональных компьютеров непосредственно в рабочих помещениях.

1.12. При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов — не менее 1,2 м.

1.13. Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

1.14. Помещения с персональными компьютерами должно быть оснащено аптечкой первой помощи и углекислотными огнетушителями.

1.15. Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680-800 мм, при отсутствии такой возможности высота рабочей поверхности должна составлять 725 мм.

1.16. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной не менее 500 мм глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм.

1.17. Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах 150 мм и по углу наклонной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

1.18. Рабочее место с персональным компьютером должно быть оснащено легко перемещаемым пюпитром для документов.

1.19. Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, или на специальной регулируемой по высоте рабочей поверхности, отдельно от основной, столешницы.

1.20. В случаях возникновения у работников с персональным компьютером зрительного дискомфорта и других неблагоприятных ощущений, несмотря на соблюдение санитарно-гигиенических требований режимов труда и отдыха, к каждому работнику следует применять индивидуальный подход в ограничении времени работы с компьютером

1.21. Для снятия общего утомления во время перерывов необходимо проводить физкультурные паузы, включающие упражнения общего воздействия, улучшающие функциональное состояние нервной, сердечно-сосудистой, дыхательной систем, а также улучшающих кровообращение, снижающих мышечное утомление.

1.22. Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 — 2,0 м.

1.23. Работающим на ВДТ и ПЭВМ с высоким уровнем напряженности во время регламентированных перерывов и конце рабочего дня должна быть предусмотрена психологическая разгрузка в специально оборудованных помещениях (комната психологической разгрузки).

# 4. ТЕМА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

Тема индивидуального задания: Информационная система учёта агентов.

Основное задание на разработку: создать программный комплекс, предназначенный для управления агентами компании. Необходимо разработать подсистему для работы с агентами компании, которая должна включать в себя следующий функционал:

1. просмотр списка агентов.
2. Фильтрация агентов по типам (ООО, ЗАО и т.п.)
3. Поиск агентов по их наименованию, E-mail или номеру телефона

Также необходимо импортировать и привести данные, выданные в различных форматах к одному единственному формату и импортировать их в локальную базу данных, структура которой записана в виде SQL скрипта.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1. Понятие информационной системы, процессы, управленческие функции, состав, структура, классификация информационных систем

Информационная система (ИС)– взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

Процессы в ИС: ввод информации из внешних или внутренних источников; обработка входной информации и представление ее в удобном виде; вывод информации для представления потребителям или в другую систему; обратная связь.

Управленческие функции:

1. Организационная – разработка организационной структуры комплекса нормативных документов, штатного расписания;
2. Планирование – разработка бизнес-плана для всей фирмы, плана производства;
3. Учетная – разработка и использование форм учета показателей деятельности фирмы: бухучет, финансовый учет, управленческий учет;
4. Анализ – аналитическая функция связана с изучением итогов выполнения планов и заказов, определение влияющих факторов, выявление резервов;
5. Контрольная;
6. Принятие решения.

В состав ИС входят: данные; информация; знания; базы данных (БД); база знаний; программное обеспечение (ПО); экспертные системы; локальные сети; защита информации; информационная безопасность.

Структуру ИС составляет совокупность отдельных ее частей, называемых подсистемами: функциональной и обеспечивающей.

Функциональная часть ИС включает в себя ряд подсистем, охватывающих решение конкретных задач планирования, контроля, учета, анализа и регулирования деятельности управляемых объектов.

Таким образом, структура любой ИС может быть представлена совокупностью обеспечивающих подсистем, среди которых обычно выделяют информационное, техническое, математическое, программное, организационное и правовое обеспечение.

Классификация ИС:

1. По отношению к среде: открытые; закрытые.
2. По происхождению: материальные; абстрактные.
3. По описанию переменных системы: с качественными переменными (имеющие лишь содержательное описание); с количественными переменными (имеющие дискретно или непрерывно описываемые количественным образом переменные); смешанного (количественно-качественное) описания.
4. По типу описания закона (законов) функционирования системы: типа «Черный ящик» (неизвестен полностью закон функционирования системы; известны только входные и выходные сообщения); не параметризованные (закон не описан; описываем с помощью хотя бы неизвестных параметров; известны лишь некоторые априорные свойства закона); параметризованные (закон известен с точностью до параметров и его возможно отнести к некоторому классу зависимостей); типа «Белый (прозрачный) ящик» (полностью известен закон).
5. По управления в системе: управляемые извне системы (без обратной связи, регулируемые, управляемые структурно, информационно или функционально); управляемые изнутри (самоуправляемые или саморегулируемые – программно-управляемые, регулируемые автоматически, адаптируемые, самоорганизующиеся); с комбинированным управлением (автоматические, полуавтоматические, автоматизированные, организационные).
6. По функциональному назначению: производственные, коммерческие, финансовые, маркетинговые.
7. По объектам управления: ИС автоматизированного проектирования, управления технологическими процессами, управления предприятием.
8. По характеру использования результатов информации: информационно-поисковые; информационно-решающие; информационно советующие; информационно управляющие.
9. По степени автоматизации: ручные; автоматизированные; автоматические.
10. По архитектуре: локальные; клиент-серверные; распределенные.
11. По сфере применения: ИС организационного управления; ИС управления технологическими процессами; автоматизированные системы научного исследования; ИС автоматизированного проектирования; автоматизированные обучающие системы; интегрированные ИС; экономические ИС.

# ГЛАВА 2. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 2.1. Анализ предметной области

2.1.1.Диаграмма вариантов использования (англ, usecasediagram) в UML диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.



Рисунок 1 – Диаграмма вариантов использования.

