**РЕФЕРАТ**

Разработан дипломный проект по теме: Технология обучения на основе использования образовательных сервисов Web 2.0 в ходе изучения темы учебного предмета «Информатика» на примере создания web-сайта «Баскетбол Беларуси в лицах»

Дипломный проект состоит из расчетно-пояснительной записки на 91 странице, 2 чертежей, 3 приложений, 5 плакатов.

Предмет исследования данного проекта – web-сайт «Баскетбол Беларуси в лицах» .

Целью дипломного проекта является изучение тем учебного предмета «Информатика» на примере создания web-сайта «Баскетбол Беларуси в лицах».

Задачи дипломного проекта:

* изучить темы учебного предмета «Информатика» на примере создания web-сайта «Баскетбол Беларуси в лицах»;
* реализовать механизм добавления, удаления и изменения данных;
* реализовать механизмы поиска, сортировки и вывода данных;
* разработать интуитивно понятный интерфейс;
* рассчитать оценку конкурентоспособности объекта проектирования;
* рассчитать технико-экономические показатели разработки программного обеспечения;
* проанализировать и охарактеризовать условия труда работников высшего образования.

Методы исследования: анализ литературы по исследуемой теме, а также методы изучения, сравнения, обобщения и анализа опыта существующих решений.

Ведомость

содержание

содержание

**Введение**

Социально-экономический прогресс в эпоху высоких технологий и информатизации во многом зависит от уровня развития системы образования, возрастает ее роль и значение, а также возникает необходимость повышения ее качества и доступности. Экономика все больше базируется на знаниях, инновационном и технологическом потенциале. Образование в Республике Беларусь является одним из приоритетов государственной политики. В современных условиях важнейшим ресурсом социально-экономического развития и главным капиталом современного общества является личность: ее творческий потенциал, навыки, умения разрабатывать новые идеи и эффективно их реализовывать. В этом контексте важное значение имеет учебный процесс, который позволяет подготовить качественного специалиста. Важно отметить, что современный учебный процесс не стоит на месте и требует непрерывной модернизации, в связи с эти вводятся экспериментальные формы организации учебной деятельности, разрабатываются разнообразные образовательные порталы, расширяющие перечень форм взаимодействия педагога и обучающегося. Особенно остро этот вопрос встал перед образовательными организации в текущем году. Огромное количество веб-сервисов в данный момент разработано специально для обеспечения образовательного процесса и его внутреннего наполнения: использование и внедрение этих сервисов позволяет разнообразить уроки, обучать на практике владению информационными технологиями, расширять возможности преподавания.

Однако, как и традиционная форма образования, электронная имеет необходимую учебно-планирующую документацию, благодаря которой учебный процесс удается планировать и систематизировать.

Образование является одной из важнейших сфер человеческой деятельности, обеспечивающей формирование интеллектуального потенциала общества. Сложное положение образования в настоящее время определяется рядом проблем, среди которых противоречие между традиционным темпом обучения и постоянно увеличивающейся скоростью появления новых знаний. По этой причине в систему образования внедряют новые информационные технологии. В связи с этим возникла необходимость в преподавателе-новаторе с творческим научно-педагогическим мышлением, который на достаточно хорошем уровне знает и умеет применять новые информационные и педагогические технологии в своей практике.

**1 Технологии обучения на основе использования образовательных сервисов web 2.0 в ходе изучения темы учебного предмета «Информатика»**

* 1. **Организация обучения в учреждения профессионально-технического и среднего специального образования Республики Беларусь с использованием образовательных сервисов**

Профессиональное образование - начальное образование, направленное на развитие личности студента, его профессиональное развитие, получение специализированной теоретической и практической подготовки, завершающейся получением квалификации работника с профессиональным образованием.

Учреждения профессионального образования включают в себя:

* профессиональное училище;
* профессиональный лицей;
* профессионально-технический колледж.

Профессиональное училище дает возможность выпускнику основной школы в течение трех лет получить не только рабочую профессию, но и общее среднее образование. На базе общего среднего образования в профессиональном училище учащиеся получают профессию, которая востребована на рынке труда, так как планы приема в профессиональные училища формируются с учетом потребностей предприятий региона в рабочей силе.

Профессиональный лицей. Наряду с профессией студенты изучают профильные предметы на продвинутом уровне.

Профессионально-технические колледжи. Они обеспечивают обучение на двух уровнях. I-й уровень профессионального образования – подготовка рабочей профессии; II-й уровень среднего специального образования -подготовка специалистов среднего звена- техников, технологов.

Прием на обучение по определенной образовательной специальности (профессии) осуществляется при отсутствии медицинских противопоказаний и противопоказаний к работе по соответствующей специальности (профессии).

На обучение по профессиям, запрещающим труд несовершеннолетних, принимаются только лица, которым на момент окончания школы исполнилось 18 лет.

Среднее специальное образование – уровень основного образования, направленный на развитие личности студента, курсанта, получающего специальную теоретическую и практическую подготовку, завершающуюся присвоением квалификации специалиста со средним специальным образованием, работника со средним специальным образованием.

Учреждения среднего специального образования представлены во всех регионах республики и готовят специалистов среднего звена для различных отраслей экономики.

К образовательным учреждениям, предлагающим среднее специальное образование, относятся техникумы, колледжи, школы искусств, гимназии, профессиональные колледжи, лингвистические колледжи.

Среднее специальное образование предоставляется на базе общего основного, общего среднего и профессионального образования. Обучение проводится в очной, заочной и вечерней формах и длится от 2 до 4 лет.

Для поступления в среднее специальное учебное заведение абитуриенты сдают два вступительных испытания: на белорусском или русском языках (по выбору абитуриента) и по учебному предмету в соответствии с профилем (направлением) специальности.

Профессионально-техническое образование призвано готовить студентов к профессиональной деятельности в соответствии с их призванием, способностями и социальными потребностями и обеспечивает приобретение ими профессиональных знаний, умений и навыков, необходимых для получения квалификации рабочих и служащих.

В соответствии с законом профессиональное обучение, переподготовка, повышение квалификации, трудовое (профессиональное) обучение, внешкольное обучение и образование, а также иные образовательные услуги предоставляются в учреждениях, осуществляющих профессиональное образование.

В целях дальнейшего развития системы профессионального образования работа Министерства образования Республики Беларусь направлена на решение следующих задач:

* организация совместной работы с областными исполнительными комитетами, Мингорисполкомом и другими органами государственного управления по созданию системы прогнозирования потребности в трудовых ресурсах для формирования заказа на подготовку кадров в системе образования на 2011-2015 годы;
* формирование структуры и объема подготовки рабочих (служащих) с учетом увеличения числа выпускников общеобразовательных учреждений, получивших общее базовое образование, и в соответствии с потребностями отраслей экономики и социальной сферы в молодых работниках;
* укрепление связей между УПТО и базовыми организациями на основе заключения соглашений об их взаимодействии при подготовке рабочих (служащих);
* повышение престижа образования в УПО, целенаправленное обучение и сокращение числа отчислений студентов по неуважительным причинам.

В 2010 году система профессионального образования продолжает совершенствоваться по следующим направлениям:

* достижение полного соответствия структуры и объемов подготовки молодых кадров потребностям организаций отдельных отраслей экономики и регионов;
* замена частей машин и оборудования, используемых для производственного обучения в учебных заведениях, которые не отвечают установленным требованиям и имеют значительный износ;
* дальнейшая интеграция в содержание подготовки кадров среднего специального и профессионального образования в целях сокращения сроков обучения, внедрения новых образовательных программ подготовки работников со средним специальным образованием;
* разработка и издание национальной учебной литературы по массовым специальностям профессионального образования;
* увеличение объема учебных программ по профессиональной подготовке, переподготовке и повышению квалификации по рабочим профессиям в отдельных регионах.

Основные направления развития среднего специального образования

В 2019 году в соответствии с заявками Минобрнауки выдало разрешения на открытие подготовки специалистов в УСО по следующим специальностям: «Промышленное и гражданское строительство», «лесное хозяйство», «машины и оборудование деревообрабатывающей промышленности», «дошкольное образование», «физическая культура».

В республике реализуется реформа среднего специального образования в направлении его интеграции в структуру высшего образования с целью развития системы непрерывного профессионального образования.

В целях интеграции среднего и высшего образования на базе средних специальных учебных заведений открыт колледж высшего образования, обеспечивающий подготовку кадров не только среднего, но и высшего образования. В рамках программы регионализации высшего образования апробируются варианты более глубокой интеграции – в том числе средние специальные учебные заведения как структурные подразделения высших учебных заведений.

Для обеспечения реализации образовательных программ ПТО по каждой специальности-уровню ТПО разработаны образовательные стандарты, устанавливающие требования к содержанию профессиональной деятельности работника (работника), его компетентности, содержанию учебно-программной документации (ДПК), уровню базового образования лиц, поступающих на получение ТПО, формам и срокам ее получения, организации учебного процесса, учебной нагрузке обучающихся, уровню подготовки выпускников, итоговой аттестации. Научно-методическое обеспечение ПОО включает в себя: УПД; образовательные программы по учебным дисциплинам; программно-плановая документация образования; учебно-методическая документация; учебные издания; информационно-аналитические материалы.

Срок получения профессионального образования в очной форме обучения составляет 1 – 3 года, в зависимости от образовательной программы, в заочной – 0,5 – 1,5 года. Учебный процесс может быть организован как в учебных заведениях, так и на дому.

Учреждения ПОО могут быть следующих типов: профессиональное училище; профессиональный лицей; профессиональное училище. В настоящее время в системе ПОО действуют 217 учреждений ПОО Министерства образования Республики Беларусь, в том числе 37 профессиональных училищ, 111 профессиональных лицеев, 69 профессиональных колледжей.

Профессиональное образование осуществляется в соответствии с учебно-программной документацией образовательных программ профессионального образования, разработанной на основе образовательных стандартов профессионального образования по специальности.

Учебный процесс при реализации образовательных программ профессионального образования организуется по учебным годам и осуществляется в учебных группах (25-30 человек в дневное время, 15-20-в вечернее и заочное время) или индивидуально. Учебный год делится на 2 полугодия. Основной формой организации учебного процесса является Учебное занятие: урок, лекция, семинар, лабораторная, практическая и другие виды деятельности. Образовательный процесс включает в себя производственную подготовку.

Лица с общим базовым, общим средним или специальным образованием принимаются в образовательное учреждение для очной профессиональной подготовки. Прием осуществляется на основании их заявлений. Если количество заявок на ту или иную специальность превышает установленные для нее контрольные показатели приема, зачисление осуществляется по конкурсу на основании среднего балла аттестата об общем базовом образовании, аттестата об общем среднем образовании или аттестата о специальном образовании.

При разработке образовательных программ профессионального обучения проводится текущая и итоговая аттестация обучающихся. Формами текущей аттестации являются: анкетирование студента; Контрольная работа; проверочная работа по производственному обучению; зачет; экзамен по предмету; квалификационный экзамен. Результаты текущей аттестации по учебным предметам профессионального компонента оцениваются балльными оценками по десятибалльной шкале или баллами «зачтено» или «не зачтено». Положительные оценки составляют не менее 3 (трех) баллов и «зачисляются».

Итоговая аттестация проводится государственной квалификационной комиссией и проводится в форме итогового квалификационного экзамена. Итоговая аттестация по образовательным предметам общеобразовательного компонента осуществляется в форме итоговых экзаменов. Результаты итоговой аттестации студентов по учебным предметам профессионального компонента, т. е. результаты итогового квалификационного экзамена, оцениваются балльными оценками по десятибалльной шкале. Положительные оценки составляют не менее 3 (трех) баллов.

По данным, приведенным в государственной программе развития профессионального образования на 2015-2019 годы, численность работников (наемных работников) в отраслях экономики и социальной сферы Республики Беларусь составляет более 2,5 млн человек (63,7% от общей численности занятых). В то же время численность рабочих (служащих) с профессиональным образованием составляет более 0,8 млн человек (30,3% от общей численности).

Таким образом, в Республике Беларусь создана эффективная система образования, основанная на передовых технологиях, отвечающая международным стандартам.

Определяющими чертами развития сферы образования в ближайшее десятилетие станут: массовый характер образования на всех уровнях образования; формирование системы образования взрослых; развитие дистанционного обучения; реализация коллективных и индивидуальных методов обучения в единстве на основе информационных технологий.

Важное значение в системе образования приобретет использование инновационных принципов и подходов в преподавании, углубление научно-методического обеспечения образования, Обновление и расширение материально-технической базы, рациональное сочетание бесплатных и платных форм обучения для населения, обеспечение широкого спектра образовательных программ и услуг, информатизация образования.

В нашем проекте мы рассматриваем электронные средства обучения, а именно, сетевые социальные сервисы Web 2.0.

Веб 2.0 (Web 2.0) – второе поколение сетевых сервисов, действующих в Интернете. В отличие от первого поколения сервисов, Web 2.0 позволяет пользователям работать с сервисами совместно, обмениваться информацией, а также работать с массовыми публикациями (на основе веб-приложений социальных сервисов). Другими словами, если в основе сервиса заложены принципы коллективизма, кооперации, открытости, доступности, интерактивности, то это Web 2.0. Появление термина Web 2.0 принято связывать со статьёй «Tim O'Reilly – What Is Web 2.0» от 30 сентября 2005 года, впервые опубликованной на русском языке в журнале «Компьютера» (№ 37 (609) и № 38 (610)). В этой статье Тим О'Рейли связал появление большого числа сайтов, объединенных некоторыми общими принципами, с общей тенденцией развития интернет- сообщества, и назвал это явление Web 2.0, в противовес «старому» Web 1.0.

