Частное учреждение образования

«Колледж бизнеса и права»

Отчёт

по производсТвенной технологической практике

ОП Т.691012

Руководитель практики от колледжа (Е.В. Багласова)

Руководитель практики от предприятия (Е.В. Нехай)

М.П.

Учащийся (Н.В. Карсюк)

2020

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

ОП Т.691012

Разраб.

Карсюк Н.В.

Провер.

Багласова Е.В.

Т.контр.

Н.контр.

Утверд.

Отчёт по технологической практике

Лит

Листов

50

КБП

У

Содержание

[1 Структура предприятия, характеристика основных видов деятельности 3](#_Toc36632880)

[2 Должностные обязанности техника-программиста и инженера-программиста 5](#_Toc36632881)

[3 Охрана труда и окружающей среды на предприятии 6](#_Toc36632882)

[3.1 Правовые, нормативные, социально-экономические и организационные вопросы охраны труда. 6](#_Toc36632883)

[3.2 Разработка мер по нормализации и защите от инфракрасного излучения 9](#_Toc36632884)

[3.3 Пожарная безопасность 12](#_Toc36632885)

[3.4 Охрана окружающей среды 14](#_Toc36632886)

[4 План мероприятий по разработке и внедрению программы 16](#_Toc36632887)

[5 Техно-рабочий проект 17](#_Toc36632888)

[5.1 Исследование предметной области 17](#_Toc36632889)

[5.2 Структура входных и выходных данных 17](#_Toc36632890)

[5.3 Проектирование модели 17](#_Toc36632891)

[5.4 Концептуальный прототип 24](#_Toc36632892)

[5.5 Реализация функций 25](#_Toc36632893)

[5.6 Функциональное тестирование 26](#_Toc36632894)

[6 Опытная эксплуатация 32](#_Toc36632895)

[6.1 Ошибки, выявленные в процессе опытной эксплуатации 32](#_Toc36632896)

[Выводы 33](#_Toc36632897)

[Список информационных источников 34](#_Toc36632898)

[Приложение А 35](#_Toc36632899)

[(обязательное) 35](#_Toc36632900)

[Текст программы 35](#_Toc36632901)

1 Структура предприятия, характеристика основных видов деятельности

Колледж бизнеса и права является частным учреждением среднего специального образования. Деятельность колледжа осуществляется в соответствии с законодательством Республики Беларусь в сфере образования и Уставом колледжа, утвержденного общим собранием ООО «Бел ТБП» (протокол от 10 апреля 2012 года № 30).

В структуру колледжа входят четыре отделения, вычислительный центр, библиотека, бухгалтерия, архив. В колледже оборудованы 63 учебных кабинета, 4 тренажерных зала, актовый зал, зал заседаний; функционируют медпункт, столовая, буфет, методический кабинет, кабинет социально-педагогической и психологической службы. Колледж имеет филиал, расположенный в городе Бресте по адресу пл. Свободы,13. В колледже обучается порядка 1600 учащихся дневной формы и около 600 – заочной. Образовательный процесс осуществляют 102 преподавателя с высшим образованием, из которых 21 человек имеют высшую квалификационную категорию, 26 – первую, 22 – вторую.

С целью организации практики учащихся заключены договоры с рядом организаций г. Минска, среди которых ОАО «Минский автомобильный завод», РУП «Минский тракторный завод», ОАО «Кристалл», строительный гипермаркет «Материк», ООО «Торговый дом «На Немиге», Минский городской суд, Главное управление юстиции Мингорисполкома, ОАО «Приорбанк», СОАО «Белгазпромбанк» и др.

Идеологическая и воспитательная работа ведется по следующим направлениям: идеологическое и гражданско-патриотическое; нравственно-правовое; гендерное и семейное; трудовое, профессиональное и экологическое; воспитание культуры безопасной жизнедеятельности и здорового образа жизни; воспитание психологической культуры. В колледже созданы первичные организации ОО «БРСМ», ОО «Белая Русь», профсоюзная организация, волонтерский отряд «Доброе сердце». Учреждение образования сотрудничает с Белорусским общественным объединением «Красный Крест», с православным приходом Храма святого Николая Японского, с учреждением «Центр экологических решений», участвует в благотворительных акциях для детей-инвалидов и детей-сирот. В целях организации внеурочной деятельности учащихся работает 14 кружков различной направленности.

Организация имеет следующую структуру, представленную на рисунке 1.1.

Директор

Заместитель директора по учебной работе

Заместитель директора по идеологической и воспитательной работе

Заместитель директора по производственной работе

 Экономическое отделение

 Отделение информационных технологий

 Юридическое отделение

 Заочное отделение

Советник колледжа

Заместитель директора по учебной работе

Рисунок 1.1 – Организационная структура

Структура управления дает представление о том, какие подразделения, службы или отдельные исполнители входят в аппарат управления, как они специализированы и каковы их соподчиненность и взаимосвязи.

Практика проходит в вычислительном центре в ЧУО «Колледж бизнеса и права». Вычислительный центр (ВЦ) - это предприятие, предназначенное для выполнения сложных и трудоёмких вычислительных работ с помощью ЭВМ. Различают ВЦ общего назначения, ВЦ для обработки экономической информации и ВЦ для управления технологическими процессами. Вычислительные центры для управления технологическими процессами не имеют больших подразделений программистов или экономистов, так как составы задач и программ этих ВЦ определяют заранее и не меняют в процессе работы. Здесь основная часть работников — инженеры и техники по обслуживанию ЭВМ и аппаратуры автоматической связи с объектами управления. Потому штат в этом отделе состоит из заведующей, инженера программиста и техника программиста. Вычислительным центром обслуживается более 10-ти компьютерных классов.

2 Должностные обязанности техника-программиста и инженера-программиста

Техник-программист обязан:

* выполнять работу по обеспечению механизированной и автоматизированной обработки поступающей информации;
* принимать участие в отладке, разработанных программ;
* составлять простые схемы технологического процесса обработки информации;
* разрабатывать программные решения простых задач, проводить их отладку и экспериментальную проверку отдельных этапов работ;
* участвовать в выполнении различных операций технологического процесса обработки информации;
* вести и представлять установленную отчетность;
* участвовать в совершенствовании системы менеджмента качества;
* соблюдать на рабочем месте требования по охране труда и технике безопасности.

Инженер-программист обязан:

* участвовать в оказании консультационных услуг заказчикам по доработке программного обеспечения;
* проводить проверку программ на основе логического анализа;
* участвовать в сопровождении программного обеспечения;
* участвовать в сопровождение внедренных программ и программных средств;
* принимать участие в презентации программного обеспечения;
* соблюдать график выполнения работ по договорам, согласно управленческой базе;
* вести и представлять установленную отчетность;
* участвовать в совершенствовании системы менеджмента качества;
* соблюдать на рабочем месте требования по охране труда и технике безопасности.

3 Охрана труда и окружающей среды на предприятии

3.1 Правовые, нормативные, социально-экономические и организационные вопросы охраны труда.

Система управления охраны труда (СУОТ) – это регламентированная законодательством совокупность методов и средств управления, направленных на организацию обеспечения безопасности, сохранения жизни, здоровья и работоспособности человека в процессе труда.

ЧУО «Колледж бизнеса и права» предоставляет услуги в сфере образования, что подразумевает отсутствие опасностей, рисков гибели и травмирования работников на производстве.

Создание и обеспечение функционирования системы управления охраной труда – обязанность каждого работодателя. В соответствии со статьей 17 Закона Республики Беларусь «Об охране труда», наниматель несет обязанности по разработке, внедрению и поддержанию системы управления охраной труда, разработке и реализации мероприятий по улучшению условий и охраны труда.

В организации ЧУО «Колледж бизнеса и права» директором создается служба охраны труда. В ее состав на паритетной основе входят уполномоченные работники.

Комитет или служба по охране труда организует разработку раздела коллективного договора (соглашения) об охране труда, совместные действия работодателя и работников по обеспечению требований, предупреждению производственного травматизма и профессиональных заболеваний, а также проведение проверок условий и охраны труда на рабочих местах и информирование работников о результатах указанных проверок.

На должности специалистов по охране труда назначаются лица в соответствии с квалификационными требованиями, предусмотренными законодательством Республики Беларусь.

Проверка знаний по охране труда должна проводиться инженером по охране труда или уполномоченным лицом, выполняющим его функции.

Проверка знаний по вопросам охраны труда работников службы и лиц, на которых возложены ее обязанности, проводится в установленном порядке не позднее одного месяца со дня назначения на должность, возложения исполнения этих обязанностей. В последующем такая проверка проводится с периодичностью не реже одного раза в три года.

Объектом управления охраной труда является деятельность ЧУО «Колледж бизнеса и права». Обязанностями в области охраны труда являются:

* соблюдать требования по охране труда, а также правила поведения на территории организации, вспомогательных и бытовых помещениях;
* выполнять нормы и обязательства по охране труда, предусмотренные коллективным договором, соглашением, трудовым договором, должностными обязанностями и правилами внутреннего трудового распорядка;
* проходить в установленном законодательством порядке медицинские осмотры, обучение, стажировку, инструктаж и проверку знаний по вопросам охраны труда;
* оказывать содействие и сотрудничать с нанимателем в деле обеспечения здоровых и безопасных условий труда, немедленно извещать своих непосредственных руководителей или иных уполномоченных должностных лиц о неисправности оборудования, инструмента, приспособлений, транспортных средств, средств защиты, об ухудшении состояния своего здоровья;
* немедленно сообщать о любой ситуации, угрожающей жизни или здоровью работников и окружающих, несчастном случае, оказывать содействие в принятии мер по оказанию необходимой помощи потерпевшим и доставке их в организацию здравоохранения;
* исполнять иные обязанности, предусмотренные законодательством об охране труда.

Система управления охраной труда на ЧУО «Колледж бизнеса и права» стремится к обеспечению безопасности условий труда на основе соблюдения действующего законодательства и технических стандартов. Колледж разрабатывает и вводит в действие стандарты в области охраны труда, которые содержат высокие требования к безопасности. Каждый из работников компании несет ответственность за соблюдение условий безопасности и технических стандартов. При реализации кадровой политики, политики оплаты труда, политики социального обеспечения запрещены любые преференции по национальности, полу, возрасту.

Организационно-методическую работу по управлению охраной труда, подготовку управленческих решений и контроль за их выполнением осуществляет служба охраны труда, непосредственно подчиняющаяся директору.