## 2.2. Выбор средств разработки информационной системы

## 2.3. Выбор СУБД

В качестве средства реализации БД была выбрана система MS SQL Server 2019. Microsoft SQL Server 2019 – это мощная система управления данными, обеспечивающая множество функций, защиту данных и высокую производительность для внедренных приложений-клиентов, «легких» веб-приложений и локальных хранилищ данных. SQL Server 2019 предназначен для упрощенного развертывания и быстрого создания прототипов; его можно получить бесплатно и свободно распространять вместе с приложениями.

SQL Server 2019 упрощает развёртывание, управление и оптимизацию данных предприятия и аналитических приложений. Как платформа управления данными предприятия, он предоставляет единую консоль управления, которая позволяет администраторам данных, находящимся в любом месте вашей организации, отслеживать, управлять и настраивать все базы данных и связанные службы по всему предприятию. Он предоставляет расширяемую инфраструктуру управления, которая может быть легко запрограммирована при помощи SQL Server Management Studio, позволяя пользователям переделывать и расширять их среду управления и независимым поставщикам программных продуктов (ISV) создавать дополнительные инструменты и функциональность для дальнейшего расширения возможностей, поставляемых по умолчанию.

## 2.4 Описание среды Visual Studio и языка программирования C#

Интегрированная среда разработки VisualStudio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, VisualStudio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Эта среда позволяет создавать кроссплатформенные проекты на различных языках программирования, таких как VisualBasic (VB), С#, C++, F# и другие. Также она позволяет создавать программы, использующие в своей работе платформу .NET, которая позволяет использовать большой набор сервисов, реализующихся в виде промежуточного, независящего от базовой архитектуры, кода.

Функции MS VS:

1. Средства тестирования. MSVS включает стандартные инструменты тестирования, которые позволяют обеспечить высокое качество программного кода.
2. Разработка БД. MSVS включает инструменты развертывания и управления изменениями, которые позволяют обеспечить постоянную синхронизацию БД и приложения.
3. Отладка и диагностика. Средства анализа кода позволяют избежать типичных ошибок программирования до запуска приложений в производство. Настройка анализа кода в MSVS очень проста благодаря наборам правил, которые предназначены для конкретных сценариев или важных участков кода. Кроме того, метрики обеспечивают простоту и удобство обслуживания создаваемого программного кода.
4. Интегрированная среда разработки. MSVS предоставляет полный набор инструментов управления цифровым рабочим местом. Система включает такие настраиваемые функции, как поддержка нескольких мониторов, которая позволяет наиболее удобно организовать работу.

Язык С#, разработанный компанией Майкрософт, один из самых популярных современных языков программирования. Он востребован на рынке разработки в различных странах, C# применяют при работе с программами для ПК, создании сложных веб-сервисов или мобильных приложений. Появившийся как язык для собственных нужд платформы Microsoft .NET, постепенно этот язык стал очень популярным.

Особенности С#:

1. Полный и хорошо определенный набор основных типов.
2. Встроенная поддержка автоматической генерации XML документации.
3. Автоматическое освобождение динамически распределенной памяти.
4. Возможность отметки классов и методов атрибутами, определяемыми пользователем.
5. Полный доступ к библиотеке базовых классов .NET, а также легкий доступ к WindowsAPI (если это действительно необходимо).
6. Указатели и прямой доступ к памяти, если они необходимы (однако язык разработан таким образом, что практически во всех случаях можно обойтись и без этого).
7. Поддержка свойств и событий в стиле VB.
8. Простое изменение ключей компиляции. Позволяет получать исполняемые файлы или библиотеки компонентов .NET, которые могут быть вызваны другим кодом так же, как элементы управления ActiveX(компоненты СОМ).
9. Возможность использования C# для написания динамических web- страниц ASP.NET.

Версия С# 6.0 содержит множество функций, помогающих повысить производительность труда разработчиков. Общим эффектом этих функций является написание более краткого кода, который также является более удобным для чтения. Синтаксис содержит меньше формальных элементов для многих общих рекомендаций. Благодаря краткости становится проще понять назначение кода.

# ЗАКЛЮЧЕНИЕ

В результате прохождения практики на ВК ПГТУ «Политехник», кабинет 206a

наименование факультета/кафедры, предприятия, организации

сформированы умения и приобретен первоначальный практический опыт по виду профессиональной деятельности для последующего освоения общих и профессиональных компетенций по избранной специальности.

Поставленные цели и задачи практики были выполнены в полном объеме.

# ПРИЛОЖЕНИЕ

Код программы

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Data.SqlClient;

namespace PriyatniyShelestWPF

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

/\*

GLOBAL PROPERTIES

\*/

Agent[] agents;

AgentType[] agentTypes;

SolidColorBrush bgcolor = new SolidColorBrush(Color.FromArgb(0xFF, 0xC6, 0xD7, 0xFF));

SolidColorBrush bgcolor25 = new SolidColorBrush(Color.FromArgb(0xFF, 0x00, 0xFF, 0x00));

static string basePath = AppDomain.CurrentDomain.BaseDirectory;

static string[] config = System.IO.File.ReadAllLines($"{basePath}\\connection.config");

string connStr = $"{config[0]}; {config[1]}; {config[2]}";

int currentPage = 1;

int maxPage = 1;

Lis1t<int> recordsToAlternate = new List<int>();

bool reverseSorting = false;

bool firstLoad = true;

/\*

FUNCTIONS

\*/

public void updateTableConfiguration()

{

string searchString = SearchBox.Text;

int filterIndex = FilterBox.SelectedIndex;

if (filterIndex == -1) { filterIndex = 0; }

int sortIndex = SortBox.SelectedIndex;

if (sortIndex == -1) { sortIndex = 0; }

int totalPages = getMaxPages(connStr, searchFor: searchString, inFilter: filterIndex);

maxPage = totalPages;

int agentsLen = getAgentsFromDB(connStr, onPage: currentPage, searchFor: searchString, inFilter: filterIndex).Length;

agents = new Agent[agentsLen];

agents = getAgentsFromDB(connStr, onPage: currentPage, searchFor: searchString, inFilter: filterIndex, sortBy: sortIndex, reversed: reverseSorting);

currentPageText.Content = ""+currentPage;

centerGrid.RowDefinitions.Clear();

centerGrid.Children.Clear();

for (int i = 0; i < agentsLen; i++)