Социальные сервисы Web 2.0 – это современные средства, поддерживающее групповые взаимодействия. Эти групповые действия включают:

* персональные действия участников и коммуникации участников между собой;
* записи мыслей, заметки и аннотирование чужих текстов (Живой журнал, блог или ВикиВики);
* размещение ссылок на интернет-ресурсы и их рейтингование (БобрДобр);
* размещение фотографий (Фликр, Фотки.ру);
* размещение книг (возможны иллюстрации) (Скрибд);
* видеосервисы (Ютьюб, Социальная сага, видеоблог);
* компиляция на одной странице «сборной солянки» из различных интернет сервисов;
* географические сервисы (Земля Гугл, Викимапия) и сервисы на их основе (так называемые мэшапы (от англ. «mash up») (Панорамио, Фликр), моделирование объектов в 3D (Скетчуп));
* обмен сообщениями (мессенджеры, электронные RSS-рассылки, Скайп). Сервисы Web 2.0 поддерживают самопроизвольный путь развития сообществ, когда они не создаются по указанию сверху, а складываются снизу- вверх из небольших усилий множества формально независимых участников.

Новые сервисы социального обеспечения Web 2.0 радикально упростили процесс создания материалов и публикации их в Сети. Теперь каждый может не только получить доступ к цифровым ресурсам, но и принять участие в формировании собственного сетевого контента. При этом общение между людьми все чаще происходит не в форме прямого обмена высказываниями, а в форме взаимного наблюдения за сетевой деятельностью. Освоение новых средств ведет не только к тому, что мы можем решать новые задачи, но и меняет наше мировоззрение, позволяет нам видеть мир с новой точки зрения.

Социальные сетевые сервисы открывают новые педагогические возможности:

* использование открытых, бесплатных и свободных электронных ресурсов. В результате распространения социальных сервисов в сетевом доступе оказывается огромное количество материалов, которые могут быть использованы в учебных целях. Сетевые сообщества обмена знаниями могут поделиться своими наработками и прочими материалами;
* самостоятельное создание сетевого учебного содержания. Новые сервисы социального обеспечения радикально упростили процесс создания материалов и публикации их в сети. С помощью сетевых социальных сервисов каждый может получить доступ к цифровым коллекциям и принять участие в формировании собственного сетевого контента;
* освоение информационных концепций, знаний и навыков. Среда информационных приложений открывает принципиально новые возможности для деятельности, в которую легко вовлекаются люди, не обладающие никакими специальными знаниями в области информатики. Новые формы деятельности связаны как с поиском в сети информации, так и с созданием и редактированием собственных цифровых объектов – текстов, фотографий, программ, музыкальных записей, видеофрагментов. Участие в новых формах деятельности позволяет осваивать важные информационные навыки – повторное использование текстов и кодов, использование мета-тегов и т. д. Мета-тег – это специальный тег, т.е. набор символов в HTML-разметке страницы, в который помещают слова, отражающие содержание веб-страницы. Информация, размещенная в мета- теге, обладает большим значением для поисковых систем и в первую очередь используется при анализе сайта;
* наблюдение за деятельностью участников сообщества. Сеть Интернет открывает новые возможности для участия школьников в профессиональных научных сообществах. Цифровая память, агенты и сеть удивительно расширяют не только мыслительные способности участников учебного процесса, но и поле для совместной деятельности и сотрудничества с другими людьми;
* создание учебных ситуаций, в которых можно наблюдать и изучать недоступные ранее феномены. С развитием социального обеспечения сетевая деятельность или сетевое поведение других людей становится все доступнее. Совместные действия участников современных сетевых объединений зачастую носят децентрализованный характер.

Таким образом, социальные сетевые сервисы Web 2.0 могут повысить эффективность процесса обучения, потому что их применение открывает широкие возможности перед всеми участниками учебного процесса.

Можно сделать вывод, что сетевые социальные сервисы Web 2.0 раскрывают широкие возможности не только перед обычными пользователями сети Интернет, но и перед всеми, без исключения, участниками образовательного процесса. Особенно сетевые социальные сервисы удобно использовать в проектной деятельности. При этом мы преследуем две цели: достаточно удобную реализацию проекта и знакомство учащихся с возможностями сетевых социальных сервисов Web 2.0.

**1.2 Дидактический анализ темы учебного предмета**

Дидактический анализ учебного предмета представляет собой анализ основных дидактических категорий (отбор методов, приемов и средств обучения, реализация принципов дидактики, дидактическая обработка учебного материала занятия, педагогическое руководство самостоятельной познавательной деятельностью учащихся).

Методы обучения – это способы совместной деятельности обучающих и обучаемых, направленные на достижение ими образовательных целей. Составной частью метода обучения являются средства обучения – это различные объекты, используемые педагогом и учащимися в процессе обучения.

В современной дидактике выделяют:

* словесные методы;
* наглядные методы;
* практические методы.

На занятии используются словесные (лекция и беседа) и наглядные (иллюстрации, демонстрации) методы обучения.

Лекция как метод обучения представляет собой последовательное изложение преподавателем темы или проблемы, при котором раскрываются теоретические положения, сообщаются факты, события и дается анализ их, раскрываются связи между ними. Выдвигаются и аргументируются отдельные научные положения, освещаются различные точки зрения по изучаемой проблеме и обосновываются правильные позиции. Лекция – самый экономичный путь получения информации, так как в лекции педагог может сообщить научные знания в обобщенном виде, почерпнутые из многих источников и которых еще нет в учебниках. Лекция, кроме изложения научных положений, фактов и событий, несет в себе силу убеждений, критической оценки, показывает студентам логическую последовательность раскрытия темы, вопроса, научного положения. Преподаватель непрерывно следит за аудиторией, за вниманием студентов, и в случае его падения, принимает меры по повышению интереса студентов к материалу: изменяет тембр и темп речи, придает ей большую эмоциональность,

Беседа – диалогический метод обучения, при котором преподаватель путем постановки тщательно продуманной системы вопросов подводит студентов к пониманию нового материала или проверяет усвоение уже изученного. Преподаватель, опираясь на знания и опыт студентов, последовательной постановкой вопросов подводит их к пониманию и усвоению новых знаний.

Метод иллюстраций предполагает показ студентам иллюстрированных пособий. На данной занятии таким пособием является презентация. Презентация содержит тактовый материал с уникальными фактами, слайды с вопросами, фотографии устройств компьютера или значка и интерфейса приложения. Использование таких средств мультимедиа экономит учебное время, способствует комплексному восприятию и лучшему запоминанию материала, нежели при работе у классной доски.

Демонстрация – метод обучения, который основан на наглядном предъявлении обучающимся динамичных изображений, например, видеороликов. Применение данного метода на занятии даёт возможность учащимся изучить комплектующие ПК в натуральном виде или их изображе­ния, а также правильные действия для взаимодействия с ними. Демонстрация средств наглядности обязательно сопровождается объяснением преподавателя, рассказом или проведением беседы.

Перед изучением нового материала используется фронтальный опрос. При фронтальном опросе за короткое время проверяется состояние знаний учащихся всей подгруппы по определенному вопросу или группе вопросов. Эта форма проверки используется для выяснения готовности класса к изучению нового материала.

После изучения темы проводится самоконтроль учащихся при использовании кроссворда. Самоконтроль является особенно важным для развития учащихся, потому что в этом случае осознается правильность своих действий, обнаружение совершенных ошибок, анализ их и предупреждение в дальнейшем.

Педагогические программные средства в сочетании с традиционными печатными материалами помогают преподавателю приблизиться к индивидуальному обучению, что наиболее эффективно в условиях преподавания на персональных компьютерах. Независимо от типа компьютера и уровня знаний учащегося, преподаватель информатики может и должен найти для каждого студента сферу применения своих интересов и способностей

Для упрощения организации обучения созданы основные принципы обучения – положения, в которых отражены теоретические подходы к построению учебного процесса. Эти принципы определяют направленность обучения, его цели и способы и многое другое, а также позволяют оптимизировать весь процесс обучения, сделав его максимально продуктивным. Отследим выполнение дидактических принципов обучения в учебном материале.

Принцип научности:

* соблюдена логика изложения учебного материала;
* содержание учебного материала не содержит фактических ошибок, соответствует развитию современных научных знаний.

Принцип наглядности:

* при обучении соблюдается мера в наглядном представлении изучаемого материала, не теряется главное его содержание;

Принцип активности и сознательности:

* учение опирается не на авторитет педагога, а на доказательства и логику.

Принцип доступности:

* на занятии выполняется правило постепенного усложнения учебного материала;
* на учебном занятии используются те методы обучения, которые соответствуют возрасту и развитию учащихся;
* при первичном знакомстве с новым материалом изучается главное, а в дальнейшем вводятся новые факты, расширяются и углубляются знания.

Принцип систематичности и последовательности:

* структура занятия хорошо продумана преподавателем, не оставляется без внимания ни один вопрос;
* реализуется логика формирования теоретических знаний.

Принцип завершенности обучения:

* применяемые методы обучения соответствуют его содержанию;
* переход от первого уровня усвоения знаний ко второму и третьему осуществляется с обязательной фиксацией факта усвоения;
* применяются аналогии, сравнения, сопоставления, противопоставления.

Принцип воспитательного воздействия занятия:

* положительное влияние профессиональных, личностных, харизматических качеств преподавателя;
* требования преподавателя не вызовут негативной реакции у обучающихся;
* на всех этапах занятия эффективно организована учебно-познавательная деятельность учащихся.

Занятие на тему «Аппаратное и программное обеспечение компьютера» является занятием формирования новых знаний и проводится в форме лекции.

Цель занятия формирования знаний – организация работы по усвоению учащимися понятий, установление свойств изучаемых объектов, научных фактов, предусмотренных учебной программой.

Современное занятие формирования знаний на основе сочетания разнообразных методов и средств обучения решает комплекс задач. Используются как объяснительно-иллюстративные, так и частично поисковые, исследовательские методы обучения, дискуссия, разнообразные источники знаний, программы телевидения, кинофрагменты, магнитофонные записи, мультимедийные курсы, интернет-технологии, другие технические средства обучения и контроля. Широко используются также разнообразные формы работы: групповая, фронтальная, звеньевая, парная, индивидуальная.

Структура занятия по формированию новых знаний сочетает следующие этапы:

* организационный момент и подготовка к изучению нового материала;
* изучение нового материала;
* обобщение, частичное закрепление и систематизация знаний;
* подведение итогов обучения и рефлексия;
* определение домашнего задания и инструктаж по его выполнению.

В соответствии с вышеперечисленной структурой и типом занятия рассмотрим каждый этап подробнее.

Организационный момент включает в себя приветствие, проверку присутствующих путем озвучивания фамилий, сообщение темы, цели, задач занятия и мотивация учебной деятельности. На данный этап отводится 5 минут.

Изучение нового материала является самым продолжительным этапом занятия формирования новых знаний. Для подачи материала в легкой для усвоения обучающимися форме используются словесный и наглядный методы обучения – применение презентации и видеороликов.

В рамках этого этапа в течении 30 минут преподаватель объясняет материал, опираясь на материал презентации и предлагая законспектировать важную или трудноусваиваемую информацию, а при просмотре видеоролика продолжительностью 25 минут комментирует некоторые моменты для лучшего усвоения нового материала.

Также нельзя исключать факт того, что тема для некоторых может быть далеко не новой. Данный факт объясняет использование на занятии такого метода активного обучения как беседа. При проведении беседы необходимо рассмотреть следующие вопросы:

* разбирали ли учащиеся системный блок;
* знакомы ли с какими-нибудь устройствами компьютера;
* какие виды различных устройств бывают;
* какие программы используют для выполнения домашних заданий, а какие для проведения досуга.

После этапа «изучение нового материала» следует этап «обобщение, частичное закрепление и систематизация знаний». Обучающимся предлагается за 18 минут решить кроссворд, содержащий 18 вопросов-определений изучаемой темы, и записать ответы на письменных бланках. После того, как время выйдет, обучающиеся сверяют ответы с правильными ответами и записывают общее количество верных ответов в бланк. После выполнения бланки сдаются преподавателю. Общее время проверки 2-3 минуты.

При «подведении итогов» преподаватель оглашает результаты выполнения. На данном этапе уместна рефлексия в виде вопросов «Что вы узнали нового?», «Что вызвало интерес?», «Что показалось сложным или неважным для изучения?». Время на этап – 5 минут.

В конце занятия преподаватель озвучивает цель, содержание и способы выполнения домашнего задания, отвечает на вопросы – 5 минут.

Чёткое следование описанной структуре занятия на тему «Аппаратное и программное обеспечение компьютера» обеспечивает изучение темы в рамках 2 учебных часов. Структурно-логическая схема представлена на плакате

ДП 1090351602-2020-01ПЛ.

**1.3 Разработка учебно-планирующей документации**

Работа по организации учебного процесса носит комплексный характер и включает в себя: изучение нормативно-правовых документов среднего профессионального образования, внедрение Федеральных государственных образовательных стандартов в образовательный процесс, разработку комплекса учебно-методического обеспечения учебного процесса, подготовку материально-технические и информационно-обучающей базы. На основе этого перечня составляется учебно-планирующая документация, играющая немаловажную роль в образовательном процессе, она позволяет его четко организовать, выполнять учебный план и программу, грамотно готовиться к занятиям, правильно подбирать учебно-производственные работы и эффективно использовать материально-технические и информационно коммуникативные средства учебных мастерских и лабораторий. План-конспект учебного занятия представлен в приложении А, технологическая карта учебного занятия представлена в приложении Б.