На службу охраны труда возлагаются следующие функции:

* анализ результатов аттестации рабочих мест по условиям труда, паспортизация санитарно-технического состояния условий и охраны труда, причин производственного травматизма и профессиональной заболеваемости в подчиненных организациях, разработка мер по их профилактике;
* организация разработки и контроль за реализацией отраслевых целевых программ по улучшению условий и охраны труда;
* организационное и методическое руководство работой структурных подразделений республиканского органа государственного управления, иной государственной организации, подчиненной Правительству Республики Беларусь, служб охраны труда подчиненных организаций по обеспечению здоровых и безопасных условий труда, анализ организации их работы;
* организация разработки и контроль за реализацией мероприятий по совершенствованию отраслевой СУОТ, организацией работы по обеспечению охраны труда в подчиненных организациях;
* участие в разработке проектов НПА, в том числе ТНПА, содержащих требования по охране труда, подготовка предложений по вопросам охраны труда к проектам законодательных и иных НПА;
* организация и руководство работой кабинета охраны труда;
* проведение проверок по осуществлению государственного контроля над соблюдением законодательства об охране труда в подчиненных организациях;
* подготовка государственной статистической отчетности по охране труда;
* пропаганда и распространение передового опыта в области охраны труда в подчиненных организациях, в том числе через средства массовой информации, организацию выставок, смотров-конкурсов, семинаров, совещаний и других мероприятий;
* участие в коллективно-договорном процессе при рассмотрении вопросов охраны труда, разработке планов мероприятий по охране труда, предупреждению производственного травматизма и профессиональных заболеваний, улучшения условий труда работающих;
* участие в установленном законодательством порядке в расследовании несчастных случаев на производстве;
* информационное обеспечение подчиненных организаций по вопросам охраны труда;
* консультирование, подготовка ответов на запросы подчиненных организаций, рассмотрение в установленном порядке писем, обращений по вопросам охраны труда;
* участие в пределах компетенции в международном сотрудничестве по вопросам охраны труда.

Порядок расследования несчастных случаев в ЧУО «Колледж бизнеса и права» определен Правилами расследования несчастных случаев на производстве и профессиональных заболеваний, утвержденными постановлением Совета Министров Республики Беларусь от 15 января 2004 г. № 30.

Основной целью расследования несчастных случаев является:

* установить причины, вызвавшие несчастный случай;
* наметить мероприятия, направленные на предотвращение подобных несчастных случаев.

На практике применяются различные методы изучения причин несчастного случая. Наиболее распространенные это: статистический, групповой, топографический и монографический.

Статистический метод позволяет судить о состоянии производственного травматизма за прошедший период с помощью двух показателей коэффициентов частоты и тяжести.

Групповой метод**.** Сущность этого метода состоит в том, что исследователь группирует акты о несчастных случаях за определенный период по профессиям, группам оборудования участкам, цехам, сопоставляет эти несчастные случаи, что позволяет намечать необходимые мероприятия.

Топографический метод**.** Особенность этого метода заключается в том, что несчастные случаи наносятся на планировку, которая дает представление о состоянии травматизма, по определенным рабочим местам.

Монографическое изучение состоит в углубленном исследовании выбранного объекта в совокупности со всей производственной обстановкой.

При изучении выявляются скрытые причины травм и заболеваний, изучаются все аварии в том числе аварии без человеческих жертв. Анализу при этом методе подвергаются также причины несчастных случаев на объектах, подобных изучаемому.

Глубокое и всестороннее изучение производственной обстановки дает возможность исключить в будущем не только случаи, аналогичные тем, которые имели место ранее, но и выявлять другие причины травматизма.

Главной задачей проводимого расследования любого несчастного случая на производстве является определение его истинных причин, необходимых для разработки практических мер по их предупреждению и направленных на ликвидацию, выявленных в ходе расследования нарушений требований охраны труда.

Только при условии, что причина несчастного случая определена правильно, можно быть уверенным в эффективности намеченных профилактических мероприятий.

Как было ранее отмечено, решающим этапом в расследовании несчастных случаев на производстве является выяснение обстоятельств, при которых они произошли. Многолетняя практика расследования несчастных случаев на производстве свидетельствует, что в ходе расследования члены комиссии обязаны лично:

* провести обследование места происшествия (рабочего места, производственного участка, цеха);
* опросить пострадавшего (по возможности) и очевидцев несчастного случая и зафиксировать результаты опросов;
* заслушать и зафиксировать объяснения должностных и иных лиц, показания которых могут быть необходимы;
* изучить действующие в организации нормативные и организационно-распорядительные документы, регламентирующие требования безопасности, обязанности и ответственность конкретных должностных лиц за обеспечение здоровых и безопасных условий труда;
* установить причины несчастного случая и лиц, допустивших нарушения законодательных и иных нормативных правовых актов о труде и охране труда;
* предложить руководству организации необходимые мероприятия по предупреждению подобных случаев.

В ЧУО «Колледж бизнеса и права» проводятся следующие виды инструктажей:

* вводный инструктаж (проводится инженером ОТ);
* первичный инструктаж на рабочем месте (инженер по охране труда или специалист организации, на которого возложены эти обязанности);
* повторный инструктаж на рабочем месте (непосредственный руководитель работ);
* внеплановый инструктаж (непосредственный руководитель работ (начальник производства, цеха, участка, мастер, инструктор и другие должностные лица);
* целевой инструктаж (проводиться непосредственным руководителем работ).

Основными видами контроля в организации являются:

* контроль за соблюдением законодательства об охране труда, осуществляемый руководителем работ (директором), в соответствии с его должностными обязанностями.
* периодический контроль за соблюдением законодательства об охране труда, осуществляемый представителями нанимателя (директора) с участием общественных инспекторов профсоюзов или уполномоченных лиц по охране труда и проводиться ежеквартально;
* общественный контроль за соблюдением законодательства об охране труда, осуществляемый профсоюзами в соответствии с Порядком осуществления профсоюзами общественного контроля за соблюдением законодательства Республики Беларусь о труде, утвержденным постановлением Совета Министров Республики Беларусь от 6 мая 2010 г.   
  № 240.
* контроль по охране труда, осуществляемый службой охраны труда организации в соответствии с Типовым положением о службе охраны труда организации, утвержденным постановлением Министерства труда и социальной защиты Республики Беларусь от 30 сентября 2013 г. № 98.

Периодический контроль осуществляет инспектор по охране труда. В зависимости от деятельности и структуры организации может проводиться:

* ежедневно – на участке;
* ежемесячно – в учреждении, на закрепленных участках за членами комиссии (комитета) по охране труда;
* ежеквартально – в учреждении в целом.

Периодический контроль осуществляется представителями нанимателя с участием общественных инспекторов профсоюзов по охране труда.

При отсутствии в организации профсоюза с согласия нанимателя для участия в проведении периодического контроля могут привлекаться уполномоченные лица по охране труда работников.

3.2 Разработка мер по нормализации и защите от инфракрасного излучения

Инфракрасное излучение - один из видов излучений, которые появились в природе, обусловленные в основном техногенным направлением развития нашей жизни. Источником инфракрасного излучения является солнце, около 50 % излучения которого лежит в инфракрасной области. Значительная доля (от 70 до 80 %) энергии излучения ламп накаливания с вольфрамовой нитью приходится на инфракрасное излучение. При фотографировании в темноте и в некоторых приборах ночного наблюдения лампы для подсветки снабжаются инфракрасным светофильтром, который пропускает только инфракрасное излучение. Мощным источником инфракрасного излучения является угольная электрическая дуга с температурой ~ 3900 К, излучение которой близко к излучению чёрного тела, а также различные газоразрядные лампы (импульсные и непрерывного горения). Для радиационного обогрева помещений применяют спирали из нихромовой проволоки, нагреваемые до температуры ~ 950 К. Для лучшей концентрации инфракрасного излучения такие нагреватели снабжаются рефлекторами. В научных исследованиях, например, при получении спектров инфракрасного поглощения в разных областях спектра применяют специальные источники инфракрасного излучения: ленточные вольфрамовые лампы, штифт Нернста, глобар, ртутные лампы высокого давления и другие [5].

Инфракрасное излучение (тепловое излучение, инфракрасные лучи) проникают в ткани организма глубже, чем другие виды световой энергии, что вызывает прогревание всей толщи кожи и отчасти подкожных тканей. Более глубокие структуры прямому прогреванию не подвергаются. Область терапевтического применения инфракрасного излучения довольно широка: негнойные хронические и подострые воспалительные местные процессы, в том числе внутренних органов, некоторые заболевания опорно-двигательного аппарата, центральной и периферической нервной системы, периферических сосудов, глаз, уха, кожи, остаточные явления после ожогов и отморожений.

Лечебный эффект инфракрасного облучения определяется механизмом его физиологического действия – он ускоряет обратное развитие воспалительных процессов, повышает тканевую регенерацию, местную сопротивляемость и противоинфекционную защиту. Нарушение правил проведения процедур может привести к опасному перегреву тканей и возникновению термических ожогов, а также к перегрузке кровообращения, опасной при сердечно-сосудистых заболеваниях.

Абсолютными противопоказаниями являются опухоли (доброкачественные или злокачественные) или подозрение на их наличие, активные формы туберкулеза, кровотечение, недостаточность кровообращения.

Инфракрасное излучение генерируется любым нагретым телом, температура которого определяет интенсивность и спектр излучаемой электромагнитной энергии.

Значительное изменение общей температуры тела (от 1,5 до 2 °С) происходит при облучении инфракрасными лучами большой интенсивности. Воздействуя на мозговую ткань, коротковолновое излучение вызывает «солнечный удар». Человек при этом ощущает головную боль, головокружение, учащение пульса и дыхания, потемнение в глазах, нарушение координации движений, возможна потеря сознания. При интенсивном облучении головы происходит отёк оболочек и тканей мозга, проявляются симптомы менингита и энцефалита.

При воздействии на глаза наибольшую опасность представляет коротковолновое излучение. Возможное последствие воздействия инфракрасного излучения на глаза – появление инфракрасной катаракты [4]. С учетом особенности биологического действия по длинам волн ИК-излучения делятся на области:

длинноволновой – от 0,76 до 15 Мкм;

средневолновой – от 16 до 100 Мкм;

коротковолновой – более 100 Мкм.

С повышением температуры тела интенсивность излучения (Е) увеличивается и определяется по формуле:

Где τ – постоянная Стефана – Больцмана, τ = 5,63032 · 10-8 Вт м-2 К-4,

T – абсолютная температура тела, К.

Допустимый уровень интенсивности интегрального потока инфракрасного излучения ТНП не должен превышать 100 Вт/м2.

Интенсивность излучения от экранов телевизоров, видеомониторов, осциллографов измерительных и других приборов, средств отображения информации с визуальным контролем не должна превышать 0,1 Вт/м2 в видимом от 400 до 760 Нм диапазоне, 0,05 Вт/м2 в ближнем инфракрасном (ИК) диапазоне от 760 до 1050 Нм, 4 Вт/м2 в дальнем свыше 1050 Нм ИК диапазоне.