{

SolidColorBrush usingBgColor;

if (agents[i] == null) { break; }

if (agents[i].getDiscount() == 25)

{ usingBgColor = bgcolor25; }

else { usingBgColor = bgcolor; }

{

Grid innerGrid = new Grid();

Grid descriptionsGrid = new Grid();

Grid discountGrid = new Grid();

Border agentDataBorder = new Border();

Border agentLogoBorder = new Border();

Image agentLogo = new Image();

Label agentDiscountLabel = new Label();

Label agentTypeNameLabel = new Label();

Label agentSalesLabel = new Label();

Label agentContactsLabel = new Label();

Label agentPriorityLabel = new Label();

Button editAgentButton = new Button();

Button deleteAgentButton = new Button();

CheckBox agentSelection = new CheckBox();

void addToAlterList(object sender, EventArgs e)

{

recordsToAlternate.Add(agents[(int)deleteAgentButton.Tag].ID);

MessageBox.Show(recordsToAlternate[0]+"");

}//functions for buttons and checkers

void removeFromAlterList(object sender, EventArgs e)

{

MessageBox.Show(recordsToAlternate[0] + "");

recordsToAlternate.Remove(agents[(int)deleteAgentButton.Tag].ID);

}//functions for buttons and checkers

void delAgent(object sender, EventArgs e)

{

MessageBox.Show($"Delete {agents[(int)deleteAgentButton.Tag].Title} ?");

}//functions for buttons and checkers

void alterAgent(object sender, EventArgs e)

{

AgentsEditWindow editWindow = new AgentsEditWindow();

editWindow.currentAgent = agents[(int)deleteAgentButton.Tag];

editWindow.isAddingNew = false;

editWindow.updateValues(agentTypes);

editWindow.ShowDialog();

}//functions for buttons and checkers

{

innerGrid.HorizontalAlignment = HorizontalAlignment.Stretch;

descriptionsGrid.VerticalAlignment = VerticalAlignment.Stretch;

descriptionsGrid.Margin = new Thickness(5);

discountGrid.Margin = new Thickness(5);

ColumnDefinition imageColumn = new ColumnDefinition();

imageColumn.Width = new GridLength(200);

innerGrid.ColumnDefinitions.Add(imageColumn);

ColumnDefinition descriptionColumn = new ColumnDefinition();

descriptionColumn.Width = new GridLength(1, GridUnitType.Star);

innerGrid.ColumnDefinitions.Add(descriptionColumn);

ColumnDefinition propertiesColumn = new ColumnDefinition();

propertiesColumn.Width = new GridLength(200);

innerGrid.ColumnDefinitions.Add(propertiesColumn);

{

RowDefinition TypeRow = new RowDefinition();

TypeRow.Height = new GridLength(30);

descriptionsGrid.RowDefinitions.Add(TypeRow);

RowDefinition TitleRow = new RowDefinition();

TitleRow.Height = new GridLength(30);

descriptionsGrid.RowDefinitions.Add(TitleRow);

RowDefinition PhoneRow = new RowDefinition();

PhoneRow.Height = new GridLength(30);

descriptionsGrid.RowDefinitions.Add(PhoneRow);

RowDefinition PriorityRow = new RowDefinition();

PriorityRow.Height = new GridLength(30);

descriptionsGrid.RowDefinitions.Add(PriorityRow);

}//description column

{

ColumnDefinition propsColumnOne = new ColumnDefinition();

propsColumnOne.Width = new GridLength(1, GridUnitType.Star);

discountGrid.ColumnDefinitions.Add(propsColumnOne);

ColumnDefinition propsColumnTwo = new ColumnDefinition();

propsColumnTwo.Width = new GridLength(1, GridUnitType.Star);

discountGrid.ColumnDefinitions.Add(propsColumnTwo);

RowDefinition discountRow = new RowDefinition();

discountRow.Height = new GridLength(1, GridUnitType.Star);

discountGrid.RowDefinitions.Add(discountRow);

RowDefinition propsRowOne = new RowDefinition();

propsRowOne.Height = new GridLength(1, GridUnitType.Star);

discountGrid.RowDefinitions.Add(propsRowOne);

}//discount column

Grid.SetColumn(descriptionsGrid, 1);

Grid.SetColumn(discountGrid, 2);

Grid.SetRow(innerGrid, i);

}// record creation

{

agentDataBorder.Background = usingBgColor;

agentLogoBorder.Background = usingBgColor;

Grid.SetColumn(agentLogoBorder, 0);

Grid.SetColumn(agentDataBorder, 1);

innerGrid.Children.Add(agentLogoBorder);

innerGrid.Children.Add(agentDataBorder);

}// border setting

{

Uri fileUri = new Uri(basePath + agents[i].Logo);

agentLogo.Source = new BitmapImage(fileUri);

agentLogo.Margin = new Thickness(10);

Grid.SetColumn(agentLogo, 0);

innerGrid.Children.Add(agentLogo);

}// image setting

{

agentTypeNameLabel.Content = $"{agents[i].AgentType} | {agents[i].Title} ";

Grid.SetRow(agentTypeNameLabel, 0);

agentSalesLabel.Content = $"Продажи: {agents[i].Sales} ";

Grid.SetRow(agentSalesLabel, 1);

agentContactsLabel.Content = $"Телефон: {agents[i].Phone} Email: {agents[i].Email}";

Grid.SetRow(agentContactsLabel, 2);

agentPriorityLabel.Content = $"Приоритетность: {agents[i].Priority} ";