**2 Разработка программного обеспечения**

**2.1 Аналитический обзор программного обеспечения по теме проекта**

2.1.1 Функциональный анализ предметной области

Бизнес-процесс является логичным, последовательным, взаимосвязанным набором мероприятий, потребляющим ресурсы, создающим ценность и выдающим результаты. Согласно международному стандарту ISO 9000:2000 было принято такое понятие как «процесс», однако сегодня подобные понятия, можно сказать, обозначаются в качестве синонимов. Построение бизнес-процессов служит в качестве действенного и наиболее продуктивного средства для определения способов в направлении оптимизации деятельности компании, которое позволяет установить, каким образом предприятие работает и каким образом осуществляется деятельность сотрудников на своих рабочих местах.

Перед тем, как выполнить автоматизацию установленной предметной области требуется решить, какое направление и разновидность построения бизнес-процессов можно использовать. В практической деятельности формирование бизнес-процессов предприятия совершается поэтапно:

* этап структурного моделирования;
* этап детального моделирования.

Структуризация бизнес-процессов компании, как правило, производится посредством нотации IDEF0 либо на языке UML.

Структурное моделирования рассматривается в следующих показателях:

* действующая организационная структура;
* документы и прочие сущности, использующиеся в ходе проведения моделируемых бизнес-процессов и требующихся для целей по составлению документооборота, с критериями их основной сути;
* структура бизнес-процессов, отражающая их иерархичные шаги от более общих групп бизнес-процессов к частным бизнес-процессам;
* диаграммы взаимодействий для конечных бизнес-процессов, которые отражают последовательные действия по составлению и перемещения документов (данные, материалов, ресурсов и пр.) среди действующими лиц.

Детальное моделирование осуществляется на основе этой же модели и обязано показывать соответствующую детализацию, в том числе организовать непосредственное представление о деятельности компании.

Потребность в построении бизнес-процессов заключается в описании методик изменения исходной информации в отчеты и диаграммы.

После того, как проанализирована предметная область, была разработана многофункциональная модель основных бизнес-процессов. Процессы моделирования были исполнены на основе нотации IDEF0.

Нотация IDEF0 представляет собой метод моделирования, который позволяет разработать многофункциональную модель, показывающую в полной мере структуру и функциональные возможности системы, в том числе информационные потоки и потоки материальных объектов, что объединяет эти функции. Бизнес-процессы в нотации IDEF0 выглядят как прямоугольники, а стрелки демонстрируют взаимосвязь с прочими процессами и окружающими факторами.

IDEF0 дает возможность выстраивать многофункциональные модели, описывающие бизнес-процессы в форме иерархической системы взаимодействующих функциональных возможностей. Любая IDEF0-диаграмма содержит блоки и дуги. Блоки изображают функции выстраиваемой ИС. Дуги объединяют блоки и показывают, каким образом происходит взаимодействующие и взаимосвязывающие процессы между ними.

Для того, чтобы начать формировать модель нужно представить всю систему как один, состоящий из блока и дуг, которые изображают интерфейсы с функционалом вне системы. Далее блок, представляющий систему в качестве единого модуля, необходимо детализировать на новой диаграмме, используя несколько блоков, которые соединяются с помощью интерфейсных дуг. Все детальные диаграммы выступает в качестве декомпозиции блока из диаграммы предшествующего уровня. На каждом этапе декомпозиции диаграмма предшествующего уровня обозначается как родительская для наиболее детальной диаграммы.

Нотация IDEF0 применяется для того, чтобы построить верхний уровень модели бизнес-процессов. Формирование IDEF0-диаграммы верхнего уровня отвечает за обеспечение более целостного или абстрактное описание объекта моделирования. Перед тем как утвердить построенную модель обязательно нужно провести согласование с архитекторами и IT-разработчиками для того, чтобы подтвердить, что структура бизнес-процессов наглядна, достаточно интуитивно-понятная и доступная. Модели в обязательном порядке необходимо согласовывать с ответственными специалистами организации, обладающими полной информацией по конкретному бизнес-процессу.

Для оценки бизнес-процессов компании использовались возможности программы AllFusion Process Modeler (BPwin), которая позволяет создавать функциональные модели в методологии IDEF0 и предназначен для функционального моделирования и анализа деятельности предприятия. Модель в AllFusion Process Modeler представляет собой совокупность диаграмм, каждая из которых описывает отдельный процесс в виде разбиения его на шаги и подпроцессы. С помощью соединяющих дуг описываются объекты, данные и ресурсы, необходимые для выполнения функций.

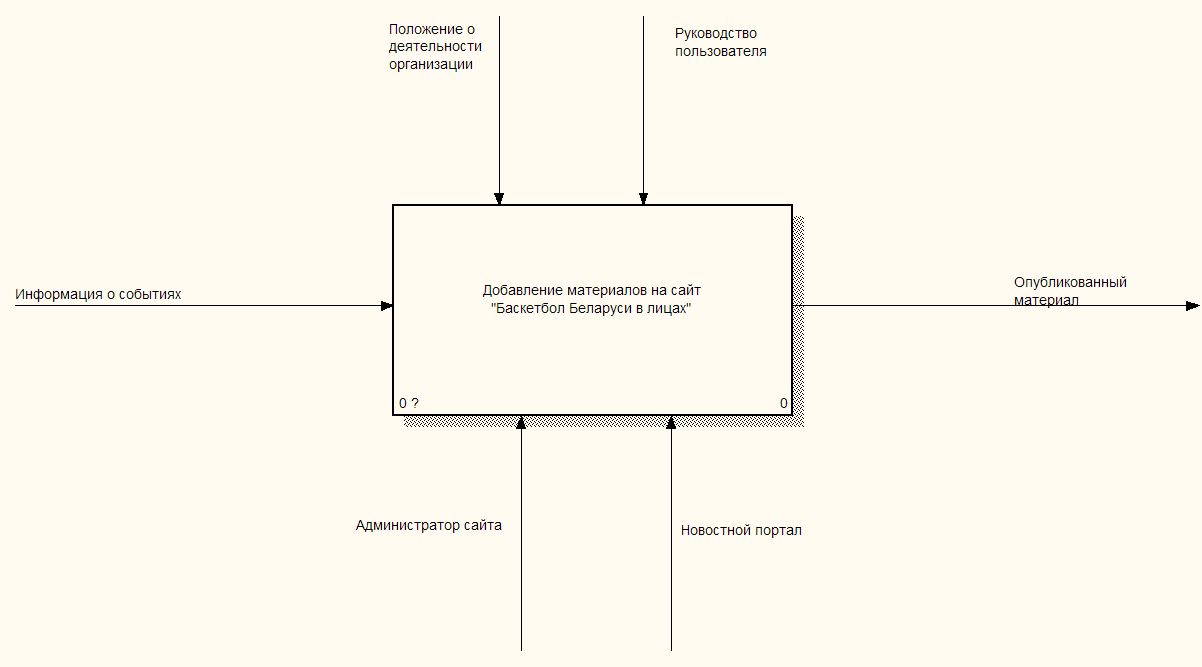


Рисунок 2.1 – Функциональная модель

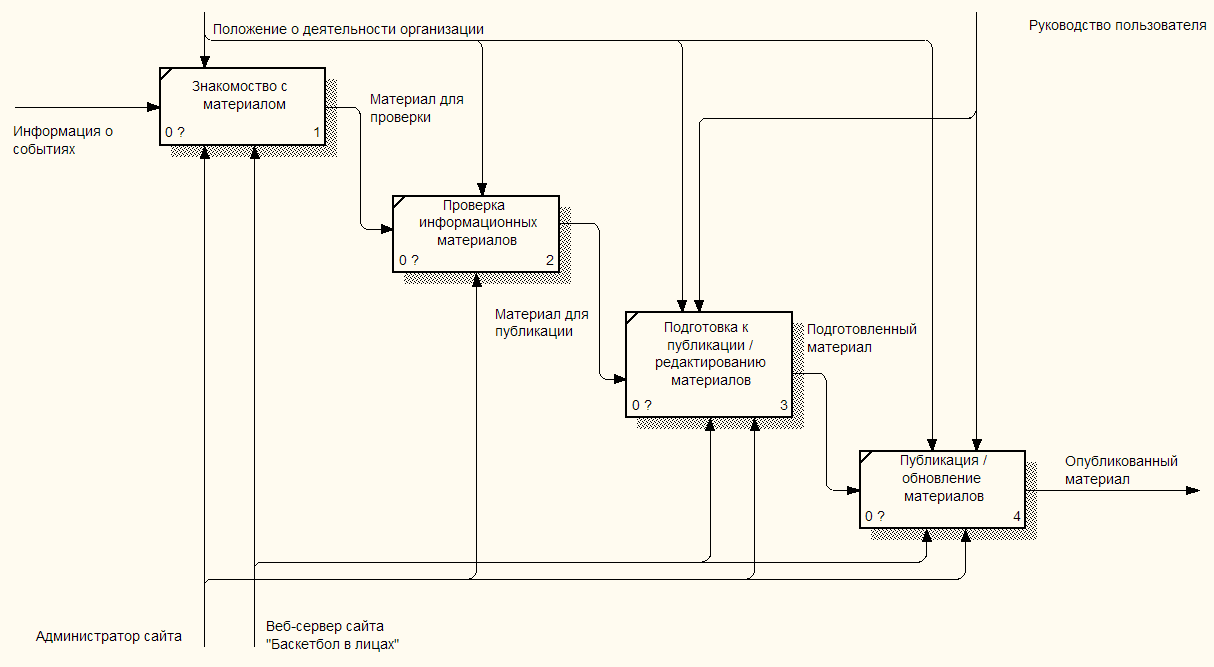


Рисунок 2.2 – Декомпозиция функциональной модели

В результате функционального анализа предметной области была структурирована следующая информация.

Баскетбол — спортивная командная игра с мячом, в которой мяч забрасывают руками в корзину (кольцо) соперника.

В баскетбол играют две команды, каждая из которых состоит из пяти полевых игроков (замены не ограничены). Цель каждой команды — забросить мяч в кольцо с сеткой (корзину) соперника и помешать другой команде завладеть мячом и забросить его в свою корзину. Корзина находится на высоте 3,05 м от паркета (10 футов). За мяч, заброшенный с ближней и средней дистанций, засчитывается два очка, с дальней (из-за трёхочковой линии) — три очка; штрафной бросок оценивается в одно очко. Стандартный размер баскетбольной площадки — 28 м в длину и 15 м — в ширину. Баскетбол — один из самых популярных видов спорта в мире.

Баскетбол входит в программу Олимпийских игр с 1936 года (изобретатель игры Джеймс Нейсмит был там в качестве гостя). Регулярные чемпионаты мира по баскетболу среди мужчин проводятся с 1950 года, среди женщин — с 1953 года, а чемпионаты Европы — с 1935 года.

В Европе проводятся международные клубные соревнования по баскетболу: Евролига, Еврокубок, Кубок вызова ФИБА.

Наибольшего развития эта игра достигла в США: чемпионат Национальной баскетбольной ассоциации (НБА) более 50 лет является сильнейшим национальным клубным турниром в мире. Баскетбол считается национальным видом спорта в Литве.

Зимой 1891 года студентам колледжа Молодёжной христианской ассоциации из Спрингфилда, штат Массачусетс, вынужденным выполнять гимнастические упражнения, считавшиеся в то время единственным средством приобщения молодёжи к спорту, было очень скучно на занятиях физического воспитания. Однообразию таких занятий необходимо было положить конец.

Выход из этого положения нашёл преподаватель колледжа Джеймс Нейсмит. 21 декабря 1891 года он привязал две корзины из-под персиков к перилам балкона спортивного зала и, разделив восемнадцать студентов на две команды, предложил им игру, смысл которой сводился к тому, чтобы забросить большее количество мячей в корзину соперников.

Идея этой игры у него зародилась ещё в школьные годы, когда дети играли в старинную игру «duck-on-a-rock» («Утка на скале»). Смысл этой популярной в то время игры заключался в следующем: подбрасывая небольшой камень, необходимо было поразить им вершину другого камня, большего по размеру.

Игра, получившая название «баскетбол», лишь отдалённо напоминала современный вид спорта. Ведения мяча не существовало, игроки только перебрасывали его друг другу, стоя на месте, и стремились затем закинуть в корзину, причём исключительно обеими руками снизу или от груди, а после удачного броска один из игроков забирался на приставленную к стене лестницу и извлекал мяч из корзины. Целью доктора Нейсмита было создать игру именно коллективную, в которую можно было бы вовлечь одновременно большое количество участвующих, и этой задаче его изобретение отвечало в полной мере.

В 1892 году преподавателем физкультуры Смит-колледжа в Нортгемптоне (Массачусетс) Сендой Беренсон были разработаны первые правила женского баскетбола.

2.1.2 Анализ существующих решений и постановка задачи

Для управления содержимым сайта «Баскетбол Беларуси в лицах» требуется система управления содержимым. Система управления сайтом — это программный продукт, который создан для упрощения создания основных видов сайтов. Как правило, системы управления достаточно универсальны, то есть на них можно собрать что угодно: от информационного сайта до интернет-магазина или портала. При этом на внешний вид публичной части сайта (на то, что видит обычный пользователь) CMS никаких ограничений не накладывают.