Проблема безопасности является составной частью проблемы сохранности. В целом организация системы безопасности складывается из трёх составляющих: организация и физическая охрана, инженерно-техническая укрепленность и сигнализация.

Успешному обеспечению безопасности способствуют заблаговременные мероприятия по выявлению и идентификации возможных угроз (опознание и предвидение, оценка, уменьшение вредного влияния их на человека и среду его обитания).

Для комплексного обеспечения безопасности объекты оборудуют системами связи, диспетчеризации, оповещения, контроля и управления доступом; охранными, пожарными, телевизионными, инженерными устройствами и системами; охранной, пожарной сигнализацией ипротивопожарной автоматикой.

Все эти виды работ входят в состав мероприятий по обеспечению сохранности и защиты.

Комплексно мероприятия по обеспечению сохранности и защиты информации, объектов (зданий, помещений, оборудования) и людей обеспечиваются охраной объектов с использованием организационных, физических, программно-технических и социально-психологических мероприятий, а также инженерно-технические средства защиты.

Для охраны зданий используют системы освещения и видеонаблюдения, датчики-извещатели охранной сигнализации, реагирующие на движущиеся источники теплового излучения и управление доступом.

К инженерно-техническим средствам защиты относят:

* специальное укрепление зданий и помещений;
* хранилища;
* системы пассивной безопасности (двери и металлоконструкции, замки, защитные стекла, витрины и стенды, сейфы и металлические шкафы; преграждающие, ограждающие и запирающие устройства, ворота);
* средства индивидуальной защиты.

Эти же мероприятия способствуют защите программно-технических средств, информационных ресурсов, людей, включают средства и системы активной безопасности.

В качестве технических средств физической безопасности используются решетки на окнах, ограждения, металлические двери, турникеты, металодетекторы и другие. Программно-технические средства включают различные системы ограничения доступа на объект, сигнализации и видеонаблюдения [6].

Для комплексного обеспечения безопасности объекты оборудуются системами связи, диспетчеризации, оповещения, контроля и управления доступом; охранными, пожарными, телевизионными и инженерными устройствами и системами; охранной, пожарной сигнализацией и автоматикой.

В состав технических средств охраны и безопасности входят системы охранно-пожарной сигнализации, извещатели, контактные, магнитоконтактные и электроконтактные, фотоэлектрические и вибродатчики, ультразвуковые, радиоволновые и иные детекторы, исполнительные устройства, системы ограничения доступа, видеонаблюдения и другие устройства.

Защита работников и посетителей входит в состав общих организационных и технических мероприятий по защите организации от различных предвиденных и непредвиденных отрицательных воздействий. Ее содержание тесно переплетается с общими мероприятиями и отличается отдельными нюансами.

Управление доступом служит для контроля входа/выхода через автоматические проходные (турникеты, арочные металодетекторы) работников и посетителей организации. Контроль их перемещения осуществляется с помощью систем видеонаблюдения. В управление доступом входят устройства и (или) системы ограждения для ограничения входа на территорию (охрана периметров).

Используются также методы визуализации (предъявление вахтеру соответствующих документов) и автоматической идентификации входящих/выходящих работников и посетителей. Арочные металодетекторы способствуют выявлению несанкционированного вноса/выноса металлизированных предметов и маркированных документов.

Автоматизированные системы управления доступом позволяют работникам и посетителям, пользуясь персональными или разовыми электронными пропусками, проходить через проходную здания организации, заходить в разрешенные помещения и подразделения.

В организациях, с целью предоставления возможности отдельным индивидам проходить в соответствующие здания и помещения и пользования информационными ресурсами, применяют: удостоверения, пропуска, читательские билеты и другие, в том числе в виде контактных и бесконтактных пластиковых карт, магнитных и электронных карт памяти или ламинированных карточек, содержащих идентифицирующие пользователей штрих-коды, а также биометрические системы.

В качестве бесконтактных идентификаторов, взамен штрих-кодов, магнитных полос и других подобных устройств, можно использовать – RFID

Контроль доступа к информации обеспечивается последовательным использованием таких методов защиты информации, как идентификация и аутентификация.

Идентификация – присвоение объектам (пользователям и осуществляемым ими процессам) специальных имен и кодов с целью опознания и учёта факторов обращения. Это процедура сопоставления предъявленных характеристик с эталонными. Проверка принадлежности пользователю предъявленного им идентификационного признака осуществляется в процессе аутентификации.

Аутентичность – проверка подлинности объекта с помощью информации, хранящейся в «матрице доступа» для определения его истинности. Подлинность информации проверяется на основе её внутренней структуры. Аутентификация информации позволяет установить факт её получения законным путём (передана законным отправителем, не была искажена или заменена).

Защита «временем» предусматривает нахождение в контакте с излучением только по служебной необходимости с четкой регламентацией по времени и пространству совершаемых действий; автоматизацию работ; уменьшение времени настроечных работ и так далее. В зависимости от воздействующих уровней (инструментальный и расчетный методы оценки) время контакта с ними определяется в соответствии с действующими нормативными документами [6].

К организационно-планировочным мерам защиты следует отнести и проведение ряда лечебно-профилактических мероприятий. Это, прежде всего, обязательное медицинское освидетельствование при приеме на работу, последующие периодические медицинские обследования, что позволяет выявить ранние нарушения в состоянии здоровья персонала, отстранить от работы при выраженных изменениях состояния здоровья. В каждом конкретном случае оценка риска здоровью работающих должна базироваться на качественной и количественной характеристике факторов. Существенным с позиции влияния на организм является характер профессиональной деятельности и стаж работы. Важную роль играют индивидуальные особенности организма, его функциональное состояние.

3.3 Пожарная безопасность

По Техническому кодексу установившейся практики (ТКП) 474 – 2013 категорию помещений можно определить, как В1-В3 (пожароопасные), потому что в помещении есть горючие вещества и материалы, способные при взаимодействии с кислородом или воздуха гореть.

Пожароопасные зоны по правилу устройства электроустановок (ПУЭ) на предприятии относятся к категории П-IIa (зоны, расположенные в помещениях, в которых обращаются твёрдые горючие вещества).

Согласно строительным нормам Республики Беларусь (СНБ) 2.02.01-98 «Пожарно-техническая классификация зданий, строительных конструкций и материалов» здание предприятия относится к V степени огнестойкости. Это здания с несущими и ограждающими конструкциями из искусственных каменных материалов, железобетона. Для перекрытий использование деревянные конструкции, защищенные штукатуркой, а также плитными материалами. К элементам покрытий не предъявляются требования по пределам огнестойкости и пределам распространения огня, при этом элементы чердачного покрытия из древесины подвергаются огнезащитной обработке.

По пределу огнестойкости и классу пожарной опасности строительных конструкций здание предприятия характеризуется:

* несущие элементы здания (R 45-K1): предел огнестойкости 45 минут - по потере несущей способности и мало пожароопасные;
* самонесущие стены (RE 30-K1): предел огнестойкости 30 минут - по потере несущей способности и потере целостности независимо от того, какое из двух предельных состояний наступит ранее, и являются мало пожароопасными;
* наружные ненесущие стены (E 15-K2): предел огнестойкости 15 минут - по потере целостности и умеренно пожароопасные;
* перекрытия междуэтажные (REI 45-K1): предел огнестойкости 45 минут - по потере несущей способности, целостности и теплоизолирующей способности независимо от того, какое из трех предельных состояний наступит ранее и являются мало пожароопасными;

Для элементов бесчердачных покрытий:

* настилы в том числе с утеплителем (RE 15-K1): предел огнестойкости 15 минут - по потере несущей способности и потере целостности независимо от того, какое из двух предельных состояний наступит ранее и являются мало пожароопасными;
* фермы, балки, прогоны (R 15-K1): предел огнестойкости 15 минут - по потере несущей способности и мало пожароопасные;

Для лестничных клеток:

* внутренние стены (REI 60-KO): предел огнестойкости 60 минут - по потере несущей способности, целостности и теплоизолирующей способности независимо от того, какое из трех предельных состояний наступит ранее и являются не пожароопасными;
* марши и площадки лестниц (R 45-KO): предел огнестойкости 45 минут - по потере несущей способности и не пожароопасные.

Для обнаружения начальной стадии загорания и оповещения в службу пожарной охраны, в рассматриваемом здании предприятия используются системы автоматической пожарной сигнализации (АПС). Системы АПС состоят из пожарных извещателей, линий связи и приемных пультов (станций). В помещениях с ПЭВМ установлены дымовые пожарные извещатели, т.к. в начале пожара при горении различных пластмассовых, изоляционных материалов и бумажных изделий выделяется значительное количество дыма и мало теплоты.

Для тушения пожаров на начальных стадиях широко применяются огнетушители. В каждом кабинете находится по одному порошковому огнетушителю, который является практически универсальным (тушит как твердые, так и жидкие вещества).

Расстояние от двери до огнетушителя выбрано так, чтобы не мешать ее полному открыванию. Расстояние от возможного очага пожара до огнетушителя не превышает 30 м. Располагается на высоте не более 1,5 м от пола до верха огнетушителя и защищен от воздействия прямых солнечных лучей, тепловых потоков, механических воздействий.

В здании пожарные краны установлены в коридорах, на лестничных клетках и у входов.

При проектировании здания были предусмотрены пути эвакуации. Путями эвакуации являются: проходы, коридоры, площадки, лестницы, ведущие к эвакуационному выходу, обеспечивающие безопасное движение людей в течение необходимого времени эвакуации.

При эвакуации нужно действовать согласно инструкции по эвакуации в экстренных ситуациях.

Для предотвращения несчастных случаев при эвакуации на предприятии проводятся практические занятия по эвакуации.

Для проведения профилактических мероприятий по предупреждению чрезвычайных ситуаций и тушению пожаров, организуется обще объектовая ДПД из числа работников и администрации в количестве 15 человек.

Организация пожарной дружины и руководство ее деятельностью возлагается на директора ЧУО «Колледж бизнеса и права». Командир ДПД, старшие расчетов, составы расчетов назначаются директором Колледжа.

Основные задачи ДПД:

* контроль над соблюдением мер безопасности при угрозе или возникновении ЧС различного характера, противопожарного режима;
* проведение разъяснительной работы среди рабочих, служащих, инженерно-технических работников по соблюдению мер безопасности и противопожарного режима на рабочем месте;
* надзор за исправностью средств пожаротушения и их укомплектованностью;
* вызов пожарной службы в случае возникновения пожара, принятие мер по его тушению имеющимися средствами пожаротушения;
* обеспечение охраны общественного порядка при возникновении чрезвычайных ситуаций (ЧС) различного характера.

3.4 Охрана окружающей среды

Воздух после прохождения через аудитории приобретает некоторые негативные факторы, которые отрицательно влияют на окружающую среду. Чтобы уменьшить степень негативного влияния на окружающую среду при проектировании систем вентиляции предусматриваются дополнительные системы рециркуляции и фильтрации воздуха.