Grid.SetRow(agentPriorityLabel, 3);

descriptionsGrid.Children.Add(agentTypeNameLabel);

descriptionsGrid.Children.Add(agentSalesLabel);

descriptionsGrid.Children.Add(agentContactsLabel);

descriptionsGrid.Children.Add(agentPriorityLabel);

}// agentDescription setting

{

int discount = agents[i].getDiscount();

Thickness thicc = new Thickness(2);

Grid.SetColumn(agentDiscountLabel, 0);

Grid.SetRow(agentDiscountLabel, 0);

agentDiscountLabel.Content = $"{discount}%";

discountGrid.Children.Add(agentDiscountLabel);

Grid.SetColumn(agentSelection, 1);

Grid.SetRow(agentSelection, 0);

if (recordsToAlternate.Contains(agents[i].ID)) { agentSelection.IsChecked = true; }

agentSelection.HorizontalAlignment = HorizontalAlignment.Stretch;

agentSelection.VerticalAlignment = VerticalAlignment.Stretch;

agentSelection.HorizontalContentAlignment = HorizontalAlignment.Center;

agentSelection.Checked += addToAlterList;

agentSelection.Unchecked += removeFromAlterList;

agentSelection.Margin = thicc;

discountGrid.Children.Add(agentSelection);

Grid.SetColumn(editAgentButton, 0);

Grid.SetRow(editAgentButton, 1);

editAgentButton.Content = "Изменить";

editAgentButton.Tag = i;

editAgentButton.Click += alterAgent;

editAgentButton.Margin = thicc;

discountGrid.Children.Add(editAgentButton);

Grid.SetColumn(deleteAgentButton, 1);

Grid.SetRow(deleteAgentButton, 1);

deleteAgentButton.Content = "Удалить";

deleteAgentButton.Margin = thicc;

deleteAgentButton.Name = $"delBtn\_{i}";

deleteAgentButton.Tag = i;

deleteAgentButton.Click += delAgent;

discountGrid.Children.Add(deleteAgentButton);

}// agentDiscountGrid setting

{

RowDefinition recordRow = new RowDefinition();

recordRow.Height = new GridLength(150);

recordRow.Name = $"row{i}";

centerGrid.RowDefinitions.Add(recordRow);

innerGrid.Children.Add(discountGrid);

innerGrid.Children.Add(descriptionsGrid);

discountGrid.UpdateLayout();

descriptionsGrid.UpdateLayout();

innerGrid.UpdateLayout();

centerGrid.Children.Add(innerGrid);

}// inserting ready row in grid

}

} // creating records on page

centerGrid.UpdateLayout();

}

public int getMaxPages(string connectionString, string searchFor = "%", int inFilter = 0, int recordsPerPage = 10)

{

if (searchFor == "") { searchFor = "%"; } //Search all

string searchForQuery = $"AND ((Agent.Title LIKE '%{searchFor}%' OR Agent.Email LIKE '%{searchFor}%' OR Agent.Phone LIKE '%{searchFor}%')) ";

string filter;

if (inFilter == 0) { filter = "%"; }

else { filter = agentTypes[inFilter].ID.ToString(); }

int salesForYears = 10;

double maxPages;

int records = 0;

string queryString = "SELECT Agent.ID, AgentType.Title AS 'Type', Agent.Title, " +

"Agent.[Address], Agent.INN, Agent.KPP, " +

"Agent.DirectorName, Agent.Phone, Agent.[Priority], " +

"Agent.Email, Agent.Logo, " +

"(SELECT ISNULL(SUM(ProductSale.ProductCount), 0) " +

"FROM ProductSale " +

$"WHERE ProductSale.AgentID = Agent.ID AND DATEDIFF(YEAR, ProductSale.SaleDate, CURRENT\_TIMESTAMP) < {salesForYears}) AS 'Sales', " +

"(SELECT ISNULL(SUM(ProductSale.ProductCount \* Product.MinCostForAgent), 0) " +

"FROM ProductSale, Product " +

$"WHERE ProductSale.AgentID = Agent.ID AND ProductSale.ProductID = Product.ID AND DATEDIFF(YEAR, ProductSale.SaleDate, CURRENT\_TIMESTAMP) < {salesForYears}) AS 'TotalSalesBy'" +

$"FROM Agent INNER JOIN AgentType ON(Agent.AgentTypeID = AgentType.ID) AND AgentType.ID LIKE '{filter}' " + searchForQuery;

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

records++;

}

reader.Close();

} // counts records

maxPages = Math.Ceiling((float)records / 10);

return (int)maxPages;

}

public void updateFilterTypes()

{

string connectionString = connStr;

agentTypes = getAgentTypes(connectionString);

FilterBox.Items.Clear();

FilterBox.Items.Add("Все типы");

for(int i = 1; i < agentTypes.Length; i++)

{

FilterBox.Items.Insert(agentTypes[i].ID, agentTypes[i].Title);

}

FilterBox.SelectedIndex = 0;

}

AgentType[] getAgentTypes(string connectionString)

{

string queryString = "SELECT COUNT(\*) FROM AgentType";

int LenOfTypes = 1;

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

LenOfTypes = reader.GetInt32(0);

}

reader.Close();

}

AgentType[] Type = new AgentType[LenOfTypes + 1];

Type[0] = new AgentType();

Type[0].ID = 0;

Type[0].Title = "%";

queryString = "SELECT AgentType.ID, AgentType.Title FROM AgentType";

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

int id = 1;

while (reader.Read())

{

Type[id] = new AgentType{

ID = reader.GetInt32(0),

Title = reader.GetString(1)

};

id++;

};

reader.Close();

}

return Type;

}

Agent[] getAgentsFromDB(string connectionString, string searchFor="%", int inFilter=0, int sortBy=0, int onPage = 1, int recordsPerPage = 10, bool reversed=false)