Есть две разновидности CMS - коробочные (готовый программный продукт, который можно скачать или купить) и самописные (которые можно получить только от разработчика).

Плюсы разработки сайта с использованием CMS:

* на CMS достаточно просто и быстро можно создать те виды сайтов, которые заложены в функционал этой системы. При этом разработка простых решений часто даже не требует участия в проекте программиста;
* многие бизнес-процессы, которые нужны для конечных пользователей или требуются для управления сайтом, уже заложены в функционал CMS. Например, в рамках предустановленного функционала сайт на CMS уже «умеет» редактировать страницы и управлять их структурой, управлять пользователями системы и их уровнем доступа, сортировать и фильтровать различные объекты и многое другое;
* поддержка сайтов на CMS проще, чем работа с системами, написанными «с нуля», этому способствует как наличие документации, так и высокий профессиональный уровень разработчиков CMS, который находит своё отражение в логичной и понятной архитектуре системы;
* возможно простое масштабирование в рамках заложенного функционала. Например, можно достаточно быстро добавить на информационный сайт каталог продукции, а затем сделать на базе каталога интернет-магазин.

Минусы разработки сайта с использованием CMS:

* реализация нестандартного функционала или тонкая настройка под особые бизнес-процессы, как правило, затруднена. Процессы, заложенные в системе, обычно тесно связаны, поэтому модификация одного из них приводит к необходимости модифицировать и зависимые от него. А в некоторых случаях реализация специфического функционала в рамках CMS является и вовсе невозможной;
* обратной стороной универсальности CMS является избыточность функционала и сопутствующие этому увеличение сложности управления и использования, а также некоторые проблемы с производительностью (сайт на CMS выполняет гарантированно больше операции с данными, чем того требует процесс, а структура хранения данных обычно несколько «раздута» относительно реальных потребностей проекта).

CMS бывают платные и бесплатные. Платные разрабатываются коммерческими компаниями и, как правило, распространяются в пакете с теми или иными условиями поддержки от производителя. Большая часть CMS распространяются в рамках лицензии, которая предусматривает неограниченный срок использования продукта и ограниченный срок обновления/поддержки.

В настоящее время CMS превратились в довольно мощные функциональные автоматизированные системы, позволяющие создавать не только блоги, но и сложные многофункциональные web-порталы, которые широко используются пользователями разной квалификации для решения стоящих перед ними Интернет-задач.

Основными задачами, которые решают современные CMS, являются следующие:

* автоматизация всего цикла разработки и публикации web-приложений в Интернете;
* автоматическое изменение представлений содержимого страниц web-портала в сети, обеспечивая возможность реализации различных дизайнов преставления страниц за счет использования разных шаблонов отображения информации, плюс возможность широкого использования шаблонов, созданных другими разработчиками;
* разграничение пользовательских полномочий, что позволяет осуществлять независимую настройку web-портала под представления разных категорий пользователей, включая возможность ведения персональных странниц;
* возможность создания web-портала внутри web-портала, что позволяет создавать сложные web-порталы;
* простота редактирования и создание информации для web-портала;
* возможность автоматически модернизировать web-портал к новым стандартам Интернета;
* возможность разрешения или запрета публикации информации без удаления ее из системы;
* возможность интеграции и использования средств управления проектами по разработке web-приложений;
* масштабируемость, что позволяет развивать системы путем подключения программных расширений от разных разработчиков;
* многоязычность, возможность автоматического перевода и поддержки web-портала для разной национальной целевой аудитории;
* поддержка разных версий сайта;
* обеспечение возможности продвижения web-портала в Интернете путем подстройки к механизмам поисковых систем типа Yandex, Rambler, Google, и др.

Реализация вышеприведенных задач приводит к:

* повышению качества разработки;
* уменьшению стоимости и сроков разработки;
* снижению эксплуатационных расходов;
* понижению требований к персоналу поддержки.

В настоящее время в мире насчитывается десятки коммерческих и бесплатных CMS. В связи с этим встает проблема на каких CMS необходимо сфокусироваться в коммерческой деятельности, чтобы дать пользователям широкие универсальные теоретические и практические для разработки и развертывания приложений в Интернете.

В таблице 2.1 приведена статистика использования CMS в мировом Интернете на июнь 2019 года согласно исследованию компании Q-Success (Software Quality Management Consulting) на сайте World Wide Web Technology Surveys.

Таблица 2.1 – Самые распространенные CMS

|  |  |
| --- | --- |
| Наименование | Использование |
| WordPress | 31,8 % |
| Joomla | 3,1 % |
| Drupal | 2,0 % |
| Squarespace | 1,3% |
| Shopify | 1,3 % |

Как видно из таблицы 2.1, самыми распространенными CMS в мировой практике Интернета являются WordPress, Joomla и Drupal.

Мировая статистика по языкам программирования, использующихся на серверной части web-приложений отражена в таблице 2.2.

Таблица 2.2 – Самые распространенные языки программирования на web-серверах

|  |  |
| --- | --- |
| Наименование | Использование |
| PHP | 83,5 % |
| ASP.NET | 13 % |
| Static files | 2,3% |
| Java | 2,2 % |
| Ruby | 0,6% |

Из таблицы 2.2 видно, что самым популярным языком программирование является PHP, который оставляет далеко позади прочие языки программирования web-приложений.

С точки зрения используемых web-серверов, наиболее распространенным является web-сервер Apache, как видно из нижеприведенной таблицы 2.3.

Таблица 2.3 - Самые распространенные web-сервера

|  |  |
| --- | --- |
| Наименование | Использование |
| Apache | 45,7 % |
| Nginx | 39,5 % |
| Microsoft-IIS | 9,4 % |
| LiteSpeed | 3,4 % |
| Google Servers | 1,0 % |

Исходя из анализа таблицы 2.3 можно сделать вывод, что наиболее распространенной системой управления базами данных (СУБД) для развертывания приложений в Интернете является MySQL, поскольку и WordPress, и Joomla и Drupal изначально и по настоящее время ориентированы в основном на применение MySQL.

В последних версиях этих CMS появилась возможность использования таких СУБД как PostgreSQL, MSSQL.

Если провести анализ отечественного Интернета, то получим подобные результаты.

Схема ресурсов системы представлена на чертеже

ДП-1090351602-2020-01.

Результаты рейтинга CMS по версии компании iTrack представлены в таблице 2.4.

Таблица 2.4 – Рейтинг CMS в доменной зоне ru по версии компании iTrack

|  |  |
| --- | --- |
| Наименование | Использование |
| WordPress | 42,8 % |
| Joomla | 15,18 % |
| 1С-Битрикс | 9,83 % |
| OpenCart | 3,97 |
| Drupal | 3,81 % |
| MODx Revolution | 3,09 % |
| Wix | 2,81 % |
| DataLife Engine | 2,33 % |
| uCoz | 1,9 % |
| MODx Evolution | 1,71 % |
| Прочие | 11,06% |

Рассмотренные программы включают функционал, который не актуален для сайта «Баскетбол Беларуси в лицах», а также не включают ряд функций, которые необходимы для проектируемого сайта. В связи с этим принято решение разработать собственный сайт на основании созданной системы управления контентом, учитывающей необходимый функционал.

Проектируемое программное средство должно включать следующий функционал:

* аутентификация пользователей ПО;
* разделение прав доступа для администратора и руководителя организации;
* добавление записи;
* редактирование записи;
* удаление записи;
* личный кабинет пользователя (администратора).

Схематически изобразить возможности приложения можно с помощью диаграмм вариантов использования (Use Case) которые позволяют отобразить пользователей системы (актеров) и функции, которые выполняет каждый актер. Диаграмма вариантов использования, которая является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения. Диаграмма вариантов использования представлена на рисунке 2.3.



Рисунок 2.3 – Диаграмма вариантов использования

Основные элементы диаграммы - участник (actor) и прецедент (вариант). Участник — это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности. Графически участник изображается «человечком».

Прецедент (use case) - описание множества последовательных событий (включая варианты), выполняемых системой, которые приводят к наблюдаемому участником результату. Прецедент представляет поведение сущности, описывая взаимодействие между участниками и системой. Прецедент не показывает, “как” достигается некоторый результат, а только “что” именно выполняется. Прецеденты обозначаются очень простым образом - в виде эллипса, внутри которого указано его название.

Также, при разработке приложения важно учитывать нефункциональные требования. Нефункциональные требования о описывают характеристики системы и ее окружения, а не поведение системы. Описывают, как именно работает создаваемая система или программный продукт, какими свойствами и характеристиками обладает конкретная разработка.

Нефункциональные требования:

* система должна быть надежной, защищенной, производительной, сопровождаемой;
* программа оптимизирована и предъявляет не высокие требования к серверному и клиентскому аппаратному обеспечению;
* приложение должно представлять собой многостраничное веб-приложение;
* система должна работать в современных браузерах;
* пользовательский интерфейс должен быть удобным, каждый пользователь мог интуитивно понять на какую кнопку нажать, чтобы совершить нужное действие.
* программа интуитивно понятна по надписям на управляющих элементах, по организации меню, самим интерфейсом, позволяющим правильно оценить назначение программы.

Входными данными служат данные, которые пользователи системы вводят в формы на сайте:

* данные пользователя;
* категории;
* статьи;
* комментарии к статьям.
* администраторы.

Выходными данными программы являются страницы сайта, которые генерируются из данных в базе данных: страницы категорий, статей, пользователей, администраторов, комментариев.

В сфере развития сайтов про баскетбол и баскетбольные команды существует несколько аналогов.

Первым представим Баскетбол МГУ им. Ломоносова. Главная страница представлена на рисунке 2.4.

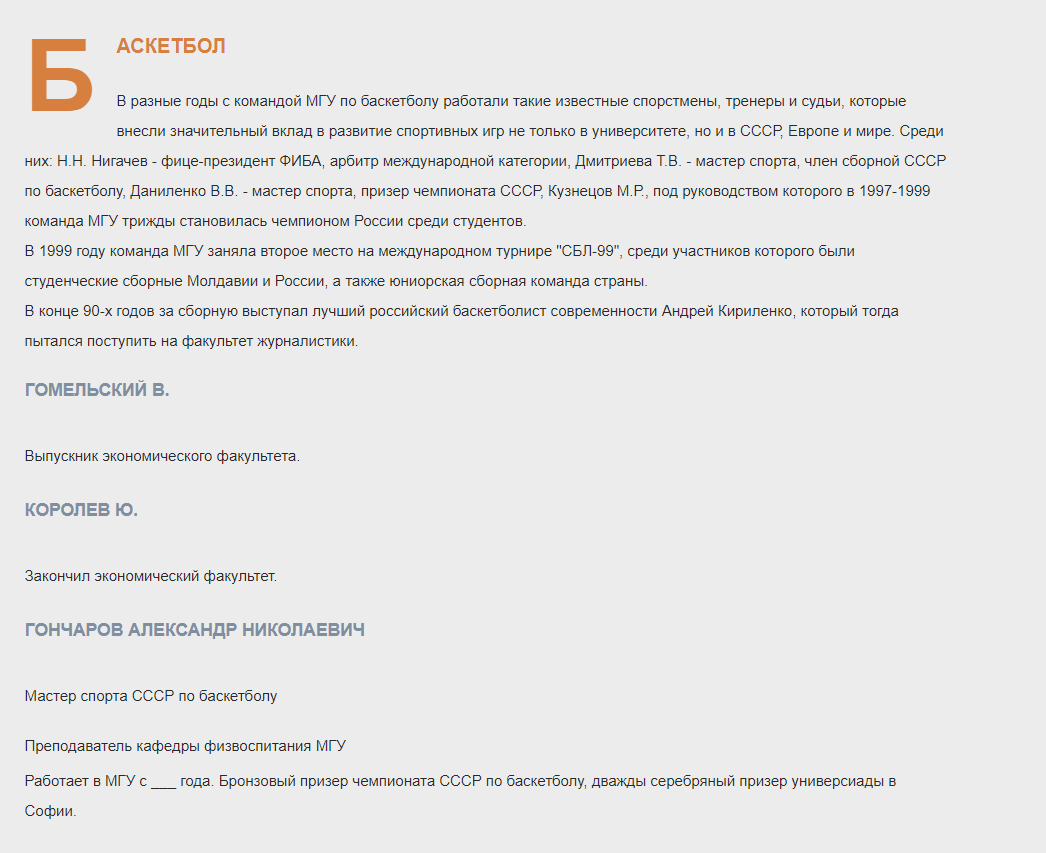


Рисунок 2.4 – Главная страница сайта МГУ ми. Ломоносова

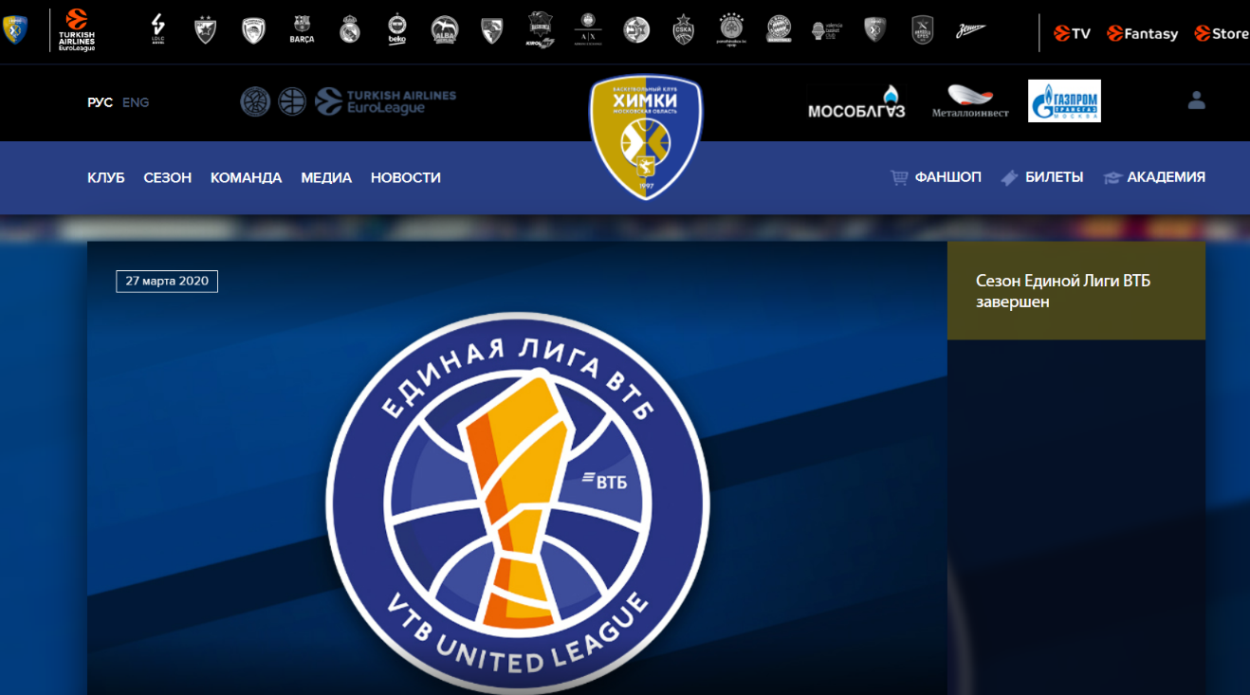
Далее рассмотрим сайт БК Химки. Главная страница представлена на рисунке 2.5. На ней мы можем найти информацию о деятельности клуба, результативности сезона, о составе команды и новости.