Перечень мероприятий по обслуживанию систем вентиляции:

* + не допустить утечку хладагента при монтаже или последующем обслуживании охлаждающих устройств, использовать специальные контейнеры, утилизировать или регенерировать его;
  + использовать современные холодильные масла и смазки;
  + при смене холодильных масел и смазки не допустить загрязнение окружающей среды, использовать специальные контейнеры, утилизировать или регенерировать их;
  + проводить своевременный осмотр и эксплуатацию механических частей системы вентиляции.

Предприятие сбрасывает очищенные или неочищенные сточные воды в окружающую среду. Это приводит к загрязнению гидрографической сети и территории района ее размещения. Основными источниками загрязнения поверхностных вод являются: неочищенные или недостаточно очищенные производственные и бытовые сточные воды; поверхностный сток с территории колледжа; фильтрационные утечки вредных веществ из емкостей, трубопроводов и других сооружений; аварийные сбросы сточных вод.

Для рационального использования и охраны поверхностных вод предприятие ЧУО «Колледж бизнеса и права» должно обеспечить: соблюдение установленных лимитов забора воды и сброса сточных вод; уменьшение потерь воды; уменьшение загрязнения воды; внедрение систем управления водными ресурсами, сокращающих или благоприятствующих уменьшению избыточного потребления воды; экономное и рациональное использование водных ресурсов.

Очистка сточных вод – это комплекс мероприятий по удалению загрязнений, содержащихся в бытовых и промышленных сточных водах перед выпуском их в водоёмы. Процесс отчистки делится на четыре этапа: механический; биологический; физико-химический; дезинфекция сточных вод.

Предельно допустимый выброс вредных веществ в атмосферу (ПДВ) устанавливают для каждого источника загрязнения атмосферы при условии, что выбросы вредных веществ отданного источника и от совокупности источников города или другого населенного пункта и рассеивания вредных веществ в атмосфере, не создадут приземную концентрацию, превышающую их предельно допустимые концентрации (ПДК) для населения, растительного и животного мира.

Предельно допустимые концентрации (ПДК) - максимальные концентрации, при которых вещества не оказывают прямого или опосредованного влияния на состояние здоровья населения (при воздействии на организм в течение всей жизни) и не ухудшают гигиенические условия водопользования.

Установление значения ПДВ, как количества вредных веществ, которое не разрешается превышать при выбросе в атмосферу в единицу времени, производят на основе методов расчета ПДВ в атмосферу, утвержденных.

Для предотвращения и максимального снижения организованных и не организованных выбросов вредных веществ должны быть использованы наиболее современная технология, методы очистки и другие технические средства в соответствии с требованиями санитарных норм проектирования промышленных предприятий.

ПДВ (ВСВ) устанавливают для каждого источника загрязнения атмосферы. Для неорганизованных выбросов и совокупности мелких одиночных источников (вентиляционные выбросы из одного производственного помещения, от одной, расположенной в помещении или на открытом воздухе, установки и т.п. устанавливают суммарный ПДВ (ВСВ). В результате суммирования ПДВ (ВСВ) отдельных источников загрязнения атмосферы устанавливают значения ПДВ (ВСВ) для предприятий или объектов и их комплексов в целом. ПДВ (ВСВ)пересматривают не реже одного раза в пять лет.

Транспортировку отходов следует производить в специально оборудованном транспорте. Транспорт для перевозки полужидких отходов должен быть снабжен шланговым приспособлением для слива. При перевозке твердых отходов необходимо самостоятельное устройство или тара с захватными приспособлениями для разгрузки автокранами полигона.

Обезвреживание и захоронение токсичных промышленных отходов осуществляется на специальных инженерных сооружениях – полигонах захоронения токсичных промышленных отходов. Обезвреживание в условиях полигона осуществляется тремя доступными методами: сжиганием, нейтрализацией и захоронением. На полигоны по обезвреживанию и захоронению токсичных промышленных отходов принимаются только отходы первых-третьих классов опасности. Твердые промышленные отходы и классы опасности по согласованию с органами санитарного надзора и коммунальными службами могут вывозиться на полигоны для складирования твердых бытовых отходов.

Захоронение не утилизируемых отходов должно происходить на специально организованных полигонах. Полигоны должны располагать резервной территорией с расчетным сроком их эксплуатации на от 20 до 25 лет. Для захоронения не утилизируемых токсичных производственных отходов на полигонах каждое министерство, ведомство, предприятие проводят паспортизацию не утилизируемых отходов, определяют их количество по четырем классам опасности, согласовывают список с администрацией полигона и с органами санитарно-эпидемиологической службы и представляют проектным организациям или полигону, принимающему отходы на утилизацию.

4 План мероприятий по разработке и внедрению программы

Разработка программного приложения происходит в несколько этапов. Обычно этих этапов шесть. На каждом этапе выполняются определённые работы. Ниже описаны все этапы разработки по порядку.

Первый этап по разработке приложения – постановка и утверждение технического задания. Целью разработки технического задания является обоснование необходимости разработки данного проекта. На этом этапе выполняются работы по обслуживанию информационных потоков в подразделение, в котором осуществляется автоматизация управления, постановка задачи, сбор исходных материалов и определение структуры входных и выходных данных.

Второй этап разработки – это разработка и утверждение эскизного проекта. На этом этапе выполняются работы по предварительной разработке структуры выходных и выходных данных, уточнение методов решения задачи, разработка общего описания методов решения задачи, а также согласование и утверждения эскизного проекта.

Третьим этапом является разработка и утверждение технического проекта. На этом этапе происходит уточнение структуры входных и выходных данных, разработка структуры в среде разработки, определение формы представления входных и выходных данных, разработка плана по разработке и внедрению программы.

Четвертым этапом является разработка рабочего проекта. В рабочий проект входит программные модули и программная документация, то есть рабочий проект непосредственно сама разрабатываемая программа о всей вспомогательной и программной документацией. На этой стадии ведутся работы по программированию и отладке модулей.

Пятым этапом является тестирование рабочего проекта. Тестирование является обязательным этапом. Рабочий проект тестируется на наличие багов и мелких ошибок, что бы в последующем действии программное средство отвечало на данные ответы пользователя.

Шестым этапом является внедрение программы на предприятие, то есть подготовка к передаче программы на предприятие. Здесь ведутся работы по подготовке и передаче программы для сопровождения, оформление и утверждение акта о передаче программы.

5 Техно-рабочий проект

5.1 Исследование предметной области

Рассмотрим предметную область задачи.

Договорные отношения давно и прочно закрепились в нашей жизни, договора встречаются везде и повсеместно, практически ничто уже не обходится без заключения какого-либо договора. Предоставление услуг, выполнение работ, поставка материальных ценностей, аренда имущества, купля-продажа и многое другое - все это договоры.

Договоры, как правило, составляются юристами и учитывают все нюансы той или иной деятельности. Кураторы же имеют дело с типовыми формами договоров и производят с ними однотипные, повторяющиеся действия, которые можно и нужно автоматизировать.

Итак, менеджеры или кураторы договоров, используя текстовый редактор, (например, Microsoft Word) вносят изменения в типовые формы, шаблоны, корректируя определенные поля, такие как номер и дата договора, наименование контрагента, стоимость договора, паспортные данные, банковские реквизиты. При этом пересчитываются различные суммы, такие, например, как НДС, кстати суммы в договорах зачастую дублируются прописью.

Для составления преамбулы договора такие параметры как имена, фамилии и должности склоняются в различные падежи. Все это требует повышенного внимания и приводит к зрительным нагрузкам. Чем объемнее договор, чем больше в нем различных форм и приложений, тем сильнее желание автоматизировать его заполнение.

Цели, достигаемые разработкой приложения для заполнения шаблонов документов следующие:

* + быстрый выбор компаний участников договора;
  + внесение выбранных данных в документ;
  + управление списком товаров;
  + управление списком организаций;
  + уменьшение нагрузки на пользователя.

5.2 Структура входных и выходных данных

Входными данными являются выбранные компании, которые принимают участие в договоре, список продукции для поставки,.

Выходными данными являются частично заполненные документы.

5.3 Проектирование модели

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченном и получила дальнейшее развитие в работах Баркера. Диаграмма «сущность-связь» представлена на рисунке 1.1

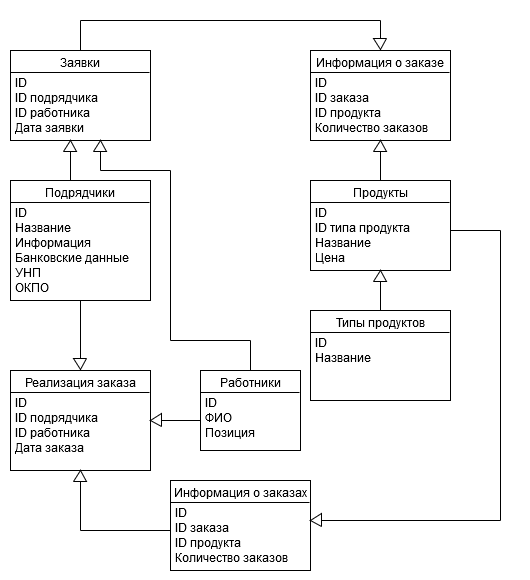


Рисунок 5.1

Исходя из исследования предметной области, можно выделить следующие сущности разработки: «Заявки», «Подрядчики», «Реализация заказа», «Работники», «Информация о заказах», «Продукты», «Типы продуктов», «Информация о заказе».

Для сущности «Заявки» можно выделить следующие атрибуты:

* + дата заявки.

Для сущности «Подрядчики» атрибутами будут являться:

* + название;
  + информация;
  + банковские данные;
  + УНП;
  + ОКПО.

Для сущности «Реализация заказа» атрибутами будут являться:

* + дата заказа.

Для сущности «Работники» атрибутами будут являться:

* + ФИО;
  + должность.

Для сущности «Информация о заказах» атрибутами будут являться:

* + количество заказов.

Для сущности «Продукты» атрибутами будут являться:

* + название;
  + цена.

Для сущности «Типы продуктов» атрибутами будут являться:

* + название.

Для сущности «Информация о заказе» атрибутами будут являться:

* + дата заказа.

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью, так называемых, вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

Актёр представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некоторая отдельная роль относительно конкретного варианта использования.

Данное программное средство имеет следующие основные (Include) функции:

* осуществить ведение базы данных;
* осуществить работы с базой данных;
* предусмотреть сохранение заполненных документов.

Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма классов для проектируемой системы представлена в графической части.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы.

Диаграмма последовательности для проектируемой системы представлена в графической части.

Рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Диаграмма компонентов описывает объекты реального мира - компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Вид диаграммы развертывания для данной проектируемой системы представлен в графической части.