{

if (searchFor == "") { searchFor = "%"; } //Search all

string searchForQuery = $"AND ((Agent.Title LIKE '%{searchFor}%' OR Agent.Email LIKE '%{searchFor}%' OR Agent.Phone LIKE '%{searchFor}%')) ";

string sortType = "Agent.Title ";

string paginatorSelectionQuery = $"OFFSET {(onPage \* recordsPerPage) - recordsPerPage} ROWS FETCH NEXT {recordsPerPage} ROWS ONLY ";

string filter;

if (inFilter == 0) { filter = "%"; }

else { filter = agentTypes[inFilter].ID.ToString(); }

switch (sortBy)

{

case 0: { sortType = "Agent.Title "; ; break; }

case 1: { sortType = "TotalSalesBy "; break; } // Discount

case 2: { sortType = "Agent.Priority "; break; }

}

if (reversed) { sortType += "DESC "; }

else { sortType += "ASC "; }

int salesForYears = 10;

string queryString = "SELECT Agent.ID, AgentType.Title AS 'Type', Agent.Title, " +

"Agent.[Address], Agent.INN, Agent.KPP, " +

"Agent.DirectorName, Agent.Phone, Agent.[Priority], " +

"Agent.Email, Agent.Logo, " +

"(SELECT ISNULL(SUM(ProductSale.ProductCount), 0) " +

"FROM ProductSale " +

$"WHERE ProductSale.AgentID = Agent.ID AND DATEDIFF(YEAR, ProductSale.SaleDate, CURRENT\_TIMESTAMP) < {salesForYears}) AS 'Sales', " +

"(SELECT ISNULL(SUM(ProductSale.ProductCount \* Product.MinCostForAgent), 0) " +

"FROM ProductSale, Product " +

$"WHERE ProductSale.AgentID = Agent.ID AND ProductSale.ProductID = Product.ID AND DATEDIFF(YEAR, ProductSale.SaleDate, CURRENT\_TIMESTAMP) < {salesForYears}) AS 'TotalSalesBy'" +

$"FROM Agent INNER JOIN AgentType ON(Agent.AgentTypeID = AgentType.ID) AND AgentType.ID LIKE '{filter}' " + searchForQuery +

"ORDER BY " + sortType + paginatorSelectionQuery;

Agent[] recivedAgents = new Agent[recordsPerPage];

using (SqlConnection connection = new SqlConnection(connectionString))

{

SqlCommand command = new SqlCommand(queryString, connection);

connection.Open();

SqlDataReader reader = command.ExecuteReader();

int id = 0;

while (reader.Read())

{

recivedAgents[id] = new Agent

{

ID = reader.GetInt32(0),

AgentType = reader.GetString(1),

Title = reader.GetString(2),

Address = reader.GetString(3),

INN = reader.GetString(4),

KPP = reader.GetString(5),

DirectorName = reader.GetString(6),

Phone = reader.GetString(7),

Priority = reader.GetInt32(8),

Email = reader.GetString(9),

Logo = reader.GetString(10),

Sales = reader.GetInt32(11),

TotalSalesBy = reader.GetDecimal(12)

};

id++;

}

reader.Close();

} // importing agents into array

return recivedAgents;

}

public void reverseTextCheck()

{

if (reverseSorting) { ChangeOrderButton.Content = "↓↓"; }

else { ChangeOrderButton.Content = "↑↑"; }

}

/\*

EVENT HANDLERS

\*/

private void ChangeOrderBtn\_Click(object sender, RoutedEventArgs e)

{

reverseSorting = !reverseSorting;

reverseTextCheck();

{ currentPage = 1; updateTableConfiguration(); reverseTextCheck(); }

}

private void FilterSource\_Changed(object sender, RoutedEventArgs e)

{

if (firstLoad == false)

{ currentPage = 1; updateTableConfiguration(); reverseTextCheck(); }

}

private void pagintorSelectionUP(object sender, RoutedEventArgs e)

{

if (currentPage != maxPage)

{ currentPage++; updateTableConfiguration(); }

}

private void pagintorSelectionDOWN(object sender, RoutedEventArgs e)

{

if (currentPage != 1)

{ currentPage--; updateTableConfiguration(); }

}

private void window\_init(object sender, EventArgs e)

{

firstLoad = !firstLoad;