Рисунок 2.5 – Главная страница сайта

В разделе «Команда», представленной на рисунке 2.6 содержится информация о каждом из участников команды, вплоть до таких параметров, как рост, вес и возраст.

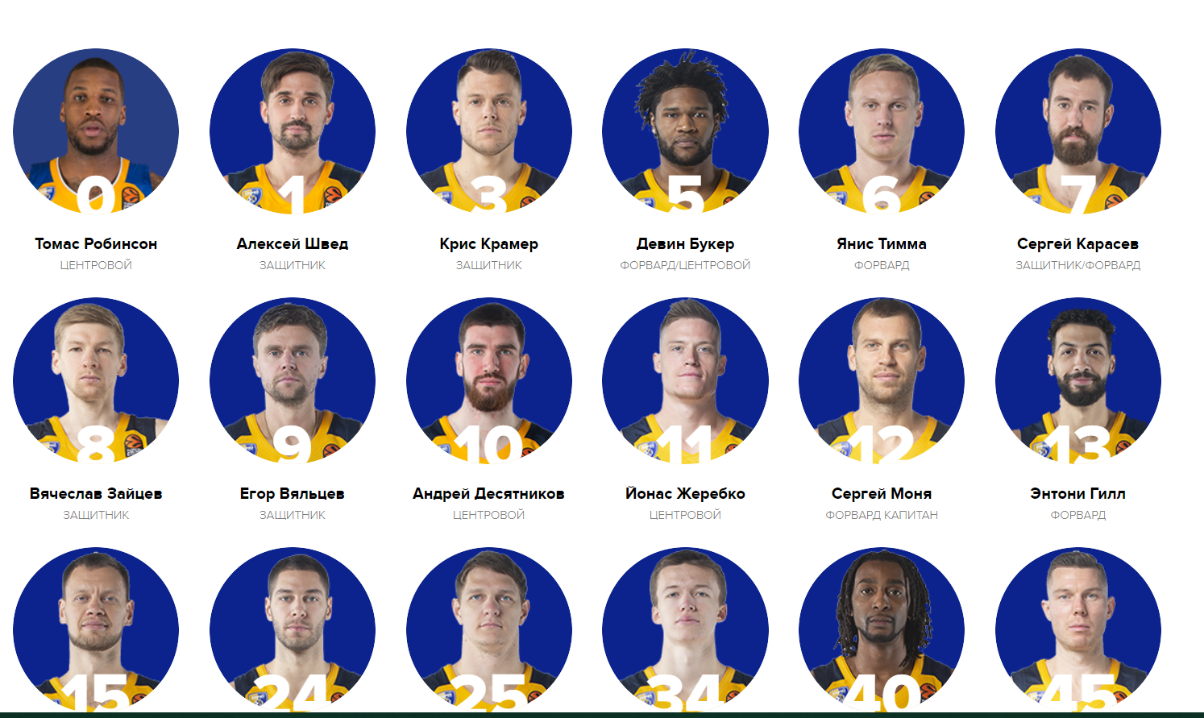


Рисунок 2.6 – Состав команды БК «Химки»

Следующим рассмотрим сайт ПБК ЦСКА.

Главная страница сайта представлена на рисунке 2.7. Здесь мы можем найти информацию о предстоящих событиях клуба, новостях, турнирах, составе команды, а также заказать билеты на матч.

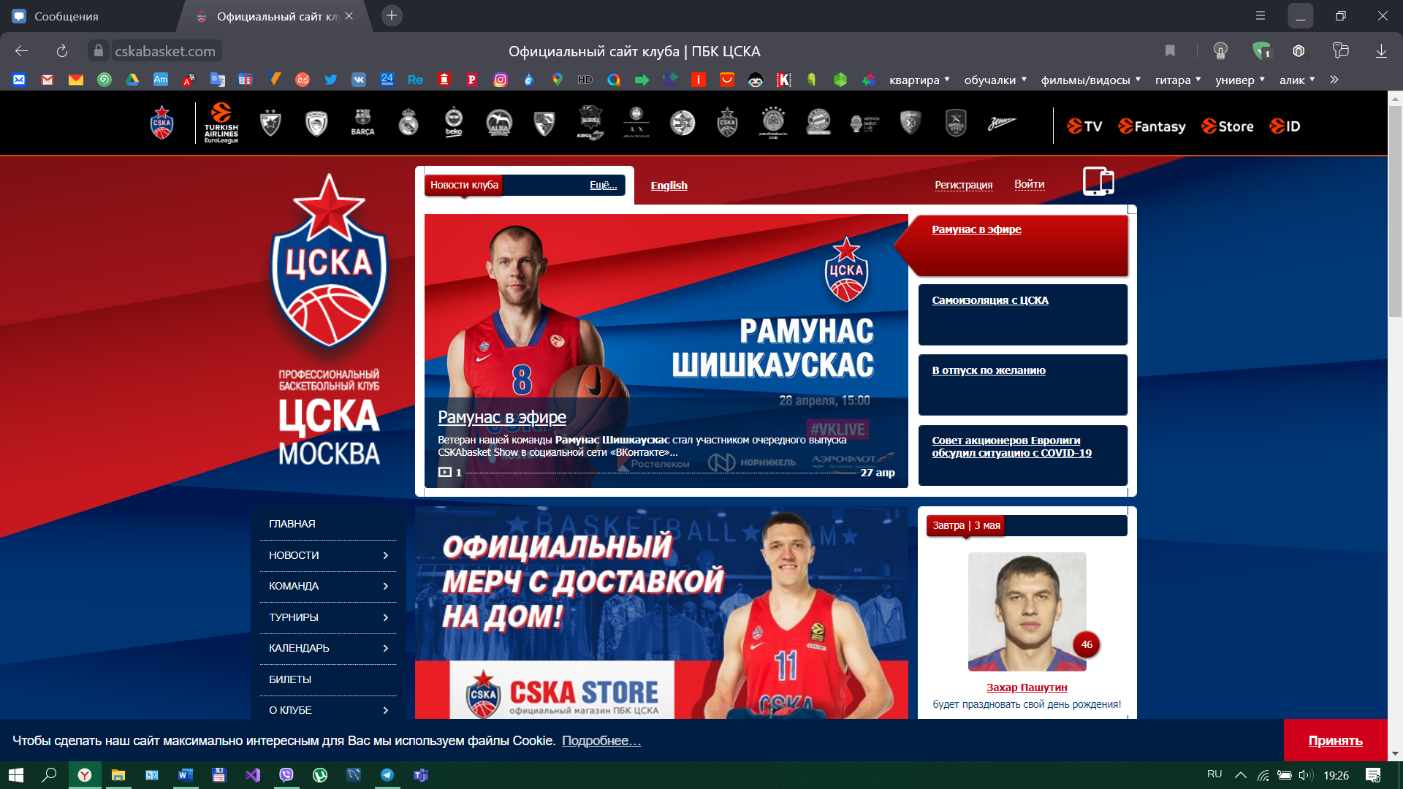


Рисунок 2.7 – Главная страница сайта ПБК ЦСКА

Сведения о составе команды представлены на рисунке 2.8

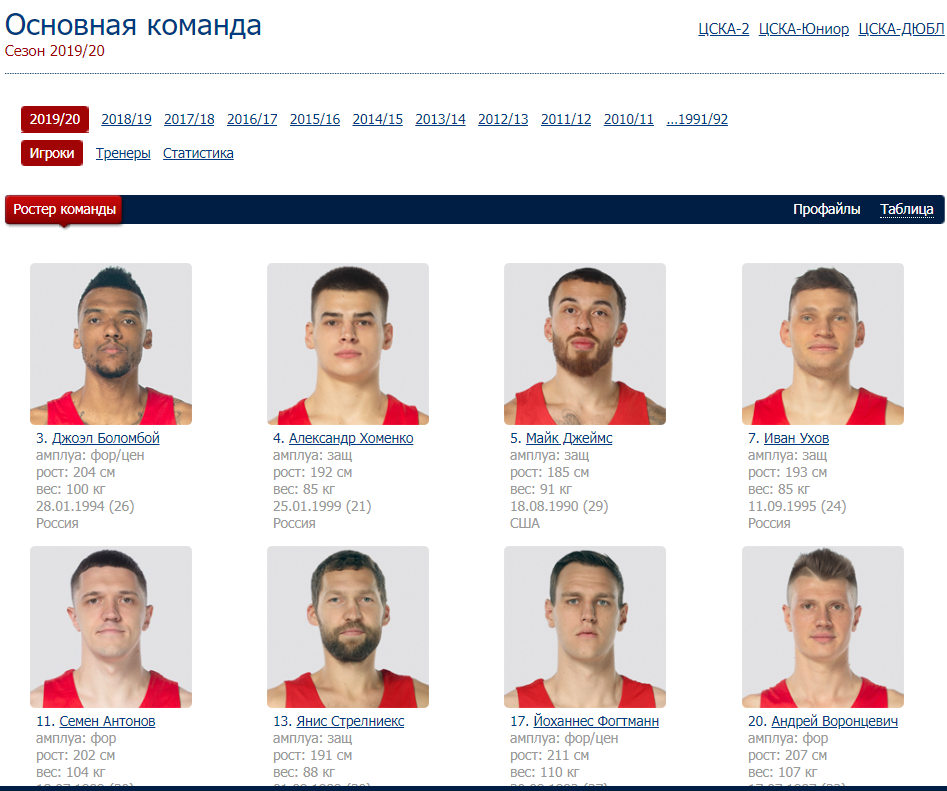


Рисунок 2.8 – Состав команды ПБК ЦСКА

На сайте ПБК Локомотив-Кубань представлена информация о новостях клуба, интернет-магазин, о проводимых матчах, составе команды. На рисунке 2.9 представлена страница сайта, на которой можно найти информацию о каждом участнике команды Локомотив-Кубань.



Рисунок 2.9 – Раздел с участниками команды Локомотив-Кубань

**2.2 Разработка проекта программного обеспечения**

2.2.1 Разработка структуры системы

Структура программного приложения, заключающая в себе программные элементы, их внешние свойства и взаимосвязи, называется архитектурой программы. Важность программной архитектуры объясняется тем, что в ней излагаются начальные проектные решения, она определяет ограничения реализации и организационную структуру разрабатываемой системы, кроме того, программная архитектура позволяет более точно планировать ресурсы, необходимые для создания программного проекта.

Проектируемая система основана на архитектуре клиент-сервер. В базовой модификации клиент-сервер все процессы в распределенных системах разделяются на две вероятно перекрывающиеся группы. Процессы, реализующие определенную службу (servers) и процессы, запрашивающие службы у серверов методом посылки запроса и дальнейшего ожидания отклика от сервера (clients).

Если базовая сеть также надежна, как локальные сети, взаимодействие между клиентом и сервером может быть реализовано посредством простого протокола, не требующего установления соединения.

Согласование клиента и сервера, известное еще под названием «режим работы запрос-ответ», показано на рисунке 2.10.



Рисунок 2.10 – Обобщенное взаимодействие клиента и сервера

У многочисленных системах клиент-сервер применяется надежный протокол (TCP/IP) с установкой соединения. Сервер, как правило, использует для отправки ответного сообщения то же самое соединение, после чего оно разрывается. Любое клиент-серверное приложение включает три уровня: пользовательский интерфейс, обработка информации, хранение данных.

Уровень пользовательского интерфейса как правило реализуется на клиентах. Этот уровень охватывает программы, с помощью которых пользователь взаимодействует с веб-приложением. Уровень обработки обеспечивает основную функциональность приложения, в частности, функции обработки данных, их передачи и т. п.