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы - атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация – это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Структура базы данных разрабатываемого программного средства включает восемь таблиц.

Структура данных таблиц, и их краткое описание приводится в таблицах 5.1-5.8.

Таблица «Сотрудники» хранит информацию о работниках. Структура приведена в таблице 5.1.

Таблица 5.1 – Структура таблицы «Сотрудники»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор сотрудника |
| ФИО | varchar | 50 | ФИО работника |
| Должность | varchar | 50 | Должность работника |

Таблица «Подрядчики» хранит информацию о компаниях. Структура приведена в таблице 5.2.

Таблица 5.2 – Структура таблицы «Подрядчики»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор сборки |
| Название | varchar | 50 | Название организации |
| Адрес | varchar | 50 | Адрес организации |
| Банковские\_данные | varchar | 50 | Банковские данные организации |
| УНП | int | 4 | УНП номер организации |
| ОКПО | varchar | 12 | Общегосударственный Классификатор Предприятий и Организаций |

Таблица «Реализация заказа» хранит информацию о текущих заказах. Структура приведена в таблице 5.3.

Таблица 5.3 – Структура таблицы «Реализация заказа»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор заказа |
| ID\_подрядчика | int | 4 | Идентификатор подрядчика |
| ID\_работника | int | 4 | Идентификатор работника |
| Дата\_заказа | date | 8 | Дата заказа |

Таблица «Информация о заказах» хранит информацию о заказе. Структура приведена в таблице 5.4.

Таблица 5.4- Структура таблицы «Информация о заказах»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор заказа |
| ID\_заказа | int | 4 | Идентификатор заказов |
| ID\_продукта | int | 4 | Идентификатор продуктов |
| Количество\_заказов | int | 4 | Количество заказов |

Таблица «Заявки» хранит поступившие заявки. Структура приведена в таблице 5.5.

Таблица 5.5- Структура таблицы «Заявки»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор заявки |
| ID\_подрядчика | int | 4 | Идентификатор подрядчика |
| ID\_работника | int | 4 | Идентификатор работника |
| Дата\_заявки | date | 8 | Дата подачи заявки |

Таблица «Информация о заказе» хранит информацию заказах. Структура приведена в таблице 5.6.

Таблица 5.6 – Структура таблицы «Информация о заказе»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор заказа |
| ID\_заказа | int | 4 | Идентификатор заказов |
| ID\_продукта | int | 4 | Идентификатор продуктов |
| Количество\_заказов | int | 4 | Количество заказов |

Таблица «Продукты» хранит информацию о продуктах. Структура приведена в таблице 5.7.

Таблица 5.7- Структура таблицы «Продукты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор продукта |
| ID\_типа\_продукта | int | 4 | Идентификатор типа продукта |
| Название | varchar | 50 | Название товара |
| Цена | float | 8 | Цена за единицу продукции |

Таблица «Тип продуктов» хранит типы продукции. Структура приведена в таблице 5.8.

Таблица 5.8- Структура таблицы «Тип продуктов»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер, байт | Описание поля |
| ID | int | 4 | Идентификатор типа продукта |
| Название | varchar | 50 | Идентификатор заказов |

На рисунке 5.2 представлена логическая схема данных.

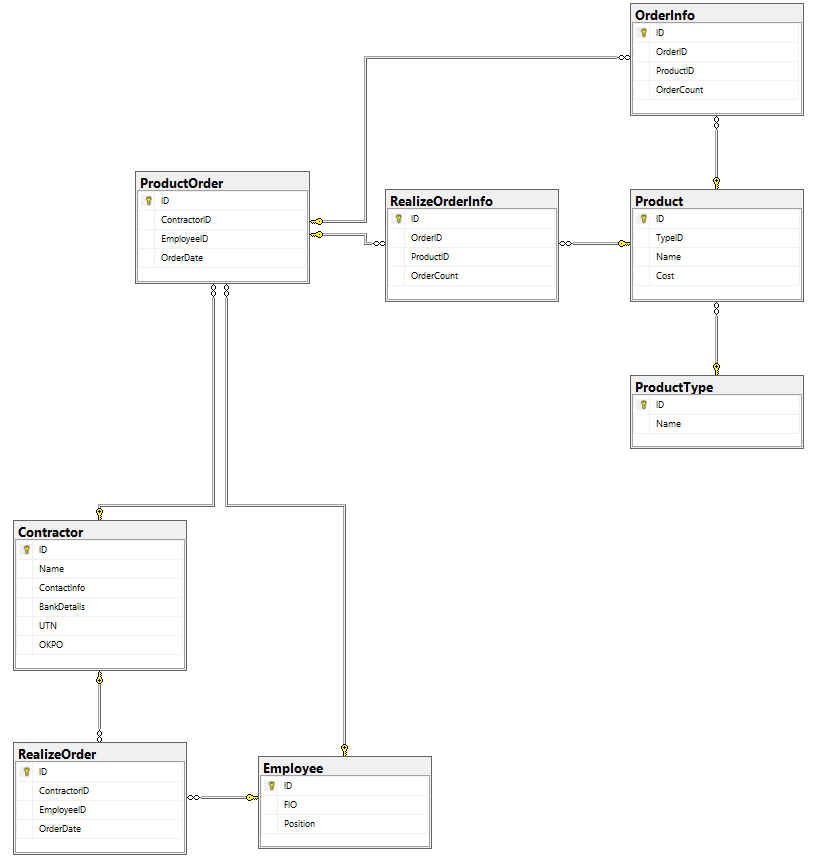


Рисунок 5.2

5.4 Концептуальный прототип

Приложение должно иметь интуитивно понятный интерфейс, который будет удобен пользователю.

На главной странице будут находиться такие компоненты как: вкладки с выбором справочника, вкладки с выбором каталогов, кнопки для добавления и удаления записей справочников. Макет главной страницы представлен на рисунке 5.3.

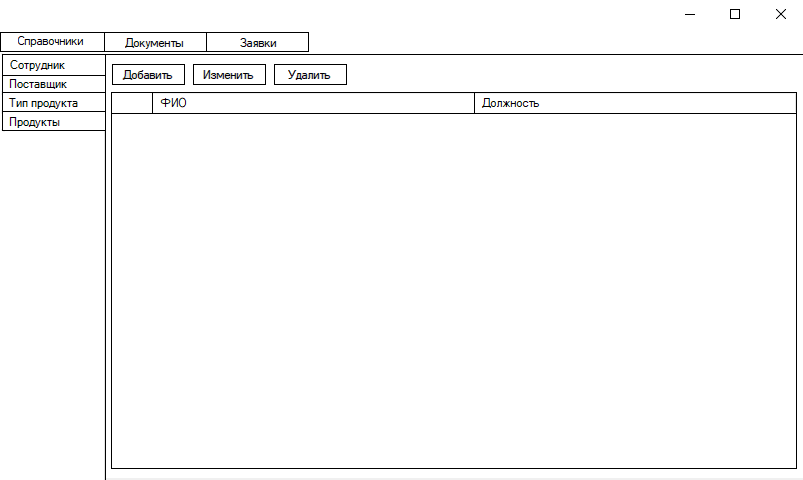


Рисунок 5.3 – Макет главной формы

На форме добавления товара будут находиться такие компоненты как: кнопка подтверждения, поля для ввода информации. Форма с добавлением типов товаров является аналогичной данной. Макет формы с добавлением товара на рисунке 5.4

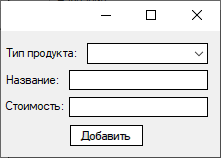


Рисунок 5.4 – Макет формы для добавления товара

На форме добавления сотрудников будут находиться такие компоненты как: кнопка подтверждения, поля для ввода информации. Макет формы добавления сотруднков представлен на рисунке 5.5

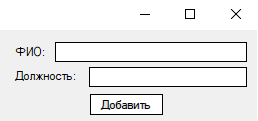


Рисунок 5.5 – Макет формы добавления сотруднков

На форме добавления заявок будут находиться такие компоненты как: кнопка подтверждения, поля для ввода информации. Формы с транспортными и товарно транспортными накладными являются идентичными данной. Макет формы добавления заявок представлен на рисунке 5.6

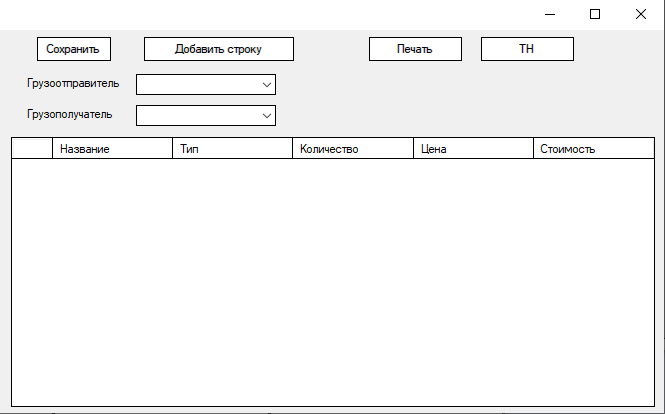


Рисунок 5.6 – Макет формы добавления заявок

5.5 Реализация функций

Для организации в приложении графического пользовательского интерфейса используются элементы управления в виде кнопок, таблиц и форм.

Пользователю предоставляется возможность просматривать, вводить/изменять и выводить данные. При нажатии на активный элемент управления, событие выполняет соответствующее действие.

Для выполнения вышеперечисленных функций необходимо запустить приложение. Откроется главное окно приложения со всеми функциями.

На основании диаграммы вариантов использования были реализованы следующие функции:

* ведение базы данных;
* добавление записей;
* удаление записей;
* сохранение отредактированных справочников;
* экспорт информации в шаблоны документов.

