**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**«ГОМЕЛЬСКИЙ ТОРГОВО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ»**

**БЕЛКООПСОЮЗА**

**Дневник-отчёт**

**по технологической практике**

**учащегося Шевелева Ильи Владимировича**

**курса 4 группы П-41**

**Специальность *2-40 01 01 «Программное обеспечение информационных технологий»***

**Специализация *2-40 01 01 35 «Программное обеспечение обработки экономической и деловой информации»***

Отметка по практике

с учетом собеседования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от колледжа Коновалов Павел Сергеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021г.

**Общие сведения о базе практики и руководителях практики**

1. **Место прохождения практики ОДО «ОНИКС»**

название организации

1. **Срок прохождения практики** с 11 ноября 2020 г. по 5 января 2021 г.
2. **Руководитель практики от организации**  Щеников Д.В., главный инженер, высшее образование, 11 лет

Фамилия, имя, отчество, должность, стаж работы

1. **Руководитель практики от колледжа** Коновалов Павел Сергеевич

Фамилия, имя, отчество преподавателя

**Подтверждение о прохождении практики,**

**вводного инструктажа по охране труда,**

**первичного инструктажа на рабочем месте**

Шевелев Илья Владимирович, учащийся учреждения образования «Гомельский торгово-экономический колледж» Белкоопсоюза, в соответствии с приказом от 09.11.2020 № 96 прошел технологическую практику с 11.11.2020 по 05.01.2021.

С Шевелевым Ильей Владимировичем проведен вводный инструктаж по охране труда и первичный инструктаж на рабочем месте.

**Глава администрации** Щеников Д.В.

**Печать**

СОГЛАСОВАНО УТВЕРЖДАЮ

Председатель цикловой комиссии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.С.Васьковцова

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

должность, подпись, инициалы, фамилия

**Учебная программа практики и календарный график прохождения практики**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование тем и  содержание работы | Место  практики | Количество дней | Сроки выполнения  в датах |
|  | **Вводное занятие. Организационные мероприятия. Ознакомление с целями, задачами и содержанием практики.** | организация | **1** | 11.11 |
|  | **Ознакомление с базой прохождения практики, структурой данной организации, перечнем и характером решаемых задач.** | организация | **1** | 12.11 |
|  | **Ознакомление (с перемещением по рабочим местам) с технологическим процессом обработки информации в подразделениях организации, IT-отделах**  Анализ технологического процесса обработки информации.  Анализ технологического процесса разработки программного обеспечения.  Оценка инструментальных средств проектирования, моделирования, программирования и тестирования ПО. | отдел информационных технологий или рабочее место программиста | **12**  4  3  5 | 13.11,14.11,  16.11,17.11  18.11,19.11,  20.11  21.11,23.11, 24.11,25.11, 26.11 |
|  | **Сопровождение производственных задач**  Участие в сопровождении задач, функционирующих в организации. Сопровождение бухгалтерских программ, мониторинг сайтов.  Ведение идентификационной базы пользователей пункта коллективного доступа к сети Интернет, настройка почтового сервера.  Инсталляция и настройка программного обеспечения.  Обслуживание системы учета рабочего времени. Учет обмена бизнес-документами.  Протоколирование антивирусной профилактики.  Составление тестов для проверки функционирования производственных задач. | отдел информационных технологий или рабочее место программиста | **18**  5  2  2  5  2  2 | 27.11,28.11, 30.12, 01.12, 02.12  03.12, 04.12  05.12, 07.12  08.12,09.12, 10.12,11.12, 12.12  14.12,15.12  16.12,17.12 |
|  | **Разработка программных средств индивидуального задания**  Анализ предметной области индивидуального задания. Разработка технического задания на программное средство.  Построение моделей программной системы.  Кодирование, тестирование программного средства.  Оценка возможности интеграции разработанного программного средства в комплекс программ организации.  Опытная эксплуатация. Анализ качества выполненного программного средства.  Расчет стоимости программного средства.  Разработка разделов отчета по практике.  Подготовка презентации организации-базы практики. | отдел информационных технологий или рабочее место программиста | **11**  2  1  2  1  2  1  1  1 | 18.12  19.12  20.12  22.12  23.12,  24.12  26.12  28.12  29.12  30.12  31.12 |
|  | **Обобщение материалов по практике.**  Систематизация материалов по практике.  Оформление отчета по практике. |  | **2**  1 | 02.01  04.01 |
|  | **Подведение итогов практики.** |  | **1** | **05.01** |
|  | ***ИТОГО*** |  | **46** |  |

**Руководитель практики от организации** Щеников Д.В.

**Практикант**  Шевелев И.В.

**Замечание руководителя практики от колледжа**

|  |  |  |
| --- | --- | --- |
| **Дата проверки** | **Содержание замечаний** | **Фамилия и подпись проверяющего** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Содержание дневника-отчёта**