{ updateFilterTypes(); updateTableConfiguration(); reverseTextCheck(); }

}

}

}



Рис.1 – Интерфейс программы

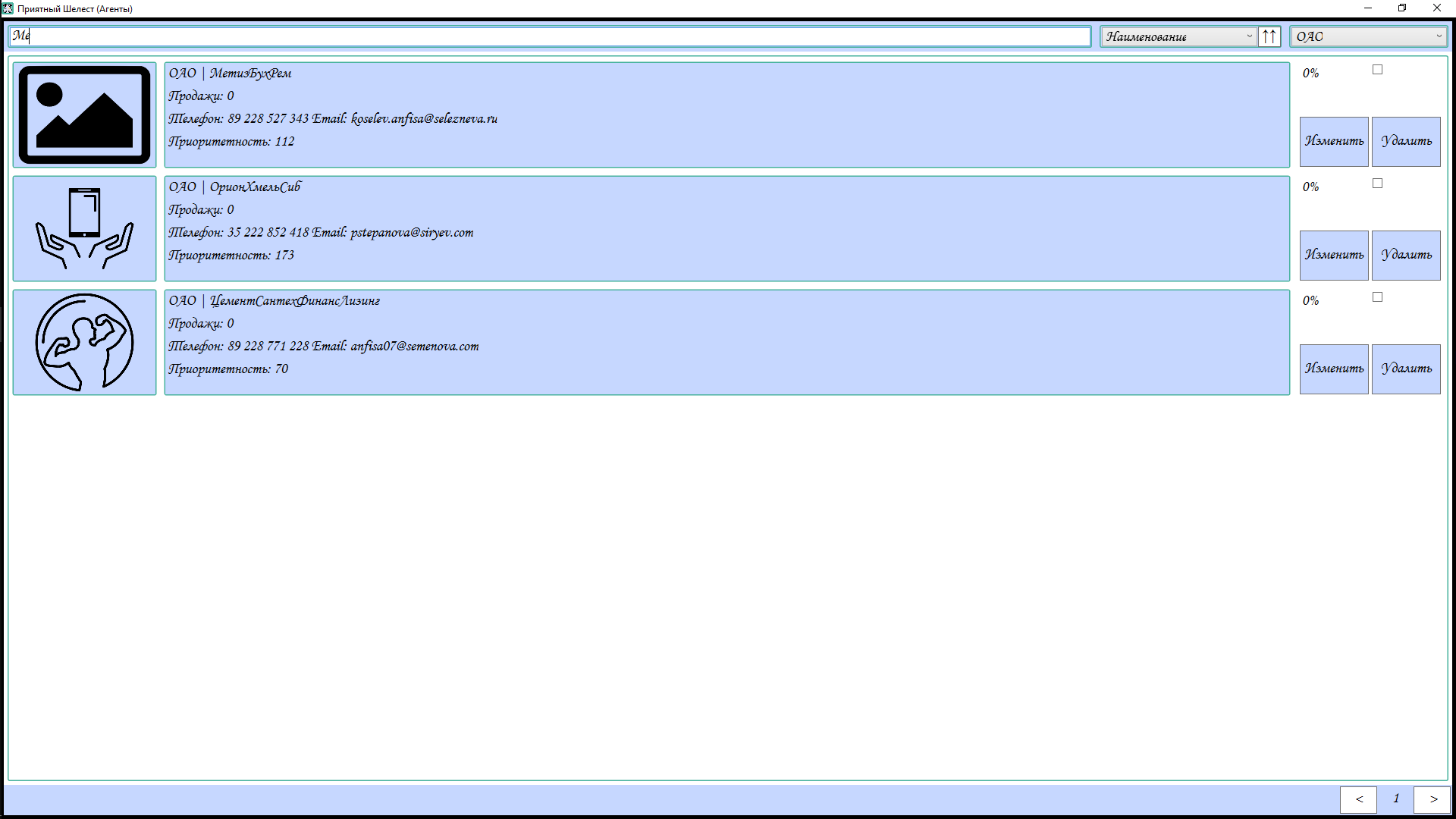


Рис.2 - Фильтр и поиск агентов

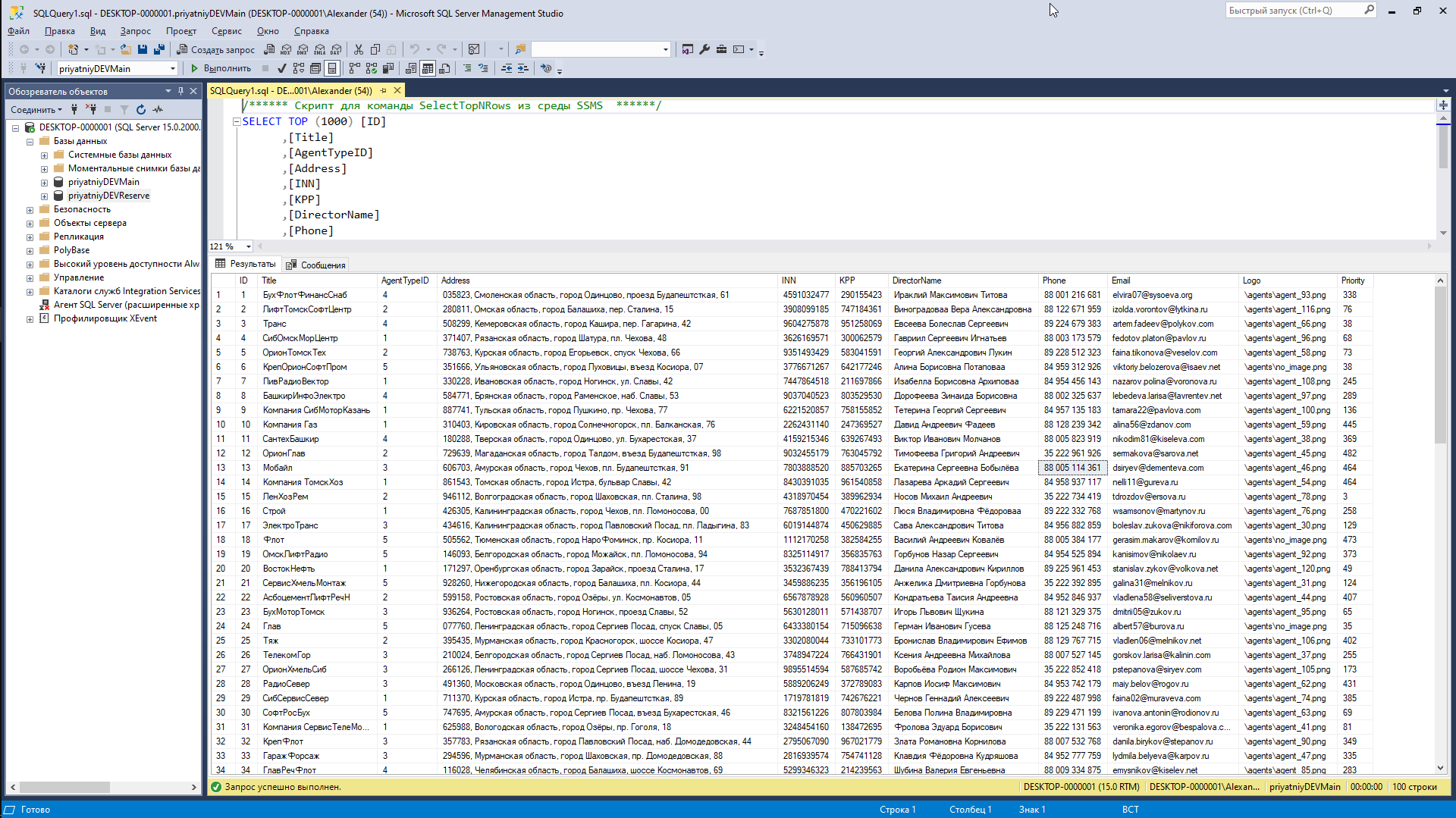


Рис.3 –Импортированные данные в БД

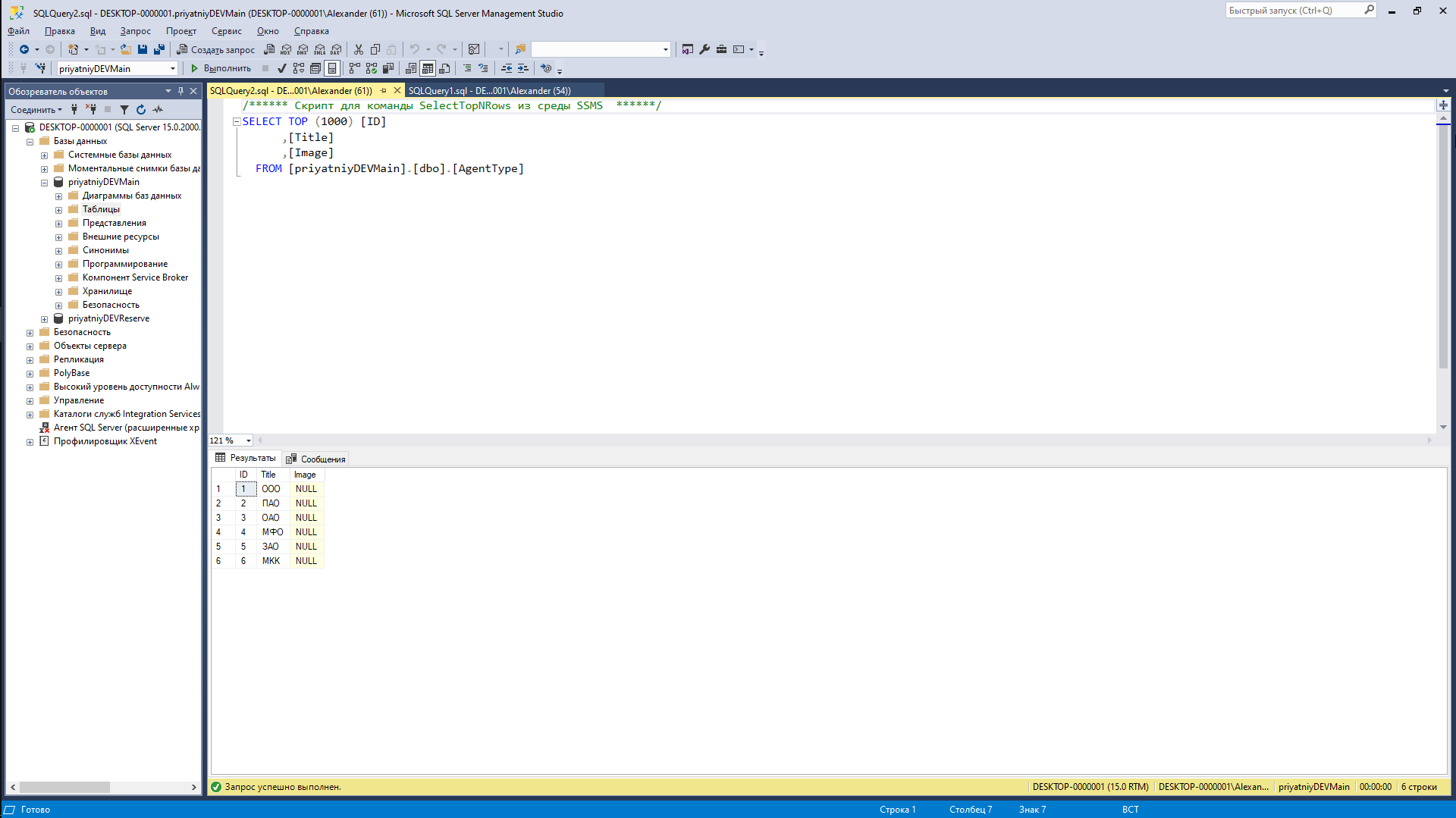


Рис.4 –Импортированные данные в БД

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Аттестационный лист прохождения практики**

*(Заполненный аттестационный лист прилагается к дневнику практики)*

Обучающийся Федорцев Александр Дмитриевич

(фамилия, имя, отчество)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код и  наименование компетенции (ПК и ОК) | Критерии оценивания | | | |
| не сформированы | сформированы частично | сформированы в достаточном объеме | сформированы полностью |
| ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |  |  |  |  |
| ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. |  |  |  |  |
| ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие. |  |  |  |  |
| ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. |  |  |  |  |
| ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. |  |  |  |  |
| ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. |  |  |  |  |
| ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. |  |  |  |  |
| ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. |  |  |  |  |
| ОК 9. Использовать информационные технологии в профессиональной деятельности. |  |  |  |  |
| ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках. |  |  |  |  |
| ОК 11 Использовать знания по финансовой грамотности |  |  |  |  |
| ПК 6.1. Разрабатывать техническое задание на сопровождение информационной системы. |  |  |  |  |
| ПК 6.2. Выполнять исправление ошибок в программном коде информационной системы. |  |  |  |  |
| ПК 6.3. Разрабатывать обучающую документацию для пользователей информационной системы. |  |  |  |  |
| ПК 6.4. Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания. |  |  |  |  |
| ПК 6.5. Осуществлять техническое сопровождение, обновление и восстановление данных информационной системы в соответствии с техническим заданием. |  |  |  |  |