Для добавления записи пользователю требуется заполнить необходимые поля и нажать на кнопку «Сохранить». Код функции представлен ниже:

private void employeeAddButton\_Click(object sender, EventArgs e) {

EmployeeForm form = new EmployeeForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Employee\_Enter);

}

Для редактирования записей пользователю требуется выбрать необходимое поле и начать на кнопку «Редактировать» и в открывшейся форме изменить необходимые данные. Код функции представлен ниже:

private void employeeChangeButton\_Click(object sender, EventArgs e) {

EmployeeForm form = new EmployeeForm(Convert.ToInt32(employeeDataGridView.Rows[employeeDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Employee\_Enter);

}

Для удаления записей пользователю требуется выбрать запись и нажать на кнопку «Удалить». Код функции представлен ниже:

private void employeeDeleteButton\_Click(object sender, EventArgs e) {

using (SharagaContext db = new SharagaContext()) {

var List = db.Employee.Find(Convert.ToInt32(employeeDataGridView.Rows[employeeDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

db.Employee.Remove(List);

db.SaveChanges();

}

Employee\_Enter(sender, e);

}

5.6 Функциональное тестирование

В процессе написания программного средства необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

Таблица 5.9 – Test-case для добавления нового сотрудника

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тип теста | Описание | Шаги воспроизведения | Результат |
| 001 | Позитивный | Добавление сотрудника, ввод корректных данных | 1. Запустить приложение;  2. На панели навигации выбрать «Справочники» и нажать на кнопку «Сотрудник».  3. На форме «Сотрудник» нажать кнопку: «Добавить».  4. Заполнить поля:  ФИО – Карсюк Никита Владимирович,  Должность – Уборщик;  5. Нажать кнопку: «Добавить». | Ожидаемый результат: добавление сотрудника в БД. |
| Фактический результат:  результат соответствует ожидаемому. Все условия соблюдены. Результат правильного ввода данных в форме регистрации, представлен на рисунке 5.9. |

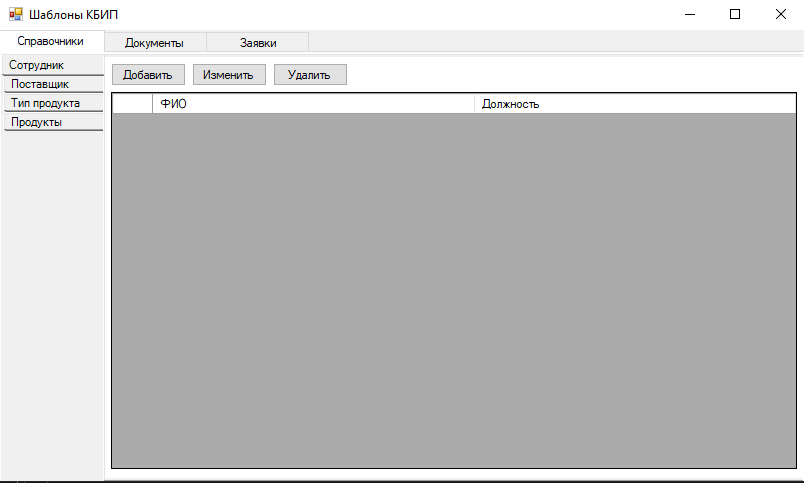


Рисунок 5.7 – Главная форма

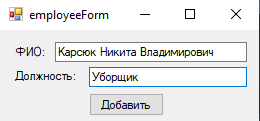


Рисунок 5.8 – Добавится новый сотрудник

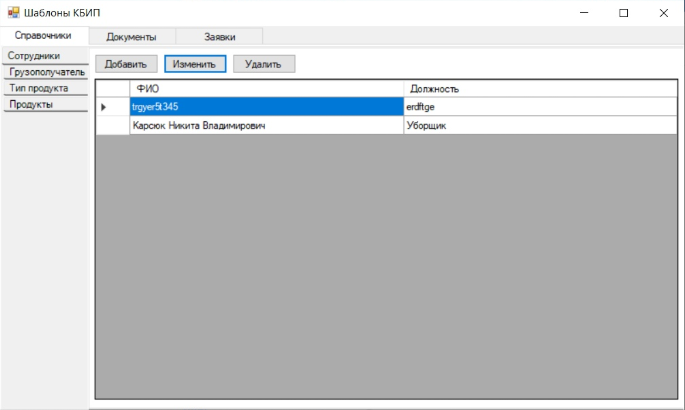


Рисунок 5.9 – Результат правильного ввода данных в форме сотрудника

Таблица 5.10 – Test-case добавления нового типа продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тип теста | Описание | Шаги воспроизведения | Результат |
| 002 | Позитивный | Добавление типа продукта | 1. Запустить приложение;  2. Перейти на вкладку «Тип продукта»  3. Заполнить поле:  Тип продукта: Молочные продукты  4. Нажать кнопку: «Добавить». | Ожидаемый результат: добавление нового типа продукта в справочник |
| Фактический результат:  результат соответствует ожидаемому. Все условия соблюдены. Результат добавления представлен на рисунке 5.11. |

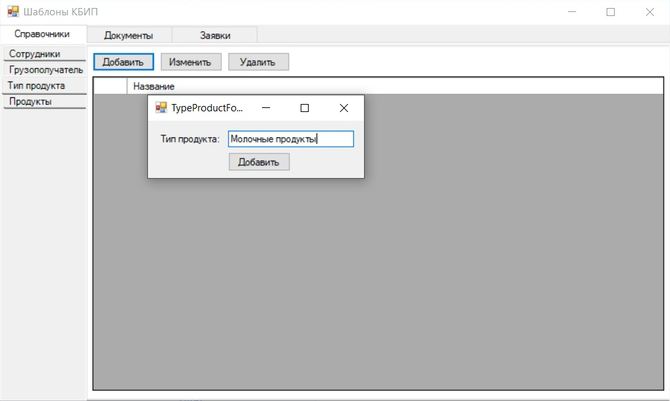


Рисунок 5.10 – Форма для добавления типов продукции

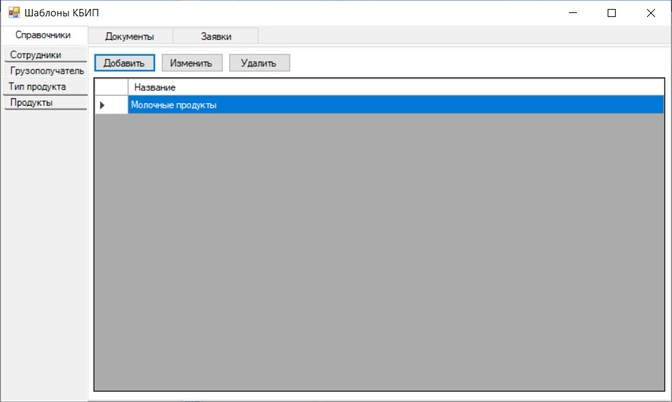


Рисунок 5.11 – Результат добавления

Таблица 5.11 – Test-case добавления нового продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тип теста | Описание | Шаги воспроизведения | Результат |
| 003 | Позитивный | Добавление продукта | 1. Запустить приложение;  2. Перейти на вкладку «Продукты»  3. Заполнить поля:  Тип продукта: Молочные продукты.  Название: Молоко.  Стоимость: 100  4. Нажать кнопку: «Добавить». | Ожидаемый результат: добавление нового продукта в справочник |
| Фактический результат:  результат соответствует ожидаемому. Все условия соблюдены. Результат добавления представлен на рисунке 5.13. |

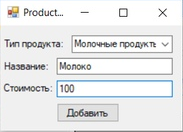


Рисунок 5.12 – Форма для добавления продукции

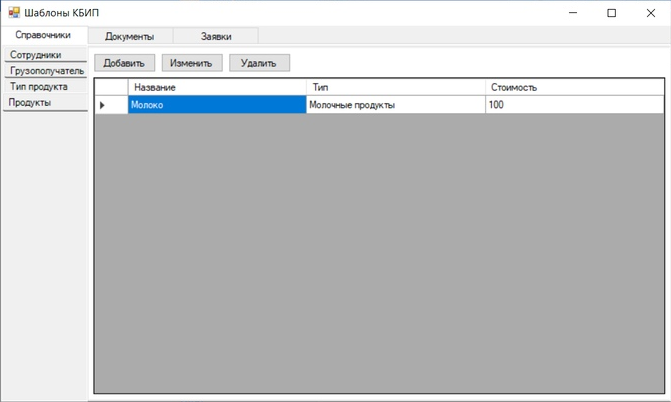


Рисунок 5.13 – Результат добавления нового продукта

6 Опытная эксплуатация

6.1 Ошибки, выявленные в процессе опытной эксплуатации

В процессе создания программного обеспечения были выявлены многочисленные ошибки, от мелких до более серьёзных. Самыми серьёзными являлись ошибки и недочёты при проектировании базы данных. Одними из причин ошибок являются непродуманность, невнимательность и другие факторы, влияющие на состояние человека.

Самой серьёзной ошибкой была непродуманность при создании базы данных, в следствии чего приходилось добавлять, изменять или удалять некоторые таблицы, либо атрибуты в них. Также можно отметить неправильную установку связи между базой данных и самим программным средством, из-за чего данные в базу данных могли заполняться некорректно, либо вовсе не добавляться в неё с выведением ошибки на экран.

Неправильное видение программного средства также являлось очень серьёзным недочётом. Каждый раз видоизменяя дизайн проекта приходилось переписывать функции.

Выводы

В рамках производственной технологической практики было разработано программное средство «Logistic.exe», предназначенное для автоматизации рабочего места сотрудника логистической компании.

В процессе разработки данного программного средства были применены и закреплены знания по уже изученному материалу, были отработаны навыки владениями методами надёжного программирования и эффективности разработки программного обеспечения в Microsoft Visual Studio с использованием языка программирования C#, разработана база данных средствами системы управления базами данных Microsoft SQL Server 2017.

Программное средство упрощает работу сотрудника логистической компании с помощью удобного интерфейса. Разработка имеет интуитивно понятный графический интерфейс, позволяющий с минимальным знанием компьютера пользоваться программным средством.

Список информационных источников

1. Багласова, Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» / Т.Г. Багласова. – Минск: КБП, 2017. – 30 c.

2. Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г.Багласова, К.О.Якимович. – Минск: КБП, 2013.

3. Лесневский, А. С. Объектно-ориентированное программирование: Бином. Лаб. знаний / Лесневский А. С. — М.: Бином. Лаб. знаний, 2010. — 232с

4. Общие требования к тестовым документам: ГОСТ 2.105-95. – Введ. 01.01.1996. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 1995. – 84 с.

5. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.301-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.

6. Текст программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.401-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 16 с.