|  |  |  |
| --- | --- | --- |
| **№ п/п,**  **дата** | **Наименование темы и**  **содержание работы** | **Замечания и подпись руководителя от организации** |
| 11.11.2020  12.11.2020  13.11.2020  14.11.2020  16.11.2020  17.11.2020  18.11.2020  19.11.2020  20.11.2020  21.11.2020  23.11.2020    24.11.2020  25.11.2020  26.11.2020  27.11.2020  28.11.2020  30.11.2020  01.12.2020  02.12.2020  03.12.2020  04.12.2020  05.12.2020  07.12.2020  08.12.2020  09.12.2020  10.12.2020  11.12.2020  12.12.2020  14.12.2020  15.12.2020  16.12.2020  17.12.2020  18.12.2020  19.12.2020  21.12.2020  22.12.2020  23.12.2019  24.12.2020  26.12.2020  28.12.2019  29.12.2020  30.12.2020  31.12.2020  02.01.2021  04.01.2020  05.01.2020 | **Вводное занятие. Организационные мероприятия. Ознакомление с целями, задачами и содержанием практики.**  Место прохождения практики – ОДО «ОНИКС».  Изучил цели, задачи и содержание технологической практики.  Основной целью технологической практики является систематизация, закрепление и расширение знаний и навыков, полученных при изучении специальных дисциплин: «Основы алгоритмизации и программирования», «Базы данных и СУБД», «Программные средства создания Интернет-приложений» и др.  Основными задачами технологической практики являются:   * приобретение учащимися профессиональных умений и навыков по специальности; * закрепление и углубление и систематизация знаний по специальным дисциплинам; * развитие профессионального мышления и профессиональной самостоятельности учащихся; * формирование навыков организаторской деятельности на производстве и в условиях коллектива;   Содержанием технологической практики является:   * вводное занятие. Организационные мероприятия. Ознакомление с целями, задачами и содержанием практики; * ознакомление с базой прохождения практики, структурой данной организации, перечнем и характером решаемых задач; * ознакомление (с перемещением по рабочим местам) с технологическим процессом обработки информации в подразделениях организации, IT-отделах, данный пункт включает в себя следующее: * анализ технологического процесса обработки информации; * анализ технологического процесса разработки программного обеспечения; * оценка инструментальных средств проектирования, моделирования, программирования и тестирования ПО; * сопровождение производственных задач, оно включает в себя: * участие в сопровождении задач, функционирующих в организации. Сопровождение бухгалтерских программ, мониторинг сайтов; * ведение идентификационной базы пользователей пункта коллективного доступа к сети Интернет, настройка почтового сервера; * инсталляция и настройка программного обеспечения; * обслуживание системы учета рабочего времени. Учет обмена бизнес-документами; * протоколирование антивирусной профилактики; * составление тестов для проверки функционирования производственных задач; * разработка программных средств индивидуального задания; * анализ предметной области индивидуального задания. Разработка технического задания на программное средство; * построение моделей программной системы; * кодирование, тестирование программного средства; * оценка возможности интеграции разработанного программного средства в комплекс программ организации; * опытная эксплуатация. Анализ качества выполненного программного средства; * расчет стоимости программного средства; * разработка разделов отчета по практике; * подготовка презентации организации-базы практики; * обобщение материалов по практике, данный пункт включает в себя: * систематизация материалов по практике; * оформление отчета по практике. * подведение итогов практики.   Ознакомился с внутренним трудовым распорядком работников ОДО «ОНИКС».  Режим работы предприятия ОДО «ОНИКС»:  Понедельник - Суббота с 8:30 до 17:00  Обед с 13:00 до 14:00  Суббота, воскресенье – выходной.  Был ознакомлен с уставом ОДО «ОНИКС».  Инженером по промышленной безопасности, охране труда и экологии был проведён вводный инструктаж по охране труда, первичный инструктаж на рабочем месте.  Был ознакомлен с инструкцией по охране труда при работе на персональном компьютере ОДО «ОНИКС»  **Ознакомление с базой прохождения практики, структурой данной организации, перечнем и характером решаемых задач.**  Ознакомился с базой прохождения практики ОДО «ОНИКС». Основные задачи решаемые предприятием:   * Производство отопительных и обогревательных приборов * Обеспечение аппаратурой застройщиков * Продажа обогревательных приборов частным и юридическим лицам   Изучил структуру частного предприятия:   * Директор * Заместитель директора * Главный инженер * Главный бухгалтер * Заместитель по производственной работе   Ознакомился с должностными обязанностями техника, также был ознакомлен со структурой частного предприятия ОДО «ОНИКС».  **Ознакомление (с перемещением по рабочим местам) с технологическим процессом обработки информации в подразделениях организации, IT-отделах**  *Анализ технологического процесса обработки информации.*  Во время прохождения технологической практики на частном предприятии ОДО «ОНИКС» ознакомился с обобщённой схемой технологического процесса обработки информации, также был ознакомлен со штатным расписанием работников и их рабочими местами, ознакомился с делением предприятия по органам обработки информации.  Технологический процесс призван автоматизировать обработку исходной информации за счет привлечения технических средств, сократить финансовые и трудовые затраты, обеспечить высокую степень достоверности информации. Для конкретной задачи той или иной предметной области технологический процесс обработки информации разрабатывается индивидуально.  Технологический процесс обработки информации с использованием ЭВМ включает в себя следующие операции:   * прием и комплектование первичных документов; * подготовка первичных документов и контроль; * ввод данных в ПК; * контроль, результаты которого задаются на терминал.   Технологическая операция – взаимосвязанная совокупность действий, выполняемая над информацией на одном рабочем месте в процессе ее преобразования для достижения общей цели технологического процесса. При этом важными являются время преобразования и качества результативной информации. Технологические операции обычно выполняются целыми совокупностями, образуя этапы:   * начальный; * подготовительный; * основной; * заключительный.   В настоящее время прослеживается к максимальному приближению информационных и программных ресурсов пользователю. ПК, работающие в сети, имеют существенное преимущество перед АРМ, работающими в режиме разделения времени.  В отделах частного предприятия ОДО «ОНИКС» для обработки информации ПК применяются на рабочих местах специалистов ответственных за сбор первичной информации с целью автоматизации, подготовки и контроля собранной и необработанной информации. Это в свою очередь позволяет объединить фазы подготовки и ввода информации  Рабочее место каждого специалиста частного предприятия ОДО «ОНИКС» оснащено современной техникой, необходимой для работы. Большинство ПК имеет следующие характеристики:   * Процессор: Intel(R) Core™ i5-8250U CPU @ 1.60 GHz 1.80 GHz; * ОЗУ: 3.00 GB; * Тип системы: Microsoft Windows 7 64 bit; * Видеоадаптеры: Intel® UHD Graphics 540; * Сетевые адаптеры: Intel Dual Band Wireless-AC 3165,   Realtek PCIe GbE Family Controller;   * Монитор диагональю в 20 дюймов; * Клавиатура, мышь.   Рабочее место инженера-программиста предназначено для обработки и хранения информации. Она может поступать из нескольких источников:   * флеш-накопитель; * почту; * SSD; * CD/DVD-диски; * Интернет. * Факс   Для обработки информации на данном рабочем месте используется следующее ПО:   * Microsoft Windows; * Microsoft Office; * Проводник; * Блокнот; * Abbby Fine Reader; * AutoCAD;   Инженер-программист имеет право постоянного хранения файлов защищенными данными на жестком магнитном диске. Для защиты ПК от несанкционированного доступа на рабочем месте специалиста установлен стандартный пароль Windows.  Для разработки документов на ПК установлен MS Office 2013.  Чтение документов в формате .pdf осуществляется с помощью программы ABBYY Fine Reader 11.  *Анализ технологического процесса разработки ПО*  Технологический процесс разработки ПО – это множество направлений деятельности, методов, практических приемов и процедур, используемое для разработки и сопровождения ПО и связанных с ним продуктов.  Результативность технологического процесса характеризует реальные результаты, достигнутые благодаря следованию процессу.  Изучив технологический процесс разработки ПО на предприятии, выяснило, что сайт предприятия был разработан используя язык гипертекстовой разметки HTML, PHP, JavaScript.  HTML – стандартизированный язык разметки документов во всемирной паутине. Большинство веб-страниц имеют содержание разметки на языке HTML. Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.  PHP – cкриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений.  JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript.  Благодаря всем этим языкам тестирования, специалисты отдела имеют все необходимое, что-бы делать сайт гибким и соответствовать современным критериям сайта.  JavaScript является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам — функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания— что придаёт языку дополнительную гибкость.  Основные возможности языка JavaScript:   * создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т.п. * реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т.п. * получать и устанавливать cookie, запрашивать данные, выводить сообщения.   При тестировании программ, специалисты государственного учреждения Администрация Советского района используют методы «черного» и «белого» ящика.  Чёрный ящик – основная идея тестирования системы как чёрного ящика состоит в том что все материалы, которые доступны тестировщику – требования на систему описывающие её поведение и сама система работать с которой он может только подавая на её входы некоторые внешние воздействия и наблюдая на выходах некоторый результат. Все внутренние особенности системы реализации скрыты от тестировщика и таким образом система представляет собой чёрный ящик правильность поведения которого по отношению к требованиям и предстоит проверить.  Белый ящик – при тестировании системы тестировщик имеет доступ не только к требованиям системы, её входам и выходам, но и к внутренней структуре, видит её программный код. Доступность программного кода расширяет возможности тестировщика, тем, что он может видеть в соответствие требований участком программного кода и определять тем самым на весь ли программный код существуют требования.  Преимущества использования метода «белого ящика»:   * знание внутренней структуры кодирования является предпосылкой, поэтому легко выяснить какой тип вводы данных может помочь в эффективном тестировании в приложении; * помогает оптимизировать код; * устранение дополнительных строк кода, которые могут привести к скрытым дефектам;   Недостатки «белого ящика»:   * поскольку знание кода и внутренней структуры является необходимым условием, то требуются квалифицированный тестировщик, что увеличивает стоимость.   Преимущества использования «чёрного ящика»:   * тестирование методом «чёрного ящика» позволяет найти ошибки, которые невозможно обнаружить методом «белого ящика»; * «чёрный ящик» позволяет быстро выявить ошибки в функциональных спецификациях; * тестировщику не нужна дополнительная квалификация; * тестирование проходит «с позиции пользователя»; пользователь всегда прав, он конечный потребитель практически любого ПО, а значит, ему должно быть удобно, комфортно и понятно; * составлять тест-кейсы можно сразу после подготовки спецификации.   Недостатки «чёрного ящика»:   * основным недостатком метода «чёрного ящика» является возможность пропуска границ и переходов, которые не очевидны из спецификации, но есть в реализации кода (собственно, это и заставляет тестировщиков использовать метод «белого ящика»); * можно протестировать только небольшое количество возможных вводимых (входящих) значений; многие варианты остаются без проверки; * тесты могут быть избыточными, если разработчик уже проверил данную функциональность (например, Unit-тестом); * при отсутствии чёткой и полной спецификации проектировать тесты и тест-сценарии оказывается затруднительно.   *Оценка инструментальных средств проектирования, моделирования, программирования и тестирования ПО*  Инструментальное ПО – программное обеспечение, предназначенное для использования в ходе проектирования, разработки и сопровождения программ.  Инструментарий программирования – средства, предназначенные для создания ПО, системного и прикладного ПО. Его составляют разнообразные языки и среды программирования.  С точки зрения ИСО 91.26 качество программных средств можно определить как совокупную характеристику исследуемого ПО, с учетом следующих составляющих: надежность, сопровождаемость, практичность, эффективность, мобильность и функциональность.  Для моделирования ПО в организации были использованы визуальные модели.  Использование визуализации качества способа анализа данных обусловливает необходимость качественных критериев, благодаря которых можно использовать визуальные приемы, соответствующие специфике поставленной задачи и особенностям данных. Отсутствие полноценной классификации категории данных, которая позволила бы быстро с высокой степенью однозначности делать выбор необходимых инструментов визуализации, является обстоятельствам, препятствующим превращению визуальных моделей данных в результативный исследовательский метод.  