*Примечание: Укажите уровень освоения каждой компетенции, который, на Ваш взгляд, проявил обучающийся в период прохождения практики.*

Оценка результатов прохождения практики руководителем практической подготовки от профильной организации/структурного подразделения ПГТУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практической подготовки от профильной организации/структурного подразделения ПГТУ

преподаватель с ученой степенью кандидата наук ВК «Политехник» \_\_\_\_\_\_\_ Илюшов Г.С

Должность подпись Ф.И.О

МП

Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| УТВЕРЖДАЮ  Зам. директора по УПР  Высшего колледжа ПГТУ «Политехник»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Мамедов Б.Ф./  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.  Место печати | «УТВЕРЖДАЮ  Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (название профильной организации, структурного подразделения ПГТУ)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, Ф.И.О.)  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_  Место печати |

**ДНЕВНИК П Р А К Т И К И**

Вид практики учебная практика

Тип практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обучающийся Федорцев Александр Дмитриевич

(фамилия, имя, отчество)

Образовательное структурное подразделение Высший колледж ПГТУ «Политехник»

(институт / факультет/ колледж/ филиал)

Специальность (направление подготовки) 09.02.07 Информационные системы и программирование

Форма обучения очное

Курс 4 Группа ИСиП-43

Место прохождения Высший колледж ПГТУ «Политехник»

(наименование профильной организации/структурного подразделения ПГТУ)

Сроки практики с «\_10\_»\_октября\_2022г. по «\_23\_»\_октября 2022г.

Основание:

Приказ по университету/филиалу № от «\_\_\_» 20\_\_\_г.

Договор о проведении практической подготовки № от «\_\_\_\_» 20\_\_\_г.

Йошкар-Ола

2022

**ВВОДНЫЙ ИНСТРУКТАЖ**

**по ознакомлению с требованиями охраны труда, техники   
безопасности, пожарной безопасности (на кафедре/ в колледже / в филиале)**

заместитель директора по УПР Высшего колледжа ПГТУ «Политехник» - Мамедов Б.Ф.

(должность, Ф.И.О. ответственного лица)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_10\_» \_\_\_октября\_2022г.

(подпись) (дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_10\_» \_\_\_октября\_2022г.

(подпись обучающегося) (дата)

**ИНСТРУКТАЖ**

**по ознакомлению с требованиями охраны труда, техники**

**безопасности, пожарной безопасности** **в профильной организации**/ **структурном подразделении ПГТУ**

Преподаватель с ученой степенью кандидата наук ВК «Политехник» Илюшов Григорий Сергеевич

(должность, Ф.И.О. ответственного лица)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_10\_» \_\_октября\_\_2022г.

(подпись) (дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_10\_» \_\_\_октября\_2022г.

(подпись обучающегося) (дата)

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОХОЖДЕНИЕ ПРАКТИКИ**

Создание программного комплекса, предназначенного для управления агентами компании. Разработать подсистему для работы с агентами компании.

**план ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ   
ПРОХОЖДЕНИЯ практики**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Содержание работы | Период выполнения работы | Отметка о выполнении работы  (выполнено /  не выполнено) | Подпись руководителя практической подготовки от профильной организации /структурного подразделения ПГТУ |
| 1 | 2 | 4 | 5 | 6 |
|  | Преобразование данных к необходимому формату | 10.10.2022 | Выполнено |  |
|  | Преобразование данных к необходимому формату | 11.10.2022 | Выполнено |  |
|  | Подготовка данных к импорту | 12.10.2022 | Выполнено |  |
|  | Импорт данных в базу данных | 13.10.2022 | Выполнено |  |
|  | Проектирование структуры приложения | 15.10.2022 | Выполнено |  |
|  | Создание приложения | 17.10.2022 | Выполнено |  |
|  | Создание приложения | 18.10.2022 | Выполнено |  |
|  | Создание приложения | 19.10.2022 | Выполнено |  |
|  | Создание приложения | 20.10.2022 | Выполнено |  |
|  | Подготовка отчета | 22.10.2022 | Выполнено |  |
|  |  |  |  |  |
|  |  |  |  |  |

**РЕЗУЛЬТАТЫ ПРОХОЖДЕНИЯ ПРАКТИКИ**

***Результаты прохождения практики отражаются в Аттестационном листе прохождения практики, который оформляется отдельно по установленной форме, подписывается у руководителя практической подготовки от профильной организации/ структурного подразделения ПГТУ.***

**ТАБЕЛЬ ПОСЕЩАЕМОСТИ**

Ф.И.О обучающегося Федорцев Александр Дмитриевич

Место прохождения практики Высший колледж ПГТУ «Политехник» каб.206а

наименование факультета/кафедры (номер кабинета, лаборатории, мастерской), предприятия, организации

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата практики | 10.10.2022 | 11.10.2022 | 12.10.2022 | 13.10.2022 | 15.10.2022 | 17.10.2022 | 18.10.2022 | 19.10.2022 | 20.10.2022 | 22.10.2022 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Часов | 8 | 8 | 8 | 8 | 4 | 8 | 8 | 8 | 8 | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Руководитель практики от организации /Илюшов Г.С./

подпись Ф.И.О.

Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_г.

М.П.

\*Табель заполняется на каждый день

6 – количество часов, проведенное на рабочем месте (от 1 до 8)

Б – болеет (при наличии мед. справки)

Х – отсутствует на рабочем месте без уважительной причины

ОТ – отсутствует на рабочем месте по уважительной причине