7. Windows 10 – Бесплатные программы для Windows на русском языке, 2020 – [Электронный ресурс] / Режим доступа: http://softcatalog.info/ru/programmy/windows-10/

8. Microsoft Office 2019 – Microsoft, 2020 – [Электронный ресурс] / Режим доступа: http://microspft-office.biz/microsoft-office-2010/microsoft-office-2010.html

9. UMLet 14.3, Бесплатные аналоги, 2020 – [Электронный ресурс] / Режим доступа: http://freeanalogs.ru/UMLet

10. Visual Studio Community 2019, Заметки IT-специалиста, 2020 – [Электронный ресурс] / Режим доступа: http://info-comp.ru/programmirovanie/568-install-visual-studio-community-2017.html

11. Microsoft Server Express 2017, Microsoft, 2020 – [Электронный ресурс] / Режим доступа: http://www.docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp

12. Язык программирования C#, Microsoft, 2020 – [Электронный ресурс] / Режим доступа: http://www.docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-cshar

Приложение А

(обязательное)

Текст программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TemplateProject

{

public partial class ContractorForm : Form

{

private int ID { get; }

public ContractorForm(int ID = -1)

{

InitializeComponent();

this.ID = ID;

if (ID != -1)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Contractor.Find(ID);

ContractorNameTextBox.Text = List.Name;

ContractorAddressTextBox.Text = List.ContactInfo;

ContractorBankDetailsTextBox.Text = List.BankDetails;

ContractorUTNTextBox.Text = List.UTN;

ContractorOKPOTextBox.Text = List.OKPO;

}

}

}

private void addButton\_Click(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

if (ID == -1)

{

db.Contractor.Add(new Contractor()

{

Name = ContractorNameTextBox.Text,

ContactInfo = ContractorAddressTextBox.Text,

BankDetails = ContractorBankDetailsTextBox.Text,

UTN = ContractorUTNTextBox.Text,

OKPO = ContractorOKPOTextBox.Text

});

}

else

{

var List = db.Contractor.Find(ID);

List.Name = ContractorNameTextBox.Text;

List.ContactInfo = ContractorAddressTextBox.Text;

List.BankDetails = ContractorBankDetailsTextBox.Text;

List.UTN = ContractorUTNTextBox.Text;

List.OKPO = ContractorOKPOTextBox.Text;

}

db.SaveChanges();

this.Close();

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TemplateProject

{

public partial class EmployeeForm : Form

{

private int ID { get; }

public EmployeeForm(int ID = -1)

{

InitializeComponent();

this.ID = ID;

if (ID != -1)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Employee.Find(ID);

EmployeeFIOTextBox.Text = List.FIO;

EmployeePositionTextBox.Text = List.Position;

}

}

}

private void addButton\_Click(object sender, EventArgs e)

{

if (ID == -1)

{

using (SharagaContext db = new SharagaContext())

{

db.Employee.Add(new Employee()

{

FIO = EmployeeFIOTextBox.Text,

Position = EmployeePositionTextBox.Text,

});

db.SaveChanges();

this.Close();

}

}

else

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Employee.Find(ID);

List.FIO = EmployeeFIOTextBox.Text;

List.Position = EmployeePositionTextBox.Text;

db.SaveChanges();

this.Close();

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TemplateProject

{

public partial class ProductForm : Form

{

private int ID { get; }

public ProductForm(int ID = -1)

{

InitializeComponent();

this.ID = ID;

}

private void ProductForm\_Load(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

foreach (var type in db.ProductType)

{

TypeComboBox.Items.Add(type.Name);

}

}

if (ID != -1)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Product.Find(ID);

TypeComboBox.Text = List.ProductType.Name;

ProductNameTextBox.Text = List.Name;

CostTextBox.Text = List.Cost.ToString();

}

}

}

private void addButton\_Click(object sender, EventArgs e)

{

if (ID == -1)

{

using (SharagaContext db = new SharagaContext())

{

db.Product.Add(new Product()

{

TypeID = db.ProductType.First(x => x.Name == TypeComboBox.Text).ID,

Name = ProductNameTextBox.Text,

Cost = Convert.ToDouble(CostTextBox.Text)

});

db.SaveChanges();

this.Close();

}

}

else

{

}

}

}

}

using System;

using System.Drawing;

using System.Windows.Forms;

namespace TemplateProject

{

public partial class TemplateForm : Form

{

public TemplateForm()

{

InitializeComponent();

}

private void tabControl1\_DrawItem(object sender, DrawItemEventArgs e)

{

Graphics g;

string sText;

int iX;

float iY;

SizeF sizeText;

TabControl ctlTab;

ctlTab = (TabControl)sender;

g = e.Graphics;

sText = ctlTab.TabPages[e.Index].Text;

sizeText = g.MeasureString(sText, ctlTab.Font);

iX = e.Bounds.Left + 6;

iY = e.Bounds.Top + (e.Bounds.Height - sizeText.Height) / 2;

g.DrawString(sText, ctlTab.Font, Brushes.Black, iX, iY);

}

private void tabControl2\_DrawItem(object sender, DrawItemEventArgs e)

{

Graphics g;

string sText;

int iX;

float iY;

SizeF sizeText;

TabControl ctlTab;

ctlTab = (TabControl)sender;

g = e.Graphics;

sText = ctlTab.TabPages[e.Index].Text;

sizeText = g.MeasureString(sText, ctlTab.Font);

iX = e.Bounds.Left + 6;

iY = e.Bounds.Top + (e.Bounds.Height - sizeText.Height) / 2;

g.DrawString(sText, ctlTab.Font, Brushes.Black, iX, iY);

}

private void Employee\_Enter(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

employeeDataGridView.Rows.Clear();

int i = 0;

foreach (var employee in db.Employee)

{

employeeDataGridView.Rows.Add();

employeeDataGridView[0, i].Value = employee.ID;

employeeDataGridView[1, i].Value = employee.FIO;

employeeDataGridView[2, i].Value = employee.Position;

i++;

}

}

}

private void Contractor\_Enter(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

сontractorDataGridView.Rows.Clear();

int i = 0;

foreach (var contractor in db.Contractor)

{

сontractorDataGridView.Rows.Add();

сontractorDataGridView[0, i].Value = contractor.ID;

сontractorDataGridView[1, i].Value = contractor.Name;

сontractorDataGridView[2, i].Value = contractor.ContactInfo;

сontractorDataGridView[3, i].Value = contractor.BankDetails;

сontractorDataGridView[4, i].Value = contractor.UTN;

сontractorDataGridView[5, i].Value = contractor.OKPO;

i++;

}

}

}

private void ProductType\_Enter(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

typeProductDataGridView.Rows.Clear();

int i = 0;

foreach (var type in db.ProductType)

{

typeProductDataGridView.Rows.Add();

typeProductDataGridView[0, i].Value = type.ID;

typeProductDataGridView[1, i].Value = type.Name;

i++;

}

}

}

private void Product\_Enter(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

productDataGridView.Rows.Clear();

int i = 0;

foreach (var product in db.Product)

{

productDataGridView.Rows.Add();

productDataGridView[0, i].Value = product.ID;

productDataGridView[1, i].Value = product.Name;

productDataGridView[2, i].Value = product.ProductType.Name;

productDataGridView[3, i].Value = product.Cost;

i++;

}

}

}

private void Request\_Enter(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

requestDataGridView.Rows.Clear();

int i = 0;

foreach (var request in db.ProductOrder)

{

requestDataGridView.Rows.Add();

requestDataGridView[0, i].Value = request.ID;

requestDataGridView[1, i].Value = "Заявка №" + request.ID;

requestDataGridView[2, i].Value = request.OrderDate;

i++;

}

}

}

private void employeeAddButton\_Click(object sender, EventArgs e)

{

EmployeeForm form = new EmployeeForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Employee\_Enter);

}

private void employeeChangeButton\_Click(object sender, EventArgs e)

{

EmployeeForm form = new EmployeeForm(Convert.ToInt32(employeeDataGridView.Rows[employeeDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Employee\_Enter);

}

private void employeeDeleteButton\_Click(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Employee.Find(Convert.ToInt32(employeeDataGridView.Rows[employeeDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

db.Employee.Remove(List);

db.SaveChanges();

}

Employee\_Enter(sender, e);

}

private void contractorAddButton\_Click(object sender, EventArgs e)

{

ContractorForm form = new ContractorForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Contractor\_Enter);

}

private void contractorChangeButton\_Click(object sender, EventArgs e)

{

ContractorForm form = new ContractorForm(Convert.ToInt32(сontractorDataGridView.Rows[сontractorDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Contractor\_Enter);

}

private void contractorDeleteButton\_Click(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Contractor.Find(Convert.ToInt32(сontractorDataGridView.Rows[сontractorDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

db.Contractor.Remove(List);

db.SaveChanges();

}

Contractor\_Enter(sender, e);

}

private void typeProductAddButton\_Click(object sender, EventArgs e)

{

TypeProductForm form = new TypeProductForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.ProductType\_Enter);

}

private void typeProductChangeButton\_Click(object sender, EventArgs e)

{

TypeProductForm form = new TypeProductForm(Convert.ToInt32(typeProductDataGridView.Rows[typeProductDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.ProductType\_Enter);

}

private void typeProductDeleteButton\_Click(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.ProductType.Find(Convert.ToInt32(typeProductDataGridView.Rows[typeProductDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

db.ProductType.Remove(List);

db.SaveChanges();

}

ProductType\_Enter(sender, e);

}

private void productAddButton\_Click(object sender, EventArgs e)

{

ProductForm form = new ProductForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Product\_Enter);

}

private void productChangeButton\_Click(object sender, EventArgs e)

{

ProductForm form = new ProductForm(Convert.ToInt32(productDataGridView.Rows[productDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Product\_Enter);

}

private void productDeleteButton\_Click(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.Product.Find(Convert.ToInt32(productDataGridView.Rows[productDataGridView.SelectedCells[0].RowIndex].Cells[0].Value));

db.Product.Remove(List);

db.SaveChanges();

}

Product\_Enter(sender, e);

}

private void requestAddButton\_Click(object sender, EventArgs e)

{

RequestForm form = new RequestForm();

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Request\_Enter);

}

private void requestChangeButton\_Click(object sender, EventArgs e)

{

RequestForm form = new RequestForm(Convert.ToInt32(requestDataGridView[0, requestDataGridView.CurrentRow.Index].Value));

form.Show();

form.FormClosed += new FormClosedEventHandler(this.Request\_Enter);

}

private void tnAddButton\_Click(object sender, EventArgs e)

{

TnForm form = new TnForm();

form.Show();

//form.FormClosed += new FormClosedEventHandler(this.Request\_Enter);

}

private void ttnAddButton\_Click(object sender, EventArgs e)

{

TtnForm form = new TtnForm();

form.Show();

//form.FormClosed += new FormClosedEventHandler(this.Request\_Enter);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

namespace TemplateProject

{

public partial class RequestForm : Form

{

enum Month

{

Января = 1,

Февраля,

Марта,

Апреля,

Мая,

Июня,

Июля,

Августа,

Сентября,

Октября,

Ноября,

Декабря

}

private int ID { get; set; }

public RequestForm(int ID = -1)

{

InitializeComponent();

this.ID = ID;

}

private void RequestForm\_Load(object sender, EventArgs e)

{

using (SharagaContext db = new SharagaContext())

{

foreach (var contractor in db.Contractor)

contractorComboBox.Items.Add(contractor.Name);

foreach (var employee in db.Employee)

employeeComboBox.Items.Add(employee.FIO);

foreach (var product in db.Product)

((DataGridViewComboBoxColumn)ProductDataGridView.Columns[2]).Items.Add(product.Name);

if (ID != -1)

{

var order = db.ProductOrder.Find(ID);

var counter = 0;

employeeComboBox.Text = order.Employee.FIO;

contractorComboBox.Text = order.Contractor.Name;

foreach (var item in order.OrderInfo)