Одной из задач визуализации информации является выполнение коммуникативной функции между произвольными данными и когнитивными усилиями исследователей. Эффективность этого процесса может быть обеспечена только при учете и понимании особенностей зрительного восприятия человека. С учетом этого визуализация позволяет решить две задачи: выбирать способ представления исходной информации, позволяет производить эффективный анализ и поиск ответов на поставленные вопросы.  Состав модели для каждого проекта включает в себя следующие факторы:   * Сложность проектируемой системы; * Необходимость полноты ее описания; * Соответствующие знания и навыки участников проекта; * Время, отведенное для проектирования.   ОДО «ОНИКС» использует для решения своих задач языки низкого уровня: C, Assembly.  Си (англ. C) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах сотрудником Bell Labs Деннисом Ритчи как развитие языка Би.  Для разработки проектов предприятие использует CASE - систему – Rational Rose.  Программа Rational Rose отлично подходит данному предприятию, т.к. программа предназначена для автоматизации этапов анализа и проектирования ПО, а также для генерации кодов на различных языках и выпуска проектной документации.  Принципиальное отличие Rational Rose от других средств заключается в объектно-ориентированном подходе. Графические модели, создаваемые с помощью этого средства, основаны на объектно-ориентированных принципах и языке [UML](https://www.kpms.ru/Automatization/BPM.htm#BPM_UML) (Unified Modeling Language). Обладая способностью представлять пользовательский интерфейс отдельно от бизнеса логики и данных, Rational Rose позволяет держать контроль над разработкой ПО. Инструменты моделирования позволяют разработчикам создавать целостную архитектуру процессов предприятия, сохраняя все взаимосвязи и управляющие воздействия между различными уровнями иерархии.  Находясь на ОДО «ОНИКС», ознакомился с интерфейсом программы Rational Rose.  Данная программа имеет простой и интуитивно понятный интерфейс, необходимых для выполнения требуемых задач.  При запуске программы откроется главное окно (рисунок 1):    Рисунок 1 – Главное окно программы Rational Rose  Главное окно содержит в себе:   * меню; * панель инструментов; * панель сообщений; * панель специальных инструментов; * навигатор моделей.   Навигатор модели помогает быстро переходить от одной модели к другой.  Основная панель инструментов содержит в себе следующие пункты (рисунок 2):    Рисунок 2 – Основная панель инструментов Rational Rose   * создание новой модели; * открытие новой модели; * сохранение построенной модели; * обрезка модели; * копирование модели; * печать модели; * масштабирование.   Rational Rose имеет элемент «Браузер», который позволяет легко и быстро осуществлять навигацию внутри модели. Всё добавленное к модели: классы, компоненты, актёры и т.д. – выводится в окне браузера.  Браузер позволяет:   * добавлять к модели элементы (сценарии, классы, компоненты, диаграммы и т.д.); * просматривать существующие элементы модели; * просматривать существующие отношения между элементами модели; * перемещать элементы модели; * переименовывать элементы модели; * добавлять элементы модели к диаграмме; * группировать элементы в пакеты; * открывать диаграммы.   **Сопровождение производственных задач**  *Участие в сопровождении задач, функционирующих в организации, сопровождение бухгалтерских программ, мониторинг сайта.*  Проходя технологическую практику в ОДО «ОНИКС», изучил бухгалтерскую программу – Инфо-бухгалтер. Популярная программа автоматизации учета Инфо-Бухгалтер идеально подходит для тех, кто ведет свои дела с прицелом на активное развитие в будущем. Программа помогает осуществлять хранение поступившей информации за определенный период, вести учет поступивших в организацию основных средств, выделять из них введенные в производство и списанные, планирование работ и управление ими, автоматизировать коммерческую деятельность и многое другое.  Программа Инфо-Бухгалтер подходит для государственного учреждения Администрация Советского района, так как с ее помощью осуществляется документооборот. Он сводится к обработке входящей и исходящей информации.  Данный программный продукт обеспечивает комплексную поддержку всем бизнес процессам и предназначен для автоматизации оперативной деятельности.  Специалисты имеют возможность быстро формировать необходимые документы, а руководство – оперативно получать и использовать данные о различных аспектах деятельности предприятия. Возможности и преимущества Инфо-Бухгалтер:  * Комплексная автоматизация бухгалтерского, налогового, складского, кадрового, управленческого учета предприятий и индивидуальных предпринимателей. * Возможность использования бухгалтерской программы на предприятиях любых форм собственности, масштабов и видов деятельности, любых систем налогообложения: ОРН, УСН, ЕНВД, ЕСХН, ПСН (Патент). * Ведение бухгалтерского, налогового, кадрового и складского учета сразу нескольких предприятий с возможностью автоматического формирования всех необходимых отчетов. * Формирование учетной политики предприятия с автоматической настройкой программы на особенности бухгалтерского и налогового учета. * Регулярное автоматическое обновление бухгалтерской программы через интернет по мере изменения Законодательства. * Гибкость настройки, легко выполняемая бухгалтерами без обращения к программистам. Учет региональных и отраслевых особенностей. * Надежность сохранения данных и возможность протоколирования действий пользователей. * Возможность обмена данными с другими бухгалтерскими программами и подключения торгового оборудования. * Совместимость с любыми системами передачи отчетности через Интернет, а также с системами Клиент-банк. * Отличное качество и высокие темпы развития популярной программы автоматизации бухгалтерского, налогового, складского, кадрового учета Инфо-Бухгалтер опираются на 26-летний успешный опыт создания программ для бухгалтеров и предпринимателей. * Легкость освоения, удобство применения, простота сопровождения.  Минусы программы Инфо-Бухгалтер: Трудности в настройке этой программы под конкретное предприятие. В такой программе совершенно невозможно работать с графикой, так как данная функция отсутствует. Других значимых недостатков в работе такого ПО выявлено не было.  Изучил возможности программы используемой в государственном учреждении, далее перешел к изучению установки программы.  Установка программы Инфо-Бухгалтер  Чтобы приступить к установке платформы, необходимо открыть папку, содержащую дистрибутив и запустить исполнимый файл Setup.exe. Ознакомившись с текстовым материалом, нажимаем «Далее».    Рисунок 3 - Начало установки программы 1С: Предприятие  Кнопка обзор позволяет выбрать дисковый раздел и папку для установки программы. Жмем «Далее» (рисунок 4).    Рисунок 4 - Выбор папки установки  В данном окне выберете необходимую вам версию программы. Жмем «Далее» (рисунок 5).    Рисунок 5 - Выбор версии программы  После начнётся процесс установки программы. По окончанию данного процесса, появится окно, оповещающее о успешном установлении программы (рисунок 6).    Рисунок 6 – Окончание установки программы  После окончания установки программы будет выведено окно создания Демонстрационной Базы Данных. Данная база содержит примеры работы с программой, т.е. заполненные данные в различных блоках. Данная база носит ознакомительный характер в ней можно опробовать выполнение операций и сравнить с уже введенными данными.  Во время прохождения технологической практики в организации провел мониторинг его сайта.  Данный сайт служит для следующих целей:   * Просмотр и ознакомление с новостями; * Получение информации о каком-либо продукте; * Оказание услуг; * Получение информации о контактах предприятия; * Получение информации о месте нахождения предприятия; * Получения информации о графике приёма граждан; * Регистрация звонков клиентам; * Электронное обращение в Администрацию.   Главная страница сайта знакомит с основными элементами производства, предоставляет быстрый доступ к подбору оборудования для клиента (рисунок 7).    Рисунок 7 - Главная страница предприятия  На главной странице представлена возможность перехода на ключевые страницы, возможность электронного обращения и узнать интересующую информацию (обратная связь), просмотр последний новостей, быстро прокрутить страницу в начало, просмотр корзины.  На главной странице предоставлена возможность по сайту, перехода на группы в социальных сетях (рисунок 8).    Рисунок 8 - Поиск по сайту  Переход в корзину предоставляет навигационная кнопка в шапке сайта (рисунок 9).    Рисунок 9 - Корзина  После выбора требуемого оборудования, оборудование попадает в коризну (рисунок 10).    Рисунок 10 – Работа корзины  На главной странице сайта включены такие разделы как:   * О компании; * Продукция; * Тех. информация; * Где купить?; * Контакты;   Так же на главной странице сайта можно перейти по интересующей вас новости (рисунок 11):    Рисунок 11 - Последние новости предприятия  В конце страницы есть ссылки на контакты и телефоны предприятия (рисунок 12):    Рисунок 12 - Контакты  *Ведение идентификационной базы пользователей пункта, коллективного доступа к сети интернет, настройка почтового сервера.*  Ознакомился с ведением идентификационной базы пользователя, пункта коллективного доступа, настройкой почтового сервера.  В организации используется локальная вычислительная сеть, обхватывающая территорию здания.  Точка доступа сети предназначена для координации обмена данными между сотрудниками. Рабочие компьютеры подключены к сетевому коммутатору (switch), которому в свою очередь подключен кабель. Локальная сеть состоит из компьютеров, работающих на 50 компьютеров, работающих на операционной системе Windows XP/7/10 x32.  В организации используется топология локальной сети «звезда».  Топология «звезда» — базовая топология компьютерной сети, в которой все компьютеры сети присоединены к центральному узлу образуя физический сегмент сети. Подобный сегмент сети может функционировать как отдельно, так и в составе сложной сетевой топологии (как правило, «дерево»). Весь обмен информацией идет исключительно через центральный компьютер или агрегат, на который таким способом возлагается очень большая нагрузка, поэтому ничем другим, кроме сети, он заниматься не может. Как правило, именно центральный компьютер или агрегат является самым мощным в сетевом отношении, и именно на него возлагаются все функции по управлению сетью и передаче данных.  Топология «звезда» на сегодняшний день стала основной при построении локальных сетей. Это произошло благодаря ее многочисленным достоинствам:   * выход из строя одной рабочей станции или повреждение ее кабеля не отражается на работе всей сети в целом; * отличная масштабируемость: для подключения новой рабочей станции достаточно проложить от коммутатора отдельный кабель; * легкий поиск и устранение неисправностей и обрывов в сети; * высокая производительность; * простота настройки и администрирования; * в сеть легко встраивается дополнительное оборудование.   Однако, как и любая топология, «звезда» не лишена недостатков:   * выход из строя центрального коммутатора обернется неработоспособностью всей сети; * дополнительные затраты на сетевое оборудование – устройство, к которому будут подключены все компьютеры сети (коммутатор); * число рабочих станций ограничено количеством портов в центральном коммутаторе.   Во время прохождения технологической практики на предприятии ознакомился с настройкой почтовых серверов.  В системе пересылки электронной почты, так обычно называют агент пересылки сообщений, это программа, созданная для возможностей осуществления информационного обмена между компьютерами посредством электронных сообщений. Работа с такими сообщениями происходит с помощью использования специального ПО – почтового клиента.  Почтовый сервер — это программа, созданная для возможности осуществления информационного обмена между компьютерами посредством электронных сообщений. Работа с такими сообщениями происходит с помощью использования специального программного обеспечения — почтового клиента.  На почтовом сервере есть два основных агента, отвечающих за проведение наиболее значимых операций: отправки (Mail Transfer Agent), получение данных (Mail Delivery Agent). Наиболее популярными почтовыми серверами на данный момент являются:   * GML; * Rambler; * Yandex; * Mail.ru; * UKR.net; * Outlook.com.   ОДО «ОНИКС» в качестве своего почтового сервера использует браузер Opera Mail.  Opera Mail — это простой и удобный почтовый клиент от компании Opera Software. Opera Mail поддерживает все необходимые функции, обладает удобным и доступным интерфейсом и включает набор дополнительных функций. Программа поддерживает получение почты по всем востребованным протоколам, поддерживает работу с новостными группами, умеет работать с самыми популярными сервисами (GMail, Mail.ru, Yandex.by), а также поддерживает RSS-потоки для получения новостей.   Opera Mail поддерживает работу с несколькими почтовыми ящиками; имеет удобный интерфейс на основе вкладок; поддерживает HTML-форматирование в сообщениях; включает простую, но эффективную систему фильтрации сообщений по набору различных параметров; имеет простую адресную книгу. В общем, все просто, удобно и доступно.  РОР3 – протокол позволяет рабочим станциям динамически получать доступ к своим почтовым ящикам, расположенным на сервере, предназначенном для обслуживания электронной почты предприятия.  Он позволяет только забрать почту из почтового ящика сервера на рабочую станцию сотрудника и удалить её из почтового ящика на сервере. Всю дальнейшую обработку почтовое сообщение проходит на компьютере специалиста.  IMAP — более современный протокол, поддерживающий прием и возможность управления электронной почтой прямо на почтовом сервере. При просмотре почты с использованием протокола POP3 все электронные письма скачиваются пользователю на компьютер и автоматически удаляются с сервера.  *Инсталляция и настройка программного обеспечения.*  Инсталляция и настройка ПО в организации осуществляется отделом технического обслуживания вычислительной техники, телекоммуникаций и систем охраны.  Установка ПО, инсталляция – процесс установки ПО на компьютер конечного пользователя.  Кроме установки или настройки, так же может потребоваться его восстановление. Такая необходимость возникает в случае повреждения файлов или осуществления вирусной атаки.  Инсталляция ПО производится по следующему алгоритму:   * Запуск (Setup) exe файла – файла установщика, находящимся на инсталляционном устройстве; * Последующие шаги установки выполняются автоматически; * Если существует потребность установки серверной части ПО, необходимо войти в систему с правами администратора; * По окончанию инсталляции требует обязательная перезагрузка ОС.   Во время прохождения технологической практики осуществил установку и настройку программы браузера Opera.  При запуске установки появляется выбора языка установки (рисунок 13). После выбора нажимаем кнопку «Ок»:    Рисунок 13 – Выбор языка установки  Затем откроется мастер установки InstallShield (рисунок 14). Нажимаем кнопку «Далее»:    Рисунок 14 – мастер установки InstallShield  После предложат ознакомится с лицензионным соглашением программы (рисунок 15). Ознакомившись с лицензионным соглашением, нажимаем кнопку «Я принимаю»:    Рисунок 15 – Лицензионное соглашение  Затем начнётся установка браузера Opera на ПК. После завершения мастера установки нужно перезагрузить ПК.  *Обслуживание системы учета рабочего времени, учет обмена бизнес-документами*  Система учета прихода и ухода сотрудников позволяет вести контроль рабочего времени, которое реально отработано сотрудниками. Ведение такого учета позволяет значительно сократить потери временных трудозатрат и повысить уровень трудовой дисциплины работника.  Целесообразность учета рабочего времени определяется финансовой стороной вопроса.  Система учета рабочего времени ­– это хороший инструмент для оптимизации работы и повышения эффективности рабочего времени.  В ОДО «ОНИКС» для учёта рабочего времени используется Инфо-Бухгалтер.  Каждому сотруднику при приеме на работу выдается его личный табель, для фиксирования его времени проведения на рабочем месте.  Для учета рабочего времени штатных сотрудников используются графики работ. Графики могут быть общими или индивидуальными. Общий график указывается при приеме сотрудника на работу, при его кадровых перемещениях. Индивидуальные графики назначаются сотрудникам персонально.  Для заполнения графиков работы используются регламентированный государственный или региональный производственный календарь.  Предусмотрена возможность формировать графики неполного рабочего времени и графики суммированного учета времени.  Программа позволяет анализировать причины потерь рабочего времени, получать информацию о количестве работников, отсутствовавших по различным причинам за определенный период.  По результатам накопленных данных формируется «табель учета рабочего времени», на основании которого рассчитывается заработная плата сотрудника (рисунок 16).    Рисунок 16 – табель учета рабочего времени  Во время прохождения технологической практики изучил, что для учета обмена бизнес документами и иными видами документов используется программа «Инфо-Бухгалтер».  Электронный документооборот – способ организации работы с документами, при котором основная масса документов используется в электронном виде и хранится централизованно.  Система электронного документооборота – компьютерная программа, которая позволяет организовать работу с электронными документами, а так же взаимодействие между сотрудниками (передача документов, выдачу заданий, отправку уведомлений и т.п.).  Так же систему электронного документооборота называют EBMS – система управления электронными документами.  Программы по документообороту выполняют задачи учета документа, взаимодействия сотрудника, контроля и анализа исполнительской дисциплины.  Основные направления данного решения – общий документооборот, управление договорной деятельностью, электронный архив и работа с обращениями.  Так же она позволяет автоматизировать полный цикл внутренних документов, коммерческих предприятий и бюджетных учреждений с поддержкой как электронного, так и бумажного документооборота; а так же обеспечить в организации централизованное безопасное хранение, оперативный доступ, поиск и эффективную работу с официальными и внутренними документами любых типов. Такая программа поддерживает многопользовательскую работу в локальной сети или через интернет с использованием веб-браузеров или тонкого клиента  Данная программа предоставляет следующие возможности:   * Быстрый поиск необходимого документа; * Организация документооборота на предприятии; * Осуществление наблюдения за электронным документом; * Надежное хранение документа; * Обеспечение только санкционного доступа к документам; * Ведение календаря заданий; * Возможность контроля работы сотрудников; * Регистрация и хранение документов; * Хранение и поиска документа в базе данных; * Контроль выполнения задач; * Безопасное хранение документов.   Все эти функции позволяющие максимально автоматизировать делопроизводство, реализуются в системе документооборота, разработанной специально с учетом потребностей компании разного масштаба.  Программа по работе с документооборотом позволяет работать не только с текстовыми файлами, появившимися в течении рабочего дня, но и со следующими типами файлов: изображение, файлами аудио и видео форматов, архивами и файловыми системами.  Система поддерживает работу с бизнес процессами, при этом бизнес процессы могут быть настроены пользователями с учетом различных параметров и опций.  Данная программа также имеет возможность создавать и использовать разнообразные шаблоны, в том числе шаблоны документов, вести учет и организовывать доступ к архивным документам, использовать в полном объеме возможности платформы, в том числе работать в режиме управляемых форм в веб-клиенте.  *Протоколирование антивирусной профилактики*  Во время прохождения технологической практики ознакомился с антивирусными программами и принципами их работы.  Антивирусная программа – это специализированная программа для обнаружения компьютерных вирусов, а также нежелательных вредоносных программ и восстановления зараженных такими программами файлов, а также для профилактики – предотвращения заражения файлов или операционной системы вредоносным кодом.  Для защиты от вирусов используют три группы методов:  Методы, основанные на анализе содержимого файла (как файлов данных, так и файлов с кодами команд). К этой группе относятся сканирования сигнатур вирусов, а также проверка целостности и сканирование подозрительных команд.  Методы, основанные на отслеживании поведения программ при их выполнении. Эти методы заключаются в протоколировании всех событий, угрожающих безопасности системы и происходящих либо при реальном выполнении кода, либо при его программной эмуляции.  Методы регламентации порядка работы с файлами и программами. Эти методы относятся к административным мерам обеспечения безопасности.  Метод сканирования сигнатур (сигнатурный анализ, сигнатурный метод) основан на поиске в файлах уникальной последовательности байтов - сигнатуры, характерной для определенного вируса. Для каждого вновь обнаруженного вируса специалистами антивирусной лаборатории выполняется анализ кода, на основании которого определяется его сигнатура. Полученный кодовый фрагмент помещают в специальную базу данных вирусных сигнатур, с которой работает антивирусная программа. Достоинством данного метода является относительно низкая доля ложных срабатываний, а главным недостатком - принципиальная невозможность обнаружения в системе нового вируса, для которого отсутствует сигнатура в базе данных антивирусной программы, поэтому требуется своевременная актуализация базы данных сигнатур.  Метод сканирования подозрительных команд ([эвристическое сканирование](https://ru.wikipedia.org/wiki/%D0%AD%D0%B2%D1%80%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D1%81%D0%BA%D0%B0%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), эвристический метод) основан на выявлении в сканируемом файле некоторого числа подозрительных команд и(или) признаков подозрительных кодовых последовательностей (например, команда форматирования жесткого диска или функция внедрения в выполняющийся процесс или исполняемый код). После этого делается предположение о вредоносной сущности файла и предпринимаются дополнительные действия по его проверке. Этот метод обладает хорошим быстродействием, но довольно часто он не способен выявлять новые вирусы.  Основные задачи антивирусных программ:   * сканирование файлов и программ в режиме реального времени; * сканирование компьютера по требованию; * сканирование Интернет-трафика; * сканирование электронной почты; * защита от атак враждебных веб-узлов; * восстановление поврежденных файлов.   Требования, выполняемы антивирусной программой должны быть следующей:  Стабильность и надежность работы. Этот параметр, без сомнения, является определяющим -- даже самый лучший антивирус окажется совершенно бесполезным, если он не сможет нормально функционировать на вашем компьютере, если в результате какого-либо сбоя в работе программы процесс проверки компьютера не пройдет до конца. Тогда всегда есть вероятность того, что какие-то зараженные файлы остались незамеченными.  Размеры вирусной базы программы (количество вирусов, которые правильно определяются программой). С учетом постоянного появления новых вирусов база данных должна регулярно обновляться -- что толку от программы, не видящей половину новых вирусов и, как следствие, создающей ошибочное ощущение “чистоты” компьютера. Сюда же следует отнести и возможность программы определять разнообразные типы вирусов, и умение работать с файлами различных типов (архивы, документы). Немаловажным также является наличие резидентного монитора, осуществляющего проверку всех новых файлов “на лету” (то есть автоматически, по мере их записи на диск).  Скорость работы программы, наличие дополнительных возможностей типа алгоритмов определения даже неизвестных программе вирусов (эвристическое сканирование). Сюда же следует отнести возможность восстанавливать зараженные файлы, не стирая их с жесткого диска, а только удалив из них вирусы. Немаловажным является также процент ложных срабатываний программы (ошибочное определение вируса в “чистом” файле).  Многоплатформенность (наличие версий программы под различные операционные системы). Конечно, если антивирус используется только дома, на одном компьютере, то этот параметр не имеет большого значения. Но вот антивирус для крупной организации просто обязан поддерживать все распространенные операционные системы. Кроме того, при работе в сети немаловажным является наличие серверных функций, предназначенных для административной работы, а также возможность работы с различными видами серверов.  В ОДО «ОНИКС» используется антивирусное ПО Kaspersky IS (рисунок 17).  http://cdn.comss.net/img/102016/2016-10-07_1738.png  Рисунок 17 - Главное окно антивирусной программы «Kaspersky IS»  Данная антивирусная программа представляет следующие возможности пользователям:   * Защита в режиме реального времени. Касперский может обнаружить и затем обезвредить вирусы и различные шпионские программы до того, как они будут активированы при запуске операционной системы. * Эвристический анализ. Антивирус анализирует систему на наличие скрытых/замаскированных утилит или неизвестных вирусов. * Обновление. Касперский уведомит пользователя о том, что появилась [новая версия антивируса](http://service-nk.ru/usefultips/avast_installation.html) или свежая антивирусная база. * Блокирование сайтов. Антивирус, обнаружив вирусную активность на сайте, выдаст соответствующее уведомление или попытается его заблокировать. * Анализ установленного программного обеспечения. При [установке](http://service-nk.ru/usefultips/comodo-firewall-antivirus.html) на компьютер программ Kaspersky IS анализирует их, а после завершения работы приложений составляет отчет, на основе которого делает вывод о том, что с ними дальше делать, если их поведение подозрительно. * Проверка файлов. Касперский проверяет файлы на надежность еще до того, как пользователь их откроет. Делается это исходя из популярности файлов (оценивается по количеству людей, обладающих такими же файлами) и уровня их новизны. * Анализ сайтов по их репутации. Антивирус анализирует ресурсы Интернета по их репутации и рейтингу, опираясь на отзывы пользователей, что дает возможность составить «черный» и «белый» список сайтов. Это обезопасит Вас от плохих сайтов. * Портал управления. Последняя версия Касперского имеет встроенный каталог под названием «Помощь пользователю», который дает пользователю возможность смотреть инструкции по настройке и использованию антивируса. Также в портале возможно отслеживать состояние антивирусной программы (новые возможности, статистику и прочее). Чтобы иметь возможность пользоваться данным порталом, требуется соединение с Интернетом. * Сканирование файловых систем. * Проверка почты. * Перехват сценариев. * Запуск программы в виртуальной среде. Если Касперский решит, что запускаемая утилита имеет подозрительную активность, то пользователю будет предложено три варианта дальнейших действий: заблокировать, запустить приложение в [AutoSandbox (виртуальная среда)](http://service-nk.ru/usefultips/avast-sandbox.html) или запустить за пределами песочницы. * Проверка дисков. Как и прочие [антивирусные программы](http://service-nk.ru/usefultips/avg-antivirus-free.html), Касперский может проверить любой накопитель информации (переносной или встроенный жесткий диск, карты памяти, флешки, прочие устройства для хранения данных). * Автообработка. Касперский при обнаружении вирусов удаляет их автоматически, не спрашивая дальнейших действий у пользователя. * Дистанционная помощь. Если возникнут проблемы с антивирусом, можно связаться со службой поддержки.   *Составление тестов для проверки функционирования производственных задач*  Во время прохождения технологической практики в организации составлял чек-листы для проверки функционирования сайта организации.  Чек-лист – это список, содержащий ряд необходимых проверок во время тестирования программного продукта. Отмечая пункты списка, команда или один тестировщик, могут узнать о текущем состоянии выполненной работы и о качестве продукта. Работая над проектом по чек-листу, исключена вероятность повторной проверки по тем же кейсам, а также повышается качество тестирования, так как вероятность оставить без внимания какой-то функционал существенно снижается. Поэтому очень важно знать из каких элементов состоит чек-лист и уметь им эффективно пользоваться.  При создании чек-листов использовал такие критерии как:   * функционал – проверка различных функциональных возможностей сайта; * дизайн – дизайн сайта должен отображаться без искажений на всех устройствах; * контент – контент должен быть тщательно перепроверен на сайте; * дополнительные функции – то, что не должно бросаться в глаза рядовому пользователю; * главное меню – главное меню сайта должно быть на своем месте, быть на всех страницах и корректно работать; * поиск – должен быть реализован качественно и охватывать весь сайт.   Для создания чек-листов использовал программа обработки электронных таблиц Excel, рисунок 00, 00, 00:    Рисунок 18 - Чек-лист для проверки функционала и дизайна сайта    Рисунок 19 - Чек-лист для проверки контента и дополнительных функций сайта    Рисунок 20 - Чек-лист для проверки главного меню и поиска сайта  **Разработка программных средств индивидуального задания**  *Анализ предметной области индивидуального задания. Разработка технического задания на программное средство.*  Индивидуальным заданием по технологические практики является разработка программного продукта на тему «Архивация работ учащихся».  Проведем анализ предметной области индивидуального задания выданного 11.11.2020 года. Данный проект предназначен для внедрения и использования на с возможным расширением в сторону общедоступного ресурса файлов с подписками. Назначение проекта – предоставление полного доступа пользователя к постоянно расширяемой пользователями базе данных уже готовых работ их учебного учреждения.  Программный продукт состоит из 7 модулей: главная страница, профиль, запись, FAQ, регистрация, вход, поиск.  Главная страница представляет собой окно в браузере, которое предоставляет обширную часть функционала проекта.  *Построение моделей, программные системы*  Данная программа представлена в виде структурной схемы (рисунок 21):    Рисунок 21 - Структурная схема программы  Модели в проекте представлены тремя сущностями: пользователь и его профиль, запись, категория.  Django ORM (Object Relational Mapping) является одной из самых мощных особенностей Django. Это позволяет нам взаимодействовать с базой данных, используя код Python, а не SQL.  Благодаря этой особенности, сущности в проекте можно описать языком Python.  from datetime import datetime  from django.contrib.auth.models import User, AnonymousUser  from django.db.models.signals import post\_save  from django.dispatch import receiver  from django.urls import reverse  from django.db import models  class Profile(models.Model):  user = models.OneToOneField(User, on\_delete=models.CASCADE)  reputation = models.IntegerField(default=0)  uploads = models.IntegerField(default=0)  userpic = models.ImageField(upload\_to="userpics", default="default.png")  def \_\_str\_\_(self):  return self.user.username  @receiver(post\_save, sender=User)  def create\_user\_profile(sender, instance, created, \*\*kwargs):  if created:  Profile.objects.create(user=instance)  @receiver(post\_save, sender=User)  def save\_user\_profile(sender, instance, \*\*kwargs):  instance.profile.save()  class Category(models.Model):  name = models.CharField(max\_length=50, blank=False, unique=True)  downloads = models.IntegerField(default=0)  def \_\_str\_\_(self):  return f"{self.name}"  def get\_absolute\_url(self):  return reverse('category\_labs', args=[self.id])  class Lab(models.Model):  name = models.CharField(max\_length=50, blank=False)  course = models.IntegerField()  variant = models.IntegerField(default=0)  year = models.IntegerField(default=0)  category = models.ForeignKey(Category, on\_delete=models.CASCADE)  author = models.ForeignKey(User, on\_delete=models.CASCADE)  views = models.IntegerField(default=0)  upload\_time = models.DateTimeField(auto\_now\_add=True)  file = models.FileField(upload\_to="files")  rating = models.IntegerField(default=0)  def get\_absolute\_url(self):  return reverse('people.views.details', args=[str(self.id)])  def \_\_str\_\_(self):  return f"<{self.author.username}> {self.name}"  class Meta:  ordering = ["-id"]  *Кодирование, тестирование программного средства*  Кодировка – система, где с каждым символом поддерживаемого набора символов, сопоставляется значение, представляющее этот символ.  Символы – абстрактные сущности, которые могут быть представлены различными способами.  Компьютерная кодировка – это система, где с каждым символом сопоставлено числовое значение, представляющее этот символ. Кодировка состоит из двух компонентов:   * Кодировщик, преобразующий последовательность символов в последовательность числовых значений (байтах); * Декодер, преобразующий последовательность в байтах последовательность символов.   Кодировка описывает правила, по которым работают кодировщик и декодер. Ниже приведены основные модули кода программного продукта.  Пути к источнику.  from django.contrib import admin  from django.urls import path, re\_path  from LabArk import settings  from LabArkApp import views as LA  from django.conf.urls.static import static  from django.contrib.staticfiles.urls import staticfiles\_urlpatterns  urlpatterns = [  re\_path('^admin/?', admin.site.urls),  re\_path(r'^faq', LA.faq, name="faq"),  re\_path(r'^popular', LA.popular, name="popular"),  re\_path(r'^libs', LA.libs, name="libs"),  re\_path(r'^category/(?P<pk>\d+?)', LA.category\_labs, name='category\_labs'),  re\_path(r'^archive', LA.archive, name="archive"),  re\_path(r'^links', LA.links, name="links"),  re\_path(r'^register', LA.register, name="register"),  re\_path(r'^login', LA.login\_view, name="login"),  re\_path(r'^logout', LA.logout\_view, name="logout"),  re\_path(r'^details/(?P<pk>\d+?)', LA.details, name="details"),  re\_path(r'^user/(?P<pk>\d+?)', LA.get\_profile, name="profile\_view"),  re\_path(r'^labs/add$', LA.add\_lab, name="lab\_add"),  re\_path(r'^category/add/?$', LA.add\_category, name="category\_add"),  re\_path(r'^random\_lab$', LA.get\_random\_lab, name="random\_lab"),  path('', LA.home, name="homepage"),  ]  urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)  urlpatterns += staticfiles\_urlpatterns()  Примеры действующих механизмов страниц:  from django.contrib.auth import authenticate, logout, login  from django.contrib.auth.decorators import login\_required  from django.contrib.auth.models import User  from django.shortcuts import render, get\_object\_or\_404  from django.http import HttpResponse, HttpResponseRedirect  from LabArkApp.models import Lab, Category  from . import forms  from random import randint  from django.urls import reverse  def home(request):  return render(request, 'index.html', context={  'last\_labs': Lab.objects.all(),  'form': forms.AddCategoryForm()  })  def faq(request):  return render(request, 'faq.html')  def popular(request):  return render(request, 'index.html', context={  'last\_labs': Lab.objects.all().order\_by('-views', 'name'),  'form': forms.AddCategoryForm()  })  def category\_labs(request, pk):  return render(request, 'index.html', context={  'last\_labs': Lab.objects.all().filter(category\_\_pk=pk),  'form': forms.AddCategoryForm()  })  def archive(request):  return render(request, 'archive.html')  def libs(request):  return render(request, 'libs.html')  def links(request):  return render(request, 'links.html')  def login\_view(request):  if request.method == "GET":  return render(request, 'login.html')  else:  username = request.POST['username']  password = request.POST['password']  user = authenticate(username=username, password=password)  if user is not None:  if user.is\_active:  login(request, user)  return HttpResponseRedirect("/")  else:  pass  else:  pass  def register(request):  if request.method == "GET":  return render(request, 'register.html', context={  'form': forms.RegisterUser()  })  else:  form = forms.RegisterUser(request.POST)  if form.is\_valid():  user = User.objects.create\_user(\*\*form.cleaned\_data)  user.save()  if request.POST.get("SetLogin") == "1":  user = authenticate(username=form.cleaned\_data['username'], password=form.cleaned\_data['password'])  if user is not None:  if user.is\_active:  login(request, user)  return HttpResponseRedirect("/")  else:  pass  return HttpResponseRedirect("/")  @login\_required  def logout\_view(request):  logout(request)  return HttpResponseRedirect("/")  def details(request, pk):  lab = get\_object\_or\_404(Lab, id=pk)  lab.views += 1  lab.save()  return render(request, "lab\_view.html", context={  "lab\_object": lab,  })  def get\_profile(request, pk):  user = get\_object\_or\_404(User, id=pk)  return render(request, "profile.html", context={  'user': user,  'last\_labs': Lab.objects.all().filter(author\_\_pk=pk)  })  def add\_lab(request):  if request.method == "POST":  form = forms.UploadLabForm(request.POST)  if form.is\_valid():  if request.user.is\_authenticated:  author = request.user  else:  author = User.objects.get\_by\_natural\_key('anonymous')  lab = Lab(\*\*form.cleaned\_data)  lab.author = author  lab.category = Category.objects.get(pk=request.POST.get("category"))  lab.file = request.FILES['file']  lab.save()  author.profile.reputation += 10  author.profile.uploads += 1  author.save()  return HttpResponseRedirect(reverse('details', args=[lab.pk]))  else:  return HttpResponse(form.errors)  else:  return render(request, "add\_lab.html", context={  'form': forms.UploadLabForm(),  'category': Category.objects.all()  })  def add\_category(request):  if request.method == "POST":  category = Category.objects.create(name=request.POST.get("name"))  category.save()  return HttpResponseRedirect("/")  else:  return HttpResponseRedirect('/')  def get\_random\_lab(request):  return HttpResponseRedirect(reverse('details', args=[randint(1, Lab.objects.count())]))  На практике выше приведенный код использует шаблоны HTML в связке с CSS, JS и отдает их пользователю через механизмы Django. Результаты работы приведен на рисунках 22, 23.    Рисунок 22 – Главная страница сайта    Рисунок 23 – Страница просмотра профиля пользователя  При тестировании программ и существующих методов испытаний был выбран метод «белого» и «черного» ящика, так как они являются более удобными.  При проверке правильности отработки и функции программы использовался метод черного ящика, если во время откладки программы ожидаемый результат был не верным, для локализации ошибки использовался метод белого ящика.  При тестировании методом черного ящика рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура.  При тестировании методом белого ящика объектом тестирования является не внешнее, а внутреннее поведение программы. Проверяется корректность построения всех элементов программы и правильность их взаимодействия друг с другом.  *Оценка возможности интеграции разработанного программного средства, комплекс программ организации*  Цель интеграции данных – объединение всех корпоративных данных, размещенных в разных источниках.  Учитывая специфику сайта, можно разделить на два подхода возможности интеграции в другие системы:   1. Внедрение продукта – как частный архив для конкретных задач системы 2. Независимое расширение сайта с использованием данных системы, стоящей над ней   Интеграция программного продукта существующей программой предприятия упрощает работу учётом записей на приём.  При интеграции программы получим следующее:   * Увеличенную скорость работы с гражданами; * Упрощение работы с приёмом граждан; * Ведение учёта посещаемости, сотрудников, граждан;   Внедрение программного продукта с программой предприятия не всегда под силу рядовому программисту – данный процесс требует достаточных знаний по специфике работы программы используемой на предприятии и опыта в интеграции программ. На данном предприятии существует специалисты организации, которые помогут обеспечить выполнение работ по установке и внедрению модуля для связи программы предприятия с разработанным программным продуктам.  Окончательное тестирование поможет выявить все дефекты и устранить их до запуска проекта.  *Опытная эксплуатация, анализ качества выполненного программного средства*  Качество программного продукта – способность программного продукта при заданных условиях предполагать установленным или заданным потребностям.  Характеристики качества ПО:   * Соответствие требованиям предполагает, что требования должны быть настолько четко определены, что не могут быть приняты и интерпретированы некорректно. Любые несоответствия должны рассматриваться как дефекты или отсутствие качества. * Пригодность к использованию принимают ко вниманию требования и ожидание конечных пользователей продукта, т.е. или предоставляемый сервис будет удобным для их нужд. Однако разные пользователи могут использовать продукт по- разному, а это означает, что продукт должен обладать максимально разнообразными вариантами использования. * Функциональность – определяется способностью ПО решать задачи, которые соответствуют зафиксированным и предполагаемым потребностям пользователя при заданным условиям использования ПО. Т.е. эта характеристика отвечает за то, что ПО работает исправно и точно, функционально совместима, соответствует стандартам отрасли и защищено от несанкционированного доступа; * Удобство использования – возможность легкого понимания, изучения, использования и привлекательности ПО для пользователя; * Эффективность – способность ПО использовать требуемый уровень производительности в соответствии с ресурсами, временем и другими обозначенными условиями; * Надежность – способность ПО выполнять требуемые задачи в обозначенных условиях на протяжении заданного промежутка времени или указанное количество операций. Атрибуты данной характеристики – это завершенность и целостность всей системы, способность самостоятельно и корректно восстанавливаться после сбоев в работе, отказоустойчивость.   Исходя из вышеизложенного разработанный программный продукт по теме индивидуального задания «Электронная база данных для записи на приём» можно сказать следующее:  В связи с тем, что на этапы разработки было отведено небольшое количество времени, разработанные требования имели определенные недоработки. В связи с этим программный продукт можно утверждать имеет определенные дефекты.  Программный продукт полностью отвечает конечным требованиям пользователей. Продукт удобен для нужд администраторов и пользователей. Программа не обладает разнообразными вариантами использования.  Функциональность программного продукта отвечает зафиксированным и предполагаемым потребностям пользователя, при заданных условиях использования. Программный продукт соответствует стандартам отрасли.  Программный продукт удобен в использовании, легок в понимании, изучении и использовании.  Программный продукт способен использовать ограниченное количество ресурсов и времени, в соответствии с системными характеристиками компьютера и заданными параметрами.  Программный продукт не способен восстановиться после сбоя, но имеет хорошую отказоустойчивость.  Опытная эксплуатация помогает решить ошибки, связанные с различными проблемами программы.  Пример ошибок, допущенных при разработке программного продукта (рисунок 30):  https://i.stack.imgur.com/BPobU.png  Рисунок 30 – Ошибка, вызванная отсутствием обработки со стороны сервера данного пути  При неполном вводе обязательных к заполнению данных при добавлении появится сообщение о необходимости заполнении пустого поля ввода (рисунок 31):    Рисунок 31 – Сообщение о необходимости заполнения полей ввода  *Расчет стоимости программного средства*  Определение экономического эффекта от внедрения программного продукта включает два этапа:  1. Определение отпускной цены программного продукта  2.Определение экономического эффекта от внедрения программного продукта и срока окупаемости вложений.  Для разработки программного продукта привлекается техник-программист из сторонней организации. Оклад техника-программиста с учетом повышений – 395 р. Среднемесячное количество рабочих часов по норме в расчетном году – 170 ч.  Определение отпускной цены программного продукта  На данном этапе определяют трудоемкость разработки программного продукта, стоимость машинного времени, отпускной цены программного продукта в условиях действующей системы налогообложения.  Определение трудоемкости разработки программного продукта производится на основании опытно-статистического метода, в целом на всю разработку, исходя из того, что ее выполняет один техник-программист.  Условное число команд принимается = 1400;  Программные продукты по степени новизны могут быть отнесены к группе Б.  По степени сложности программные продукты могут быть отнесены к 1 группе сложности.  Коэффициент, учитывающий группу сложности и степень новизны 1,08  Определим условное число команд    Определяем время, затраченное на каждый этап создания программного продукта:   1. Tпо (время на подготовку описания задачи), берется по факту и составляет:   Tпо =  Коэффициент учета изменений задачи - 1,2.  Коэффициент, учитывающий квалификацию программиста = 1.  Подсчитаем время на описание задачи:  Tо =  Определим время на разработку алгоритма.  Tбс =  Подсчитываем время написания программы на языке программирования:  Tн =  Общее время на создание программного продукта составит:  T = 8 + 36,29+ 30,24 + 45,36 = 119,89 чел.- час  Структура общего времени и трудоемкость разработки программного продукта представлены в таблице 1.  Таблица 1 – Структура общего времени и трудоемкость разработки программного продукта   |  |  |  |  | | --- | --- | --- | --- | | Вид работ | Процент от трудоемкости работ | Трудоемкость в часах | | | Всего | *Время машинное* | | Техническое задание, эскизный проект (время на подготовку описания задачи, время на описание задачи) | 33,33 | 8+36,29=44,29 | – | | Технический проект (время на разработку алгоритма, время на разработку блок-схемы, время написания программы на языке программирования) | 56,89 | 30,24+45,36=75,6 | – | | Рабочий проект (машинное время), (время набивки программы, время отладки и тестирования программы) | 6,02 | 8 | 8 | | Внедрение в том числе:   * машинное время * немашинное время | 3,76  3,01  0,75 | 5  4  1 | 4  4  – | | Итого: | 100 | 132,89 | 12 |   Примечание: рабочий проект – машинное время является перекрываемым, т. е. учитывается и в немашинном времени для расчета оплаты труда техника-программиста.  Процент от трудоемкости работ определяют отношением трудоемкости по каждому виду работ к общей трудоемкости, выраженное в процентах.  Определение стоимости машинного времени.  а) Определение стоимости потребляемой электроэнергии, руб.:  (1)  где – МК мощность, потребляемая компьютером;  FK – полезный фонд времени работы компьютера за год;  МП – мощность, потребляемая принтером;  FП – полезный фонд времени работы принтера за год;  С – стоимость 1кВт / час;  Квр – коэффициент, учитывающий использование по времени (Квр=0,8);  Кс – коэффициент, учитывающий потери в сети (Кс=1,05);  Сэл = (0,3\*1500 + 0,1\*80)\*0,30287\*0,8\*1,05 = 116,52 р.  б) Определение суммы амортизационных отчислений, руб.:  (2)  где БК - Балансовая стоимость компьютера;  БП - Балансовая стоимость принтера;  Нао - Норма амортизации для оборудования.  Сао = (500+150)\*0,2 = 130 р.  Стоимость машинного времени составит за год: 116,52+130= 246,52  на 1 машино-час: 246,52 / 1500 = 0,16  д) Расчет стоимости машинного времени, рублей:  , (3)  где Тм – машинное время работы компьютера в расчете на программу.  Смвр = 132,89 \* 0,16 = 21,26 р.  Определение отпускной цены программного продукта в условиях действующей системы налогообложения.  a) Затраты на материалы и покупные комплектующие изделия.  Потребуется 1 гибкий CD-диск стоимостью 1,78 р. Тонер стоимостью 17 . Бумага белая, А4 (210х297 мм), 80г/м2 – цена 8 р. Всего затраты на материалы составят 26,78 р.  Часовая тарифная ставка определяется путем деления месячной тарифной ставки на среднемесячное количество рабочих часов и результат умножается на тарифный коэффициент.  Прямая заработная плата определяется путем умножения часовой тарифной ставки на трудоемкость по соответствующему виду работ.  Произведем расчеты:   * часовая тарифная ставка за выполнение технического задания эскизного проекта:   395/170=2,32 р.  По техническому и рабочему проекту расчеты аналогичны.   * прямая заработная плата за выполнение технического задания эскизного проекта:   2,32\*44,29=102,75р.   * прямая заработная плата за выполнение технического проекта:   2,32\*75,6=175,39 р.   * прямая заработная плата за выполнение рабочего проекта:   2,32\*8=18,56 р.   * прямая заработная плата за внедрение рабочего проекта:   2,32\*5=11,6 р.   * итого: 102,75+175,39+18,56+11,6=308,3 р.  1. Определим материальные затраты:   21,26 +26,78=48,04р.   1. Дополнительная заработная плата составит:   308,3\*10/100=30,83 р.   1. Расходы на оплату труда:   308,3+30,83=339,13 р.   1. Отчисления от заработной платы:   339,13\*34,6/100=117,34 р.   1. Прочие затраты:   308,3\*1/100=3,08 р.   1. Общехозяйственные расходы:   308,3\*50/100=154,15 р.   1. Полная себестоимость:   48,04+339,13+117,34+3,08+154,15=661,74 р.   1. Планируемая прибыль:   661,74\*20/100=132,35 р.   1. Цена разработчика:   661,74+132,35=794,09 р.   1. Налог на добавленную стоимость:   794,09\*20/100=158,82 р.   1. Стоимость программного продукта с НДС:   794,09+158,82=952,91 р.  Срок окупаемости затрат, вложенных в программный продукт определяется по формуле:  ТВ = , (4)  где ПУ – прирост условной прибыли, получаемой в результате автоматизации, р.;  И – размер инвестиций (капитальных вложений), необходимых для автоматизации, р.  ТВ = 661,74/952,91=0,7 года.  Срок окупаемости разработанного программного продукта составил 0,7 года. Следовательно, внедрение программного продукта не нанесет убытков организации и является более эффективным решением вложения средств.  Экономический эффект от улучшения использования компьютеров  (5)  *Ср*– стоимость выполнения функций без компьютера, руб.;  Ср - количество часов, которые необходимы на выполнение работы без компьютера.  *Св –* стоимость одного часа использования компьютера, руб.;  *Тв –* время выполнения функции на компьютере, ч.  *Разработка разделов отчета по практике*  Процесс организации и проведения технологической практики состоит из четырех этапов:   * вводный; * теоретический; * практический; * заключительный.   Вводный этап подразумевает ознакомление с государственным учреждением Администрация Советского района, изучения основной деятельности организации, ознакомление с рабочим планом и персоналом.  На теоретическом этапе идет изучение предстоящих задач, ход и план их дальнейшего выполнения, изучение области применения знаний и навыков, полученных на предприятии.  Практический этап включает в себя применение всех полученных теоретических знаний, совершенствование собственных навыков, выполнение поставленных задач, заполнение дневника-отчета.  Заключительный этап предполагает окончательное оформление дневника-отчета и приложений к нему, закрепление знаний, полученных в ходе теоретических и практических работ.  *Подготовка презентации*  Разработана презентация по итогам прохождения практики:    Рисунок 32 – Слайд 1    Рисунок 33 – Слайд 2    Рисунок 34 – Слайд 3    Рисунок 35 – Слайд 4    Рисунок 37 – Слайд 5    Рисунок 38– Слайд 6    Рисунок 39– Слайд 7    Рисунок 40– Слайд 8  **Обобщение материалов по практике**  *Систематизация материалов по практике*  Во время прохождения технологической практики, были выполнены все поставленные цели и задачи, что положительно сказалось на практических и теоретических знаниях профессиональной деятельности по итогам обучения в колледже. Был разработан программный продукт Web-browser 1.00, который решает следующие задачи:   * Работа с сайтами; * Открытие сохраненных сайтов в формате html; * Просмотр кода страницы; * Ввод HTML-кода в HTML редактор.   По мере прохождения практики сложились хорошие отношение с коллективом, повысились профессиональные навыки, расширился кругозор, хорошо сказалось на качестве получаемого практического и теоретического опыта.  *Оформление отчета по практике*  Оформление отчета по практике соответствует предъявляемым нормам. Данные для дневника отчета и приложения были взяты непосредственно на предприятии. Темы для каждодневного заполнения отчета соответствует учебной программе практики и календарному графику прохождения практики.  **Подведение итогов практики**  Вовремя прохождение технологической практики на предприятии были получены знания по эксплуатации таких программ как:   * Инфо-Бухгалтерия; * Электронная база данных для регистрации на приём; * Kaspersky IS; * 1С бухгалтерия; * Opera Mail.   Так же был обучен более грамотному составлению отчета по результатам работы. И усовершенствовал знания во время прохождения практики. |  |

**Выводы и заключение**

В результате прохождения практики цель была достигнута, все задачи решены в полном объёме, профессиональные компетенции приобретены.

Главной проблемой, которая была характерна для ОДО «ОНИКС», являлась отсутствие схем сети и старые интернет кабеля. Наиболее оптимальным решением было составление L1, L2, L3 схем сети, докупка коммутаторов и замена старых проводов.

В процессе прохождения практики были составлены L1, L2, L3 схемы, заменены коммутаторы и проведены новые интернет кабеля, что позволило увеличить скорость документооборота в ОДО «ОНИКС», а также убрать сбои сети.

Знания, умения, навыки, полученные за период практики, вились отличным стимулом для активной работы в освоении будущей специальности, позволили практически реализовать теоретически изученные моменты, получить первый профессиональный опыт работы и сформировать общее представление о специфике деятельности ОДО «ОНИКС».

**5 января 2020 г.**

**Практикант** Шевелев И.В.

**Заключение руководителя практики от организации по дневнику-отчёту**

Учащийся Шевелев Илья Владимирович учреждения образования «Гомельский торгово-экономический колледж» Белкоопсоюза во время прохождения технологической практики в ОДО «ОНИКС» ежедневно предоставлял отчет по проделанной работе, в котором производил записи об изученном материале и проделанной работе. Дневник-отчет составлен грамотно и отражает содержание работы.

Все задания дневника-отчета полностью соответствуют календарному графику.

Представлены выводы и предложения, в которых проанализированы положительные моменты и недостатки, внесены предложения по улучшению работы организации в рассматриваемой области.

Дневник-отчет заслуживает оценки.

**Руководитель практики от организации**  Щеников Д.В.

**Печать**

**ХАРАКТЕРИСТИКА**

В течении прохождения технологической практики, студент Шевелев Илья зарекомендовал себя с хорошей стороны. Он проявил себя, как ответственный, творческий, неординарно мыслящий работник. Все поставленные задачи выполнялись в заданные сроки. За студентом был закреплён ряд задач, с которым он смог справится, при исполнении заданий действовал с профессиональной точки зрения.

**Руководитель практики** Щеников Д.В.

**Печать**

**ПРИЛОЖЕНИЕ 1**

**ИНСТРУКЦИЯ**

**по охране труда при работе на персональном компьютере**

**1. Общие требования охраны труда**  
1.1 К работе на персональном компьютере (ПК) допускаются лица, прошедшие медицинское освидетельствование, вводный инструктаж, первичный инструктаж, обучение и стажировку на рабочем месте, проверку знаний требований охраны труда, имеющие группу I по электробезопасности.  
1.2 При работе на персональном компьютере работник обязан:  
1.2.1 Выполнять только ту работу, которая определена его должностной (рабочей) инструкцией.  
1.2.2 Выполнять правила внутреннего трудового распорядка.  
1.2.3 Соблюдать режим труда и отдыха в зависимости от продолжительности, вида и категории трудовой деятельности (приложение № 1).  
1.2.4 Правильно применять средства индивидуальной и коллективной защиты.  
1.2.5 Соблюдать требования охраны труда.  
1.2.6 Немедленно извещать своего непосредственного или вышестоящего руководителя о любой ситуации, угрожающей жизни и здоровью людей, о каждом несчастном случае, происшедшем на производстве, или об ухудшении состояния своего здоровья, в том числе о проявлении признаков острого профессионального заболевания (отравления).  
1.2.7 Проходить обучение безопасным методам и приемам выполнения работ и оказанию первой помощи пострадавшим на производстве, инструктаж по охране труда, проверку знаний требований охраны труда.  
1.2.8 Проходить обязательные периодические (в течение трудовой деятельности) медицинские осмотры (обследования), а также проходить внеочередные медицинские осмотры (обследования) по направлению работодателя в случаях, предусмотренных Трудовым кодексом и иными федеральными законами.  
1.2.9 Уметь оказывать первую помощь пострадавшим от электрического тока и при других несчастных случаях.  
1.2.10 Уметь применять первичные средства пожаротушения.  
1.3 При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:  
-повышенный уровень электромагнитных излучений;  
-повышенный уровень статического электричества;  
-пониженная ионизация воздуха;  
-статические физические перегрузки;  
-перенапряжение зрительных анализаторов  
-недостаточная освещенность рабочего места.  
1.4 Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана ВДТ. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4 - 0,6 и не иметь блестящих деталей, способных создавать блики.  
1.5 Конструкция ВДТ должна предусматривать регулирование яркости и контрастности.  
1.6 Площадь на одно рабочее место пользователей ПЭВМ с ВДТ на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 м2, в помещениях культурно-развлекательных учреждений и с ВДТ на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м2.  
При использовании ПВЭМ с ВДТ на базе ЭЛТ (без вспомогательных устройств - принтер, сканер и др.), отвечающих требованиям международных стандартов безопасности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м2 на одно рабочее место пользователя (взрослого и учащегося высшего профессионального образования).  
1.7 Помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.  
1.8 Рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2м.  
1.9 Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.  
1.10 Оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.  
1.11 Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).  
1.12 Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.  
1.13 Рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:  
-высота рабочей поверхности стола должна регулироваться в пределах 680-800мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725мм;  
-рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм;  
-рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию;  
-рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300мм, глубину не менее 400мм, регулировку по высоте в пределах до 150мм и по углу наклона опорной поверхности подставки до 20о; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10мм;  
-клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.  
1.14 В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.  
1.15 Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену).  
1.16 В случаях травмирования или недомогания необходимо прекратить работу, известить об этом руководителя работ и обратиться в медицинское учреждение.  
1.17 За невыполнение данной инструкции виновные привлекаются к ответственности согласно законодательства Российской Федерации.

**2. Требования охраны труда перед началом работы**  
2.1 Подготовить рабочее место.  
2.2 Отрегулировать освещение на рабочем месте, убедиться в отсутствии бликов на экране.  
2.3 Проверить правильность подключения оборудования к электросети.  
2.4 Проверить исправность проводов питания и отсутствие оголенных участков проводов.  
2.5 Убедиться в наличии заземления системного блока, монитора и защитного экрана.  
2.6 Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.  
2.7 Проверить правильность установки стола, стула, подставки для ног, пюпитра, угла наклона экрана, положение клавиатуры, положение «мыши» на специальном коврике, при необходимости произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

**3. Требования охраны труда во время работы**  
3.1 Работнику при работе на ПК запрещается:  
-прикасаться к задней панели системного блока (процессора) при включенном питании;  
-переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;  
-допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;  
-производить самостоятельное вскрытие и ремонт оборудования;  
-работать на компьютере при снятых кожухах;  
-отключать оборудование от электросети и выдергивать электровилку, держась за шнур.  
3.2 Во время регламентированных перерывов с целью снижения нервноэмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.

**4. Требования охраны труда в аварийных ситуациях**  
4.1 Во всех случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, немедленно отключить питание и сообщить об аварийной ситуации руководителю.  
4.2 Не приступать к работе до устранения неисправностей.  
4.3 При возникновении пожара, задымлении:  
4.3.1 Немедленно сообщить по телефону «01» в пожарную охрану, оповестить работающих, поставить в известность руководителя подразделения, сообщить о возгорании на пост охраны.  
4.3.2 Открыть запасные выходы из здания, обесточить электропитание, закрыть окна и прикрыть двери.  
4.3.3 Приступить к тушению пожара первичными средствами пожаротушения, если это не сопряжено с риском для жизни.  
4.3.4 Организовать встречу пожарной команды.  
4.3.5 Покинуть здание и находиться в зоне эвакуации.  
4.4 При несчастном случае:  
4.4.1 Немедленно организовать первую помощь пострадавшему и при необходимости доставку его в медицинскую организацию.  
4.4.2 Принять неотложные меры по предотвращению развития аварийной или иной чрезвычайной ситуации и воздействия травмирующих факторов на других лиц.  
4.4.3 Сохранить до начала расследования несчастного случая обстановку, какой она была на момент происшествия, если это не угрожает жизни и здоровью других лиц и не ведет к катастрофе, аварии или возникновению иных чрезвычайных обстоятельств, а в случае невозможности ее сохранения – зафиксировать сложившуюся обстановку (составить схемы, провести другие мероприятия).

**5. Требования охраны труда по окончании работы**

5.1 Отключить питание компьютера.  
5.2 Привести в порядок рабочее место.  
5.3 Выполнить упражнения для глаз и пальцев рук на расслабление.

**ПРИЛОЖЕНИЕ 2**

**ДОЛЖНОСТНАЯ ИНСТРУКЦИЯ**

**Инженера-программиста**

1. **Общие положения**

1. Инженер-программист относится к категории специалистов.

2. На должность:

- инженера-программиста назначается лицо, имеющее высшее профессиональное (техническое или инженерно-экономическое) образование без предъявления требований к стажу работы или среднее профессиональное (техническое или инженерно-экономическое) образование и стаж работы в должности техника I категории не менее 3 лет либо других должностях, замещаемых специалистами со средним профессиональным образованием, не менее 5 лет;

- инженера-программиста III категории - лицо, имеющее высшее профессиональное (техническое или инженерно-экономическое) образование и опыт работы по специальности, приобретенный в период обучения, или стаж работы на инженерно-технических должностях без квалификационной категории;

- инженера-программиста II категории - лицо, имеющее высшее профессиональное (техническое или инженерно-экономическое) образование и стаж работы в должности инженера-программиста III категории или других инженерно-технических должностях, замещаемых специалистами с высшим профессиональным образованием, не менее 3 лет;

- инженера-программиста I категории - лицо, имеющее высшее профессиональное (техническое или инженерно-экономическое) образование и стаж работы в должности инженера-программиста II категории не менее 3 лет.

3. Назначение на должность инженера-программиста и освобождение от нее производственного отдела ВЦ (ИВЦ); начальника отдела разработок и внедрения ВЦ; руководителя иного структурного подразделения ВЦ.

4. Инженер-программист должен знать:

4.1. Руководящие и нормативные материалы, регламентирующие методы разработки алгоритмов и программ и использования вычислительной техники при обработке информации.

4.2. Основные принципы структурного программирования.

4.3. Виды программного обеспечения.

4.4. Технико-эксплуатационные характеристики, конструктивны особенности, назначение и режимы работы ЭВМ, правила ее технической эксплуатации.

4.5. Технологию автоматической обработки информации и кодирования информации.

4.6. Формализованные языки программирования

4.7. Действующие стандарты, системы счислений, шифров и кодов.

4.8. Порядок оформления технической документации.

4.9. Передовой отечественный и зарубежный опыт программирования и использования вычислительной техники.

4.10. Основы экономики, организации производства, труда и управления.

4.11. Основы трудового законодательства.

4.12. Правила внутреннего трудового распорядка.

4.13. Правила и нормы охраны труда.

5. На время отсутствия инженера-программиста (отпуск, болезнь, пр.) его обязанности исполняет лицо, назначенное в установленном порядке. Данное лицо приобретает соответствующие права и несет ответственность за качественное и своевременное исполнение возложенных на него обязанностей.

**2. Должностные обязанности**

Инженер-программист:

1. На основе анализа математических моделей и алгоритмов решения экономических и других задач разрабатывает программы, обеспечивающие возможность выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит их тестирование и отладку.

2. Разрабатывает технологию решения задачи по всем этапам обработки информации.

3. Осуществляет выбор языка программирования для описания алгоритмов и структур данных.

4. Определяет информацию, подлежащую обработке средствами вычислительной техники, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и вывода, методы ее контроля.

5. Выполняет работу по подготовке программ к отладке и проводит отладку.

6. Определяет объем и содержание данных контрольных примеров, обеспечивающих наиболее полную проверку соответствия программ их функциональному назначению.

7. Осуществляет запуск отлаженных программ и ввод исходных данных, определяемых условиями поставленных задач.

8. Проводит корректировку разработанной программы на основе анализа выходных данных.

9. Разрабатывает инструкции по работе с программами, оформляет необходимую техническую документацию.

10. Определяет возможность использования готовых программных продуктов.

11. Осуществляет сопровождение внедрения программ и программных средств.

12. Разрабатывает и внедряет системы автоматической проверки правильности программ, типовые и стандартные программные средства, составляет технологию обработки информации.

13. Выполняет работу по унификации и типизации вычислительных процессов.

14. Принимает участие в создании каталогов и картотек стандартных программ, в разработке форм документов, подлежащих машинной обработке, в проектировании программ, позволяющих расширить область применения вычислительной техники.

**3. Права**

Инженер-программист имеет право:

1. Знакомиться с проектами решений руководства предприятия, касающихся его деятельности.

2. Вносить на рассмотрение руководства предложения по совершенствованию работы, связанной с предусмотренными настоящей инструкцией обязанностями.

3. В пределах своей компетенции сообщать своему непосредственному руководителю о всех выявленных в процессе осуществления должностных обязанностей недостатках в деятельности предприятия (его структурных подразделениях) и вносить предложения по их устранению.

4. Запрашивать лично или по поручению своего непосредственного руководителя от специалистов подразделений информацию и документы, необходимые для выполнения его должностных обязанностей.

5. Привлекать специалистов всех (отдельных) структурных подразделений к решению задач, возложенных на него (если это предусмотрено положениями о структурных подразделениях, если нет - то с разрешения их руководителей).

6. Требовать от своего непосредственного руководителя, руководства предприятия оказания содействия в исполнении им своих должностных обязанностей и прав.

**3. Ответственность**

Техник-программист несет ответственность:

1. За ненадлежащее исполнение или неисполнение своих должностных обязанностей, предусмотренных настоящей должностной инструкцией, - в пределах, определенных действующим трудовым законодательством Республики Беларусь.

2. За правонарушения, совершенные в процессе осуществления своей деятельности, - в пределах, определенных действующим административным, уголовным и гражданским законодательством Республики Беларусь.

3. За причинение материального ущерба - в пределах, определенных действующим трудовым и гражданским законодательством Республики Беларусь.

**ПРИЛОЖЕНИЕ 3**

**ОРГАНИЗАЦИОННАЯ СТРУКТУРА АДМИНИСТРАЦИИ СОВЕТСКОГО РАЙОНА Г. Гомеля**

Учреждение «Редакция газеты «Советский район»

Учреждение «Центр социального обслуживания населения советского района»

КАПУП «Архитектурное бюро Советского района»

Отдел экономики

Заместитель главы администрации

Отдел идеологической работы, культуры и по делам молодежи

Отдел образования, спорта и туризма

Управление социальной защиты

Отдел жилищно-коммунального хозяйства и благоустройства

Отдел по учёту и распределению жилья

ОАО «Кирмаш»

КЖРЭУП «Советское»

Отдел по работе с обращениями граждан и юридических лиц

ЗАГС

Обслуживающий персонал

Группа бухгалтерского учёта и отчётности

Управляющий делами администрации

Заместитель главы администрации

Первый заместитель главы администрации

ОВД

Отдел организационно-кадровой работы

**ГЛАВА АДМИНИСТРАЦИИ**