Уровень данных в модели клиент-сервер содержит программные средства, предоставляющие данные для обработки в приложении. Основной характеристикой и назначением уровня данных является сохранность данных. Это обозначает, что, когда программа не действует, данные должны сохраняться в назначенном месте в расчете на последующее использование. В модели клиент-сервер уровень данных традиционно располагается на стороне сервера. Для базы данных поддержание целостности означает, что метаданные хранятся на этом уровне. Главным после этого представляется самостоятельность данных. Данные сохраняются в хранилище (базе данных), при этом обеспечивается их независимость от веб-приложения, при которой изменения в организации данных не влияют на веб-приложение, а изменения веб-приложение не затрагивают данные в базе данных.

Структура разрабатываемой системы основана на трехуровневой архитектуре. Трехуровневая архитектура является стандартной моделью клиент-серверного приложения. Цель прибавочного («среднего») уровня - управлять прикладным выполнением и управлением базой данных. Как и с двухуровневой моделью, уровни могут размещаться или на разных компьютерах, или на одном компьютере в тестовом режиме, как показано на рисунке 2.11.



Рисунок 2.11 – Трехуровневая архитектура системы

В архитектуре «клиент-сервер» строятся web-приложения, которые представляют собой программные продукты, взаимодействующие через обычный Интернет браузер. Взаимодействие с пользователем осуществляется через страницы и формы, из каких пользователь принимает требуемую ему информацию, а также отправляет некоторую информацию обратно.

Взаимодействие компонентов выбранной трехуровневой архитектуры происходит благодаря использованию протокола HTTP (Hyper Text Transfer Protocol). Он представляет собой протокол прикладного уровня для передачи гипертекста. Является протоколом высокого уровня, который используется в сетях TCP/IP для извлечения файлов с соответствующих ресурсов. При этом HTTP не хранит информацию о своем состоянии: каждая транзакция рассматривается независимо от остальных.

В разрабатываемой системе будет использована иерархическая структура представленная на рисунке 2.12. Такое представление информации хорошо подходит для проработанной, систематизированной и организованной информации. Перемещение по уровням такой структуры осуществляется в направлении от общего к частному.

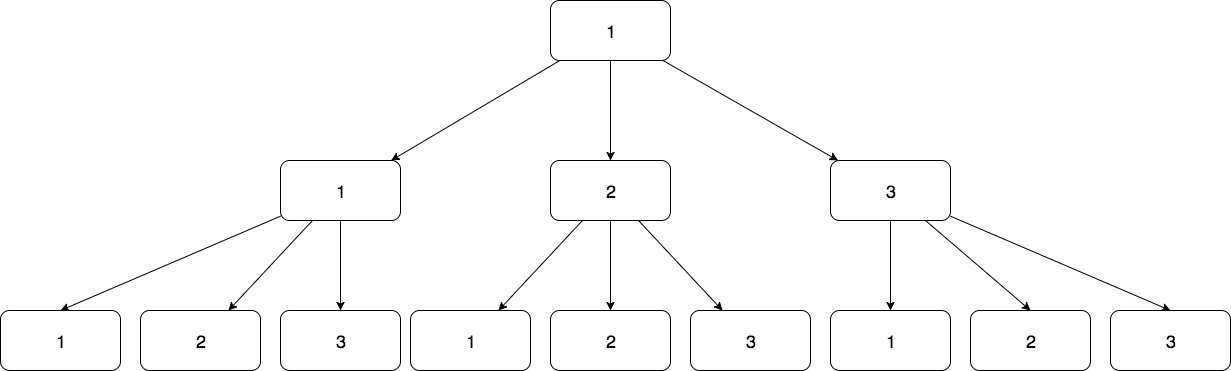


Рисунок 2.12 – Иерархическая структура приложения

Веб-приложение будет состоять из следующих страниц:

* главная страница;
* регистрация;
* авторизация;
  + добавление материала;
  + добавление категории;
  + редактирование материала;
  + удаление материала.

Таким образом, разработанная структура веб-приложения отображает проект информационной структуры сайта и будет использована в дальнейших этапах проектирования и разработки приложения.

2.2.2 Разработка схемы работы системы

Для работы системы необходим ряд алгоритмов, которые обеспечивают защиту и другие функции системы. Одними из важнейших алгоритмов системы является алгоритмы регистрации и авторизации, что позволяют обеспечить защиту веб-сайта от несанкционированного доступа и действий злоумышленника.

Идентификация - процедура распознавания пользователя в системе обычно с помощью заранее определенного имени (идентификатора) или другой априорной информации о нем, которая воспринимается системой. Данный термин обычно означает установление личности пользователя. Аутентификация - процедура установления принадлежности пользователю в системе предъявленного им идентификатора. С помощью аутентификации система убеждается, что субъект действительно тот, за кого себя выдает. Процедура аутентификации пользователя является обязательным этапом функционирования любой современной информационно-коммуникационной системы. Существует несколько методов идентификации и аутентификации, которые отличаются своей сложностью, надежностью, стоимостью и другими показателями. Каждый из этих методов имеет свои положительные и отрицательные стороны, анализу которых посвящена эта работа.

Контроль доступа пользователей – одна из основных задач безопасности любой ИС. Поэтому, важностью частью работы с любой ИС является начальная регистрация и дальнейшая авторизация пользователя. Процесс авторизации включает в себя идентификацию и аутентификацию. Рассмотрим эти понятия.

Идентификация – предоставление при входе в систему своего имени или регистрационного номера, получаемый пользователем при предварительной регистрации в системе.

Аутентификация – подтверждение права на доступ, соответствия предоставленной идентификационной информации пользователя. В настоящее время существует довольно большой перечень протоколов аутентификации, позволяющих защищать как персональные данные, так и конфиденциальную информацию.

В зависимости от типа ресурса для организации доступа к нему могут применяться различные методы аутентификации:

* в случае реализации аутентификации для доступа к системам, содержащим информацию, публикация которой не приведёт к существенным последствиям, рекомендуется следовать минимальным требованиям с использованием многоразовых паролей;
* в случае реализации аутентификации для доступа к системам, содержащим информацию, публикация и модификация которой приведёт к значительному ущербу, рекомендуется использовать строгую аутентификацию в совокупности с одноразовыми паролями;
* в случае реализации аутентификации для доступа к системам, содержащим конфиденциальную информацию, рекомендуется использовать строгую аутентификацию.

Процедура аутентификации является обязательным этапом функционирования любой информационной системы и основным условием доступа пользователя к ресурсам этой системы. Использование web-ориентированных информационных системы также требует идентификации пользователя для предоставления ему прав и доступа в соответствии с его ролевой принадлежностью.

Методы аутентификации условно можно разделить на однофакторные и двухфакторная. Однофакторные методы делятся на:

* логические (пароли, ключевые фразы, которые вводятся с клавиатуры компьютера или клавиатуры специализированного устройства);
* идентификационные (носителем ключевой информации являются физические объекты: дискета, магнитная карта, смарт-карта, штрих-кодовая карта и т. п.);
* биометрические (в их основе - анализ уникальных характеристики человека, например: отпечатки пальцев, рисунок радужной оболочки глаза, голос, лицо).

Надежная идентификация и аутентификация замедляется рядом принципиальных причин.

Во-первых, компьютерная система основывается на информации в том виде, в котором она была получена; строго говоря, источник информации остается неизвестным. Во-вторых, почти все данные аутентификации можно услышать, украсть или подделать. В-третьих, есть противоречия между надежностью аутентификации с одной стороны, и удобствами пользователя и системного администратора с другой. Так, из соображений безопасности необходимо с определенной частотой просить пользователя повторно вводить информацию аутентификации, что как правило не очень удобно для пользователей, а также подвержено риску считыванию данных дополнительными средствами (кейлоггеры, социальная инженерия и др.).

В-четвертых, чем надежное средство защиты, тем он дороже.

Для работы системы необходим ряд алгоритмов, которые обеспечивают защиту и другие функции системы. Одними из важнейших алгоритмов системы является алгоритмы регистрации и авторизации, что позволяют обеспечить защиту веб-сайта от несанкционированного доступа и действий злоумышленника.

Контроль доступа к системе начинается с системы регистрации и авторизации, которые позволяют контролировать доступ к системе. В связи с этим необходимо реализовать способы регистрации пользователей - сотрудников организации в системе и последующие их авторизации для получения доступа к возможностям системы.

Следует отметить, что алгоритм регистрации пользователей является важной частью современного веб-приложения, которое обеспечивает безопасность системы. Регистрация позволяет пользователю внести в систему свои данные, и таким образом, приложение будет выделять пользователя среди других, и предоставлять определенный функционал и возможности для конкретного пользователя в случае авторизации по регистрационным данным. Алгоритм регистрации пользователя на сайте представлен на рисунке 2.13.



Рисунок 2.13 – Алгоритм регистрации пользователя

После регистрации пользователя, алгоритмом, который дает возможность использовать регистрационные данные и авторизоваться на сайте, является алгоритм авторизации пользователей.

Для каждого пользовательского пароля (при его регистрации), создается своя соль и записывается в базу рядом с паролем, чтобы использовать при хешировании и последующем сравнении кеша данных авторизации пользователей. Плюсы алгоритма и метода авторизации рядовых пользователей: снижены требования к паролю пользователей, исключен метод подбора и дешифрования пароля пользователя, отсутствие возможности авторизации через сторонние сервисы, частный подход к алгоритмам шифрования.

При входе в систему появляется форма с полями для ввода персональных данных, состоящих из логина и пароля. После происходит проверка на ввод данных, если есть незаполненные поля, то появляется сообщение о том, что нужно заполнить все имеющиеся поля. Если всё введено, идёт поиск пользователя с такими данными. Если пользователь не найден, осуществляется переход к концу, если найдено совпадение по логину и паролю происходит вход в систему и установление сессии, как показано на рисунке 2.14.



Рисунок 2.14 – Схема алгоритма авторизации в системе

Формирование материалов является одной из основных функций, которая позволяет добавлять материалы на сайт. В связи с этим алгоритм добавления материалов является важной составляющей системы. Пред размещением статьи на сайте нужно авторизоваться на сайте. Появляется форма входа в систему, вводится логин с паролем. После этого осуществляется проверка данных, если данные введены не верно появляется сообщение об ошибке. При правильном вводе происходит вход в систему. После этого появляется возможность перейти в модуль управления статьями, на страницах которого доступна форма с добавлением данных с необходимыми полями для ввода. После ввода данных заказа эти данные при нажатии кнопки «Сохранить» сохраняются в базу данных. При добавлении идет проверка корректности добавления записи, если запись не добавлена, то появляется сообщение об ошибке заполнения полей. Иначе запись успешно добавляется. Схема добавления статей представлена на рисунке 2.15, а схема работы системы на чертеже ДП-1090351602-2020-02.



Рисунок 2.15 – Схема алгоритма добавления записи

2.2.3 Разработка проекта базы данных

Одним из элементов проектирования программного обеспечения является проектирование базы данных, которое включает три этапа: концептуальное, логическое и физическое проектирование. Концептуальный уровень – это сущности, атрибуты, связи. Концептуальная модель – это модель, которая отображает знания в предметной области об ее объектах и их взаимосвязях, процессах и результатах деятельности. На этом этапе могут быть использованы тексты, таблицы, графики, графы, блок-схемы.

Концептуальная модель изучаемой предметной области, должна представлять собой модель, в которой в содержательной форме описывается состав системы, ее компоненты и их взаимосвязи, перечень основных показателей, переменных, как контролируемых, так и неконтролируемых внешних факторов. А также их взаимосвязей с показателями качества системы, перечень решений, которые необходимо определить в результате решения поставленной задачи.

Концептуальное проектирование завершается составлением концептуальной модели, инвариантной к структуре базы данных, которая может быть представлена в виде модели «сущность–связь».

Для изображения концептуальной модели БД в виде графической схемы, представленной на плакате ДП-1090351602-2020-02-ПЛ, разработан ряд специальных языков.

Основными конструктивными элементами модели «сущность-связь» (entity-relationship model, ER-model) являются сущности, их свойства (атрибуты) и связи между сущностями.

Каждая сущность в модели изображается в виде прямоугольника с наименованием (по общепринятому соглашению об именовании сущностей имя сущности должно быть в единственном числе). Атрибуты на диаграммах изображаются в виде овалов, соединенных линиями со своими сущностями. Ключевые атрибуты выделяются на диаграмме подчеркиванием. Связи между сущностями обозначают линиями, которые соединяют прямоугольники соответствующих сущностей. Для сущности, находящейся со стороны «многие», линия связи может заканчиваться значком из трех расходящихся линий.

Логическое проектирование – процесс преобразования требований к данным в структуре данных. На выходе получается СУБД-ориентированная структура базы данных и спецификации прикладных программ. На этом этапе могут моделироваться базы данных относительно различных СУБД и производиться сравнительный анализ моделей.

Логический уровень (представление программиста) это записи, элементы данных, связи между записями.

Большинство современных подходов к проектированию баз данных базируется на использовании разновидностей так называемой ER–модели. Моделирование предметной области основывается на использовании графических диаграмм, которые включают небольшое число компонентов разного рода.

Основными понятиями ER-модели являются сущность, связь и атрибут. Сущность – это реальный или представляемый объект, информация о котором должна быть сохранена и доступна. В диаграммах ER–модели сущность представлена в виде прямоугольника, который содержит имя сущности. Каждый экземпляр сущности должен отличаться от любого другого экземпляра той же сущности. Связь – это графически представляемая ассоциация, которая устанавливается между двумя сущностями. В любой связи выделяются два конца, на каждом из которых должно быть указано имя конца связи, степень конца связи и обязательность связи. Атрибутом сущности является любая деталь, служащая для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности. Имена атрибутов записываются в прямоугольник, который изображает сущность, под именем сущности и изображаются малыми буквами.

Уникальным идентификатором сущности может быть атрибут, комбинация атрибутов, комбинация связей или комбинация связей и атрибутов, которая уникально отличает любой экземпляр сущности от других экземпляров сущности того же типа. Физический уровень (представление администратора) проектирования базы данных, это группирование данных, индексы, методы доступа. Физическая модель данных должна соответствовать описанию конкретной системы управления базами данных, то есть схеме данных.

На рисунках 2.16 – 2.24 представлены модели таблиц базы данных. На рисунке 2.24 представлена логическая модель базы данных.

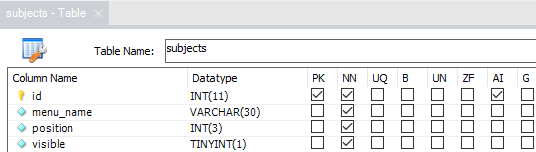


Рисунок 2.16 – Модель таблицы subjects

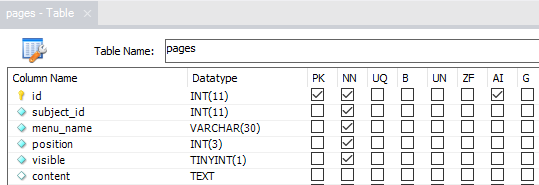


Рисунок 2.17 – Модель таблицы pages

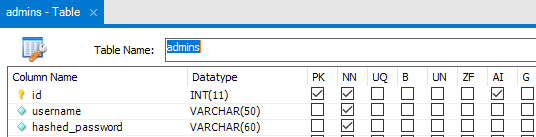


Рисунок 2.18 – Модель таблицы admins

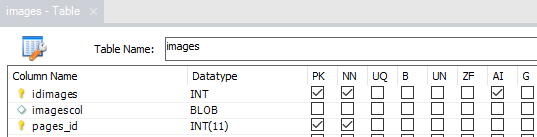


Рисунок 2.19 – Модель таблицы images

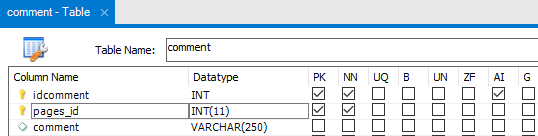


Рисунок 2.20 – Модель таблицы comment

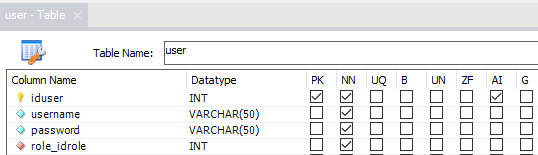


Рисунок 2.21 – Модель таблицы user

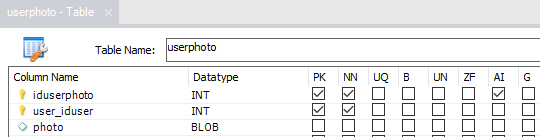


Рисунок 2.22 – Модель таблицы userphoto

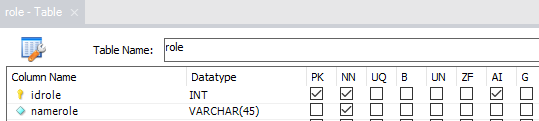


Рисунок 2.23 – Модель таблицы role

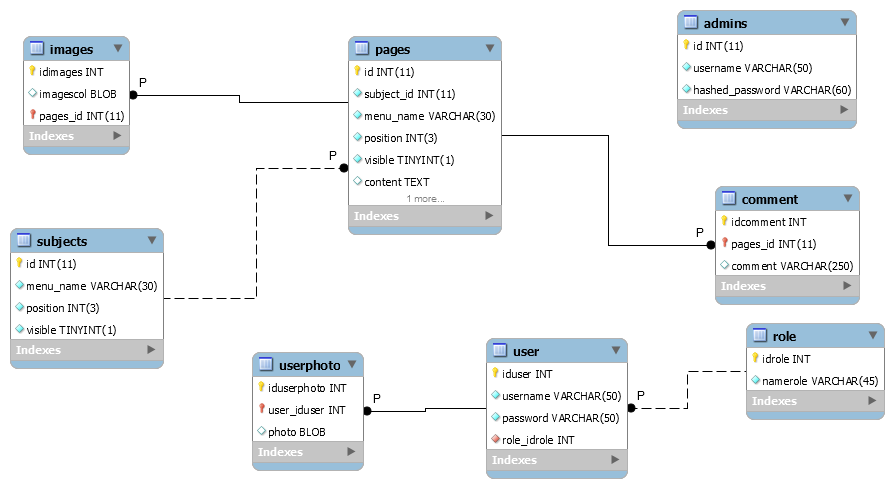


Рисунок 2.24 – Логическая модель базы данных

**2.3 Разработка алгоритмов и программная реализация**

2.3.1 Описание инструментальных средств для реализации основных алгоритмов

В качестве языка программирования веб-приложения выбран JavaScript и его платформа Node.js. Более подробно рассмотрим эти понятия и технологии. Так, язык программирования JavaScript был создан в 1995 году в компании Netscape в качестве языка сценариев в браузере Netscape Navigator 2. Первоначально язык назывался LiveScript, но на волне популярности в тот момент другого языка Java LiveScript был переименован в JavaScript.

Первоначально JavaScript обладал довольно небольшими возможностями. Его цель состояла лишь в том, чтобы добавить немного поведения на веб-страницу. Например, обработать нажатие кнопок на веб-странице, произвести какие-нибудь другие действия, связанные прежде всего с элементами управления. Однако развитие веб-среды, появление HTML5 и технологии Node.js открыло перед JavaScript гораздо большие горизонты. Сейчас JavaScript продолжает использоваться для создания веб-сайтов, только теперь он предоставляет гораздо больше возможностей.

Также он применяется как язык серверной стороны. То есть если раньше JavaScript применялся только на веб-странице, а на стороне сервера надо было использовать такие технологии, как PHP, ASP.NET, Ruby, Java, то сейчас благодаря Node.js можно обрабатывать все запросы к серверу также с помощью JavaScript.

Возможности JavaScript в браузере ограничены ради безопасности пользователя. Цель заключается в предотвращении доступа недобросовестной веб-страницы к личной информации или нанесения ущерба данным пользователя.

Примеры таких ограничений включают в себя:

* JavaScript на веб-странице не может читать/записывать произвольные файлы на жёстком диске, копировать их или запускать программы. Он не имеет прямого доступа к системным функциям ОС. Современные браузеры позволяют ему работать с файлами, но с ограниченным доступом, и предоставляют его, только если пользователь выполняет определённые действия, такие как «перетаскивание» файла в окно браузера или его выбор с помощью тега <input>. Существуют способы взаимодействия с камерой/микрофоном и другими устройствами, но они требуют явного разрешения пользователя. Таким образом, страница с поддержкой JavaScript не может незаметно включить веб-камеру, наблюдать за происходящим и отправлять информацию на внешний сервер;
* различные окна/вкладки не знают друг о друге. Иногда одно окно, используя JavaScript, открывает другое окно. Но даже в этом случае JavaScript с одной страницы не имеет доступа к другой, если они пришли с разных сайтов (с другого домена, протокола или порта). Это называется «Политика одинакового источника» (Same Origin Policy). Чтобы обойти это ограничение, обе страницы должны согласиться с этим и содержать JavaScript-код, который специальным образом обменивается данными. Это ограничение необходимо, опять же, для безопасности пользователя;
* JavaScript может легко взаимодействовать с сервером, с которого пришла текущая страница. Но его способность получать данные с других сайтов/доменов ограничена. Хотя это возможно в принципе, для чего требуется явное согласие (выраженное в заголовках HTTP) с удалённой стороной. Это тоже ограничение безопасности. Подобные ограничения не действуют, если JavaScript используется вне браузера, например — на сервере. Современные браузеры предоставляют плагины/расширения, с помощью которых можно запрашивать дополнительные разрешения.

Распространённость и преимущества JavaScript обеспечены следующими факторами:

* полная интеграция с HTML/CSS.
* простые вещи делаются просто.
* поддерживается всеми основными браузерами и включён по умолчанию.

Как было указано выше, в качестве платформы для создания веб-приложения выбрана Node.js. Node.js представляет среду выполнения кода на JavaScript, которая построена на основе движка JavaScript Chrome V8, который позволяет транслировать вызовы на языке JavaScript в машинный код. Node.js прежде всего предназначен для создания серверных приложений на языке JavaScript. Также существуют проекты по написанию десктопных приложений (Electron) и по созданию кода для микроконтроллеров. Но главным образом Node.js используется в качестве платформы для создания веб-приложений. Node.js является открытым проектом, исходники которого можно посмотреть на github.com.

В Node реализована асинхронная событийно-управляемая платформа для серверных JavaScript-приложений. JavaScript-код выполняется на сервере точно так же, как в клиентском браузере.

Ядро Node.js разработано в соответствии с несколькими принципами, один из которых заключается в предоставлении минимального набора функциональных возможностей, оставляя реализацию всех дополнительных возможностей за модулями экосистемы, не входящими в ядро.

Платформа Node.js использует идею модуля как основное средство структурирования программного кода. Модули являются строительными блоками приложений и библиотек, часто называемых пакетами (термин «пакет» нередко используется взамен термина «модуль», потому что стало обычным делом, когда пакет состоит из единственного модуля). Один из самых активно продвигаемых принципов платформы Node.js заключается в разработке небольших модулей, не только с точки зрения размера, но и, что более важно, с точки зрения охватываемых возможностей.

Этот принцип исходит из философии Unix, в частности из двух следующих ее заповедей:

* чем меньше, тем лучше;
* любая программа должна делать что-то одно, но делать это хорошо.

Платформа Node.js выносит эти идеи на новый уровень. Официальный диспетчер пакетов платформы Node.js помогает решить проблему зависимостей, гарантируя каждому установленному пакету наличие собственного отдельного набора зависимостей, что позволяет программе использовать множество пакетов без появления конфликтов. Подход Node обеспечивает оптимальный уровень повторного использования кода, поскольку при его применении приложения состоят из большого числа малых, четко направленных зависимостей. Хотя это считается непрактичным или даже совершенно не осуществимым в других платформах, для разработки на платформе Node.js рекомендуется именно такая методика. Как следствие npm-пакеты нередко содержат менее 100 строк кода и предназначены для реализации только одной-единственной функции.

Кроме того, естественные преимущества многократно используемых небольших модулей заключаются в следующем:

* простота понимания и использования;
* легкость тестирования и поддержки;
* оптимальность при обмене с браузером.

Наличие небольших, узкоспециализированных модулей обеспечивает возможность совместного многократного использования фрагментов кода. Это позволяет поднять принцип «не повторяйся» (Don’t Repeat Yourself, DRY) на совершенно новый уровень.

Помимо того, что модули Node.js должны иметь небольшой размер и конкретную направленность, они обычно экспортируют минимальный набор функциональных возможностей. Основным преимуществом является повышение удобства и четкости программного интерфейса, что уменьшает вероятность неправильного использования. Как правило, пользователям компонента требуется весьма ограниченный, конкретный набор функций, а не расширение его функциональности или погружение в его особенности.

В платформе Node.js широко используется шаблон определения модулей, экспортирующих только одну функциональную возможность, например функцию или конструктор, при этом доступ к более сложным аспектам или дополнительным возможностям осуществляется через свойства, экспортируемые этой функцией или конструктором. Это помогает пользователю разделить важные и вторичные возможности. Часто модули экспортируют только одну функцию и ничего кроме нее, обеспечивая явную единственную точку входа.

Еще одной характерной чертой многих модулей платформы Node.js является их предназначение для использования, а не для расширения. Сокрытие внутреннего содержания модулей, запрещающее любое его расширение, может показаться лишенным гибкости подходом, но он обеспечивает сокращение вариантов использования, упрощает реализацию, облегчает обслуживание и повышает удобство использования.

Основными составляющим блоками Node.js является шаблон Reactor и библиотека libuv.

Основная суть шаблона Reactor заключается в наличии обработчика (который в Node.js представлен функцией обратного вызова), связанного с любой операцией ввода/вывода, который будет вызван непосредственно при появлении события в цикле событий. Структура шаблона Reactor приводится на рисунке 2.25.

A picture containing screenshot

Description automatically generated

Рисунок 2.25 – Структура шаблона Reactor

Работу приложения, которое использует шаблон Reactor можно представить следующим образом:

* приложение создает новую операцию ввода/вывода, передав запрос демультиплексору событий. Также приложение определяет обработчика для вызова по­ сле завершения операции. Отправка нового запроса демультиплексору событий не приводит к блокировке, управление немедленно возвращается приложению;
* после завершения обработки набора операций ввода/вывода демультиплексор событий добавляет новые события в очередь событий;
* в этом месте цикл событий выполняет обход элементов в очереди событий;
* для каждого события вызывается соответствующий обработчик;
* обработчик, являющийся частью кода приложения, возвращает управление циклу событий (5а). Однако во время выполнения обработчика могут запрашиваться новые асинхронные операции (5b), что приводит к добавлению новых операций в демультиплексор событий (1) до возврата управления циклу событий.

Суть асинхронной обработки заключается в следующем: приложение в некоторый момент времени желает обратиться к ресурсу (без блокировки) и передает обработчика, который̆должен быть вызван некогда в будущем, после завершения операции.

Приложение на платформе Node.js завершится автоматически, когда в демультиплексоре событий не останется отложенных операций и событий в очереди. Шаблон Reactor в Node.js обеспечивает обработку операций ввода/вывода, блокируя выполнение до момента доступности новых событий из набора наблюдаемых ресурсов с последующей обработкой каждого события вызовом связанного с ним обработчика.

Библиотека libuv, обеспечивает совместимость Node.js со всеми основными платформами и нормализующую неблокирующую модель поведения для различных типов ресурсов. На настоящий момент библиотека libuv является низкоуровневым движком ввода/вывода платформы Node.js.

Помимо абстрагирования основных системных вызовов, библиотека libuv реализует шаблон Reactor, обеспечивая программный интерфейс для создания циклов событий, управления очередью событий, выполнения асинхронных операций ввода/ вывода и организации очереди заданий разных типов.

Для построения полной платформы Node.js кроме вышеперечисленных элментов необходимы еще три компонента:

* набор привязок, ответственных за обертывание и использование библиотеки libuv, а также других низкоуровневых JavaScript-функций;
* JavaScript-движок V8, изначально разработанный компанией Google для браузера Chrome. Он является одной из причин быстроты и эффективности Node.js. ДвижокV8 отличается революционным дизайном, высокой скоростью и эффективным управлением памятью;
* ядро JavaScript-библиотеки, реализующее высокоуровневый программный интерфейс Node.js.

Полная архитектура Node.js представлена на рисунке 2.26.

A screenshot of a cell phone

Description automatically generated

Рисунок 2.26 – Архитектура платформы Node.js

Для разработки под Node JS достаточно простейшего текстового редактора, при этом для более удобного процесса разработки больше подходит среда разработки, одной из которых является WebStorm. Указанная программа является интегрированной средой разработки с интеллектуальным редактором, которая глубоко понимает код, обеспечивает автодополнение кода, рефакторинг, а также предотвращает ошибки в процессе работы и поддерживает смешивание языков.

В программе поддерживаются передовые технологии веб-разработки, включая HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMA Script Harmony, шаблоны Jade, ZenCoding, Emmet, JavaScript.

Основные функции WebStorm: Программное обеспечение JetBrains WebStorm представляет собой инструмент для разработки web-сайтов и редактирования HTML, CSS и JavaScript кода. Решение обеспечивает быструю навигацию по файлам и генерирует уведомления о возникающих проблемах в коде в режиме реального времени. JetBrains WebStorm позволяет добавлять разметку HTML-документов или элементов SQL непосредственно в JavaScript. JetBrains WebStorm осуществляет развертывание и синхронизацию проектов через протокол FTP.

Используя возможности кода HTML/XHTML и XML, WebStorm обеспечивает автоматическое завершение стилей, ссылок, атрибутов и других элементов кода. При работе с CSS осуществляется завершение кода классов, HTML-номеров, ключевых слов и т. д. WebStorm предлагает автоматическое решение таких проблем, как выбор формата, свойств, классов, ссылок на файлы и других атрибутов CSS. Решение позволяет использовать мощность инструмента Zen coding для верстки HTML, отображает действия тэга на web-странице. Продукт WebStorm осуществляет завершение кода JavaScript для ключевых слов, лейблов, переменных, параметров и функций DOM и поддерживает специфические особенности популярных браузеров. Реализованные в решении функции рефакторинга JavaScript позволяют преобразовывать структуру кода и файлов и .js.

WebStorm обеспечивает отладку кода JavaScript и предоставляет широкий диапазон возможностей: нахождение точки останова в HTML и JavaScript, настройка параметров точки останова, тестирование синтаксиса кода в режиме реального времени и т. д. Продукт поддерживает платформы JQuery, YUI, Prototype, DoJo, MooTools, Qooxdoo и Bindows. WebStorm предусматривает интегрированную проверку текста на тэги, последовательность кода, ошибки в написании и т. д. WebStorm позволяет редактировать файлы и автоматически синхронизировать их по требованию при удаленной работе или хранении [16].

Продукт поддерживает функцию контроля версий и предварительных вариантов кода и фиксирует все произведенные действия и изменения. Благодаря созданию истории в WebStorm можно восстанавливать кодовые выражения, блоки и даже целые файлы.

Новые возможности WebStorm:

* поддержка фреймворков Meteor и React;
* интеграция PhoneGap, Apache Cordova и IonicFramework для разработки мобильных приложений;
* поддержка Polymer;
* интеграция Gulp;
* инструмент spy-js для трассировки и профилирования приложений Node.js, автозавершения кода и обнаружения проблемных фрагментов;
* поддержка модуля EcmaScript 6;
* автоматическое обновление приложений Node.js;
* проверка стилистики кода с помощью ESLint и JSCS;
* встроенная поддержка плагина EditorConfig, позволяющего автоматически определять и выбирать конфигурации стиля кода для различных IDE и редакторов;
* усовершенствования CSS3;
* рostfix-шаблоны для JavaScript;
* новая функция для отладки приложений, отображающая значение переменных непосредственно в редакторе;
* функция Scratch Files для создания набросков непосредственно в редакторе без изменения проекта или создания каких-либо файлов;
* поддержка последних версий Stilus и Sass.

Для работы приложения необходима базы данных. База данных - структурированный упорядоченный набор данных. Система управления базами данных (СУБД) - программа, предназначенная для организации и ведения базы данных. Основные функции СУБД — это описание структуры базы данных, обработка данных и управление данными.

В таблице 2.5 приведена сравнительная таблица четырех систем управления базами данных, конкурирующих на рынке программного обеспечения по основным показателям.

Таблица 2.5 – Сравнение СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерий | Система управления базой данных | | | |
| IBM DB2 | Informix | Microsoft SQL Server | MySQL |
| Моделирование данных | 4 | 3 | 3 | 5 |
| Особенности архитектуры и функционал | 4 | 2 | 4 | 5 |
| Контроль работы системы | 3 | 4 | 4 | 5 |
| Производительность | 5 | 2 | 4 | 5 |
| Особенности разработки | 4 | 2 | 5 | 5 |
| Надежность | 5 | 3 | 4 | 5 |
| Требования к рабочей среде | 3 | 3 | 4 | 4 |
| Смешанные критерии | 4 | 1 | 5 | 5 |
| Итого: | 32 | 20 | 33 | 39 |

Таким образом, для проекта, рассматриваемого в данном дипломном проекте наиболее приемлема СУБД MySQL. Более подробно опишем СУБД MySQL. Это реляционная БД, дающая с высокой степени удобства и надежности возможность хранения информации. MySQL является высокопроизводительной многопоточной и многопользовательской системой управления реляционными БД, основанной на архитектуре «клиент–сервер». За последнее время система управления БД явилась стандартом для применения в личной и деловой сфере во многом за счет ее продвинутого набора инструментов для управления данными, мягкой политики лицензирования, а также общемировой поддержки со стороны сообщества, в которую входят обычные пользователи и серьезные разработчики.

Популярность MySQL объясняет определенное сочетание уникальных особенностей, в число которых входят надежность, скорость, расширяемость, а также открытость исходного кода. Схема данных БД представлена на плакате – ДП-1090351602-2020-03-ПЛ.

2.3.2 Реализация модели базы данных

После создания логической модели базы данных был сгенерирован и выполнен не сервере MySQL запрос на создание физических моделей, представленных выше таблиц.

CREATE TABLE IF NOT EXISTS `newsartsek`.`admins` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(50) NOT NULL,

`hashed\_password` VARCHAR(60) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`subjects` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`menu\_name` VARCHAR(30) NOT NULL,

`position` INT(3) NOT NULL,

`visible` TINYINT(1) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`pages` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`subject\_id` INT(11) NOT NULL,

`menu\_name` VARCHAR(30) NOT NULL,

`position` INT(3) NOT NULL,

`visible` TINYINT(1) NOT NULL,

`content` TEXT NULL DEFAULT NULL,

`subjects\_id` INT(11) NOT NULL,

PRIMARY KEY (`id`),

INDEX `subject\_id` (`subject\_id` ASC),

INDEX `fk\_pages\_subjects1\_idx` (`subjects\_id` ASC),

CONSTRAINT `fk\_pages\_subjects1`

FOREIGN KEY (`subjects\_id`)

REFERENCES `newsartsek`.`subjects` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`images` (

`idimages` INT NOT NULL AUTO\_INCREMENT,

`imagescol` BLOB NULL,

`pages\_id` INT(11) NOT NULL,

PRIMARY KEY (`idimages`, `pages\_id`),

INDEX `fk\_images\_pages\_idx` (`pages\_id` ASC),

CONSTRAINT `fk\_images\_pages`

FOREIGN KEY (`pages\_id`)

REFERENCES `newsartsek`.`pages` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`comment` (

`idcomment` INT NOT NULL AUTO\_INCREMENT,

`pages\_id` INT(11) NOT NULL,

`comment` VARCHAR(250) NULL,

PRIMARY KEY (`idcomment`, `pages\_id`),

INDEX `fk\_comment\_pages1\_idx` (`pages\_id` ASC),

CONSTRAINT `fk\_comment\_pages1`

FOREIGN KEY (`pages\_id`)

REFERENCES `newsartsek`.`pages` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`role` (

`idrole` INT NOT NULL AUTO\_INCREMENT,

`namerole` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idrole`))

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`user` (

`iduser` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(50) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`role\_idrole` INT NOT NULL,

PRIMARY KEY (`iduser`),

INDEX `fk\_user\_role1\_idx` (`role\_idrole` ASC),

CONSTRAINT `fk\_user\_role1`

FOREIGN KEY (`role\_idrole`)

REFERENCES `newsartsek`.`role` (`idrole`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`userphoto` (

`iduserphoto` INT NOT NULL AUTO\_INCREMENT,

`user\_iduser` INT NOT NULL,

`photo` BLOB NULL,

PRIMARY KEY (`iduserphoto`, `user\_iduser`),

INDEX `fk\_userphoto\_user1\_idx` (`user\_iduser` ASC),

CONSTRAINT `fk\_userphoto\_user1`

FOREIGN KEY (`user\_iduser`)

REFERENCES `newsartsek`.`user` (`iduser`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

После выполнения запроса на сервере создана база данных, структура которой представлена на рисунке 2.27.

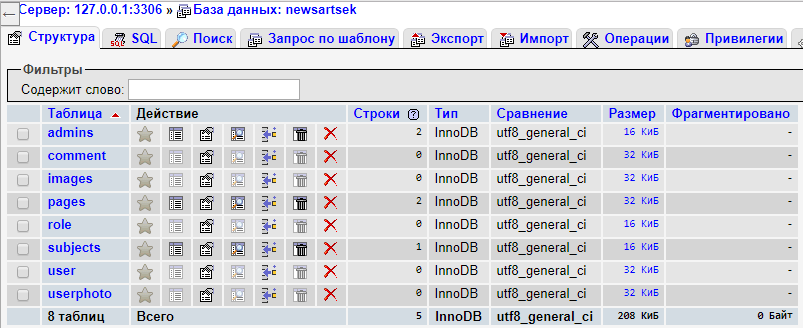


Рисунок 2.27 – Структура базы данных

2.3.3 Обоснование и разработка интерфейса

Для того, чтобы придать сайту соответствующий вид, интерфейс, мы использовали, так называемые, каскадные таблицы стилей, в частности CSS-Фреймворк-Bootstrap. CSS представляет собой официальный язык, который описывает внешнюю структуру и вид документа, написанный с применением языка разметки HTML.

Bootstrap представляет собой интуитивно-понятный и многофункциональный при этом интерфейс, который повышает скорость и облегчает создание web-приложений. На рисунке 2.28 можно наглядно увидеть, как выглядит файловая структура Bootstrap.

A screenshot of a cell phone

Description automatically generated

Рисунок 2.28 – Файловая структура Bootstrap

На рисунке представлена главная форма Bootstrap: скомпилированные файлы можно использовать в любом web-проекте. Существуют сборки CSS и JS (bootstrap.\*) и самый минимизированный вариант (bootstrap.min.\*). Как дополнительная функция в Bootstrap интегрированы шрифты Glyphicons.

Исходный код Bootstrap состоит из перекомпилированных файлов CSS, JavaScript и шрифтов, включая исходных код Less, JavaScript и соответствующие документы. Его пример представлен на рисунке 2.29.

A close up of a logo

Description automatically generated

Рисунок 2.29 – Схема исходного кода Bootstrap

less/, js/, шрифты/ с иконками. Папка dist / содержит – вышеперечисленные скомпилированные файлы, которые указаны на раздел ранее. Папка docs/ состоит из документов и примеров / применения Bootstrap, представлено на рисунке 2.30, а работа над макетом сайта представлена на рисунке 2.31. Также в ней содержатся сведения о лицензиях и предшествующих версиях.

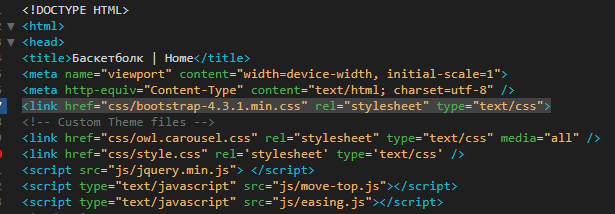


Рисунок 2.30 – Подключение bootstrap к проекту