{

ProductDataGridView.Rows.Add();

ProductDataGridView[0, counter].Value = item.ID;

ProductDataGridView[1, counter].Value = item.ProductID;

ProductDataGridView[2, counter].Value = item.Product.Name;

ProductDataGridView[4, counter].Value = item.OrderCount;

counter++;

}

}

}

}

private void SaveButton\_Click(object sender, EventArgs e)

{

var order = new ProductOrder();

using (SharagaContext db = new SharagaContext())

{

order = new ProductOrder()

{

ContractorID = db.Contractor.First(x => x.Name == contractorComboBox.Text).ID,

EmployeeID = db.Employee.First(x => x.FIO == employeeComboBox.Text).ID,

OrderDate = DateTime.Now

};

db.ProductOrder.Add(order);

for (int i = 0; i < ProductDataGridView.Rows.Count; i++)

{

db.OrderInfo.Add(new OrderInfo()

{

ProductOrder = order,

ProductID = Convert.ToInt32(ProductDataGridView[1, i].Value),

OrderCount = Convert.ToInt32(ProductDataGridView[4, i].Value),

});

}

db.SaveChanges();

}

this.Close();

}

private void AddRowButton\_Click(object sender, EventArgs e)

{

ProductDataGridView.Rows.Add();

}

private void Value\_Changed(object sender, DataGridViewCellEventArgs e)

{

if (e.ColumnIndex == 2 && ProductDataGridView.Rows.Count > 0)

{

using (SharagaContext db = new SharagaContext())

{

var name = ProductDataGridView[2, e.RowIndex].Value.ToString();

var product = db.Product.First(x => x.Name == name);

ProductDataGridView[1, e.RowIndex].Value = product.ID;

ProductDataGridView[3, e.RowIndex].Value = product.ProductType.Name;

ProductDataGridView[5, e.RowIndex].Value = product.Cost;

ProductDataGridView[6, e.RowIndex].Value = product.Cost \* Convert.ToInt32(ProductDataGridView[4, e.RowIndex].Value);

}

}

else if (e.ColumnIndex == 4 && ProductDataGridView.Rows.Count > 0)

{

using (SharagaContext db = new SharagaContext())

{

var name = ProductDataGridView[2, e.RowIndex].Value.ToString();

var product = db.Product.First(x => x.Name == name);

ProductDataGridView[6, e.RowIndex].Value = product.Cost \* Convert.ToInt32(ProductDataGridView[4, e.RowIndex].Value);

}

}

}

private void PrintButton\_Click(object sender, EventArgs e)

{

var wordApp = new Word.Application();

wordApp.Visible = false;

try

{

var wordDocument = wordApp.Documents.Open($"{Environment.CurrentDirectory}/Templates/Договор.docx");

var contractor = new Contractor();

var employee = new Employee();

using (SharagaContext db = new SharagaContext())

{

contractor = db.Contractor.First(x => x.Name == contractorComboBox.Text);

employee = db.Employee.First(x => x.FIO == employeeComboBox.Text);

}

ReplaceWordStub("{day}", DateTime.Now.Day.ToString(), wordDocument);

ReplaceWordStub("{month}", ((Month)DateTime.Now.Month).ToString(), wordDocument);

ReplaceWordStub("{year}", DateTime.Now.Year.ToString(), wordDocument);

ReplaceWordStub("{contractor}", contractorComboBox.Text, wordDocument);

ReplaceWordStub("{contractorName}", contractorComboBox.Text, wordDocument);

ReplaceWordStub("{contractorAddress}", contractor.ContactInfo, wordDocument);

ReplaceWordStub("{contractorDetails}", contractor.BankDetails, wordDocument);

ReplaceWordStub("{contractorUtn}", contractor.UTN, wordDocument);

ReplaceWordStub("{contractorOKPO}", contractor.OKPO, wordDocument);

ReplaceWordStub("{contractorEmployee}", employeeComboBox.Text, wordDocument);

ReplaceWordStub("{Employee}", employeeComboBox.Text, wordDocument);

wordDocument.SaveAs2($"{Environment.CurrentDirectory}/Documents/Договор.docx");

wordApp.Visible = true;

}

catch

{

MessageBox.Show("Ошибка");

}

}

private void ReplaceWordStub(string stubToReplace, string text, Word.Document wordDocument)

{

var range = wordDocument.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubToReplace, ReplaceWith: text);

}

private void tnButton\_Click(object sender, EventArgs e)

{

Excel.Application excelApp = new Excel.Application();

Excel.Workbook workbook = excelApp.Workbooks.Open($"{Environment.CurrentDirectory}/Templates/ТН.xls");

Excel.Worksheet worksheet = workbook.ActiveSheet;

using (SharagaContext db = new SharagaContext())

{

worksheet.Range["I20"].Value = db.Contractor.First(x => x.Name == contractorComboBox.Text).Name;

worksheet.Range["I22"].Value = db.Contractor.First(x => x.Name == contractorComboBox.Text).Name;

}

worksheet.Range["O17"].Value = DateTime.Now.Day;

worksheet.Range["R17"].Value = ((Month)DateTime.Now.Month).ToString();

worksheet.Range["W17"].Value = DateTime.Now.Year;

using (SharagaContext db = new SharagaContext())

{

var row = 32;

foreach (var item in db.ProductOrder.Find(ProductDataGridView[0, 0].Value).OrderInfo)

{

if (row > 32)

InsertRow(33, worksheet);

worksheet.Range[$"C{row}"].Value = item.Product.Name;

worksheet.Range[$"L{row}"].Value = item.OrderCount;

worksheet.Range[$"O{row}"].Value = item.Product.Cost;

worksheet.Range[$"R{row}"].Value = item.OrderCount \* item.Product.Cost;

row++;

}

}

excelApp.AlertBeforeOverwriting = false;

workbook.SaveAs($"{Environment.CurrentDirectory}/Documents/ТН.xls");

excelApp.Quit();

}

public void InsertRow(int rowNum, Excel.Worksheet \_workSheet)

{

Excel.Range cellRange = (Excel.Range)\_workSheet.Cells[rowNum, 1];

Excel.Range rowRange = cellRange.EntireRow;

rowRange.Insert(Excel.XlInsertShiftDirection.xlShiftDown, false);

Excel.Range \_excelCells = (Excel.Range)\_workSheet.get\_Range($"C{rowNum}", $"H{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"I{rowNum}", $"K{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"L{rowNum}", $"N{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"O{rowNum}", $"Q{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"R{rowNum}", $"U{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"V{rowNum}", $"Y{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"Z{rowNum}", $"AC{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"AD{rowNum}", $"AG{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

\_excelCells = (Excel.Range)\_workSheet.get\_Range($"AH{rowNum}", $"AK{rowNum}").Cells;

\_excelCells.Merge(Type.Missing);

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TemplateProject

{

public partial class TypeProductForm : Form

{

private int ID { get; }

public TypeProductForm(int ID = -1)

{

InitializeComponent();

this.ID = ID;

if (ID != -1)

{

using (SharagaContext db = new SharagaContext())

{

var List = db.ProductType.Find(ID);

ProductTypeNameTextBox.Text = List.Name;

}

}

}

private void addButton\_Click(object sender, EventArgs e)

{

if (ID == -1)

{

using (SharagaContext db = new SharagaContext())

{

db.ProductType.Add(new ProductType()

{

Name = ProductTypeNameTextBox.Text,

});

db.SaveChanges();

this.Close();

}

}

else

{

using (SharagaContext db = new SharagaContext())

{

var List = db.ProductType.Find(ID);

List.Name = ProductTypeNameTextBox.Text;

db.SaveChanges();

this.Close();

}

}

}

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("Contractor")]

public partial class Contractor

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Contractor()

{

ProductOrder = new HashSet<ProductOrder>();

RealizeOrder = new HashSet<RealizeOrder>();

}

public int ID { get; set; }

[Required]

[StringLength(100)]

public string Name { get; set; }

[Required]

public string ContactInfo { get; set; }

[Required]

public string BankDetails { get; set; }

[Required]

[StringLength(9)]

public string UTN { get; set; }

[Required]

[StringLength(12)]

public string OKPO { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<ProductOrder> ProductOrder { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<RealizeOrder> RealizeOrder { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("Employee")]

public partial class Employee

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Employee()

{

ProductOrder = new HashSet<ProductOrder>();

RealizeOrder = new HashSet<RealizeOrder>();

}

public int ID { get; set; }

[Required]

[StringLength(100)]

public string FIO { get; set; }

[Required]

[StringLength(50)]

public string Position { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<ProductOrder> ProductOrder { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<RealizeOrder> RealizeOrder { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("OrderInfo")]

public partial class OrderInfo

{

public int ID { get; set; }

public int OrderID { get; set; }

public int ProductID { get; set; }

public int OrderCount { get; set; }

public virtual ProductOrder ProductOrder { get; set; }

public virtual Product Product { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("Product")]

public partial class Product

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public Product()

{

OrderInfo = new HashSet<OrderInfo>();

RealizeOrderInfo = new HashSet<RealizeOrderInfo>();

}

public int ID { get; set; }

public int TypeID { get; set; }

[Required]

[StringLength(50)]

public string Name { get; set; }

public double Cost { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<OrderInfo> OrderInfo { get; set; }

public virtual ProductType ProductType { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<RealizeOrderInfo> RealizeOrderInfo { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("ProductOrder")]

public partial class ProductOrder

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public ProductOrder()

{

OrderInfo = new HashSet<OrderInfo>();

RealizeOrderInfo = new HashSet<RealizeOrderInfo>();

}

public int ID { get; set; }

public int ContractorID { get; set; }

public int EmployeeID { get; set; }

public DateTime OrderDate { get; set; }

public virtual Contractor Contractor { get; set; }

public virtual Employee Employee { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<OrderInfo> OrderInfo { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<RealizeOrderInfo> RealizeOrderInfo { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("ProductType")]

public partial class ProductType

{

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]

public ProductType()

{

Product = new HashSet<Product>();

}

public int ID { get; set; }

[Required]

[StringLength(50)]

public string Name { get; set; }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]

public virtual ICollection<Product> Product { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("RealizeOrder")]

public partial class RealizeOrder

{

public int ID { get; set; }

public int ContractorID { get; set; }

public int EmployeeID { get; set; }

public DateTime OrderDate { get; set; }

public virtual Contractor Contractor { get; set; }

public virtual Employee Employee { get; set; }

}

}

namespace TemplateProject

{

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Data.Entity.Spatial;

[Table("RealizeOrderInfo")]

public partial class RealizeOrderInfo

{

public int ID { get; set; }

public int OrderID { get; set; }

public int ProductID { get; set; }

public int OrderCount { get; set; }

public virtual Product Product { get; set; }

public virtual ProductOrder ProductOrder { get; set; }

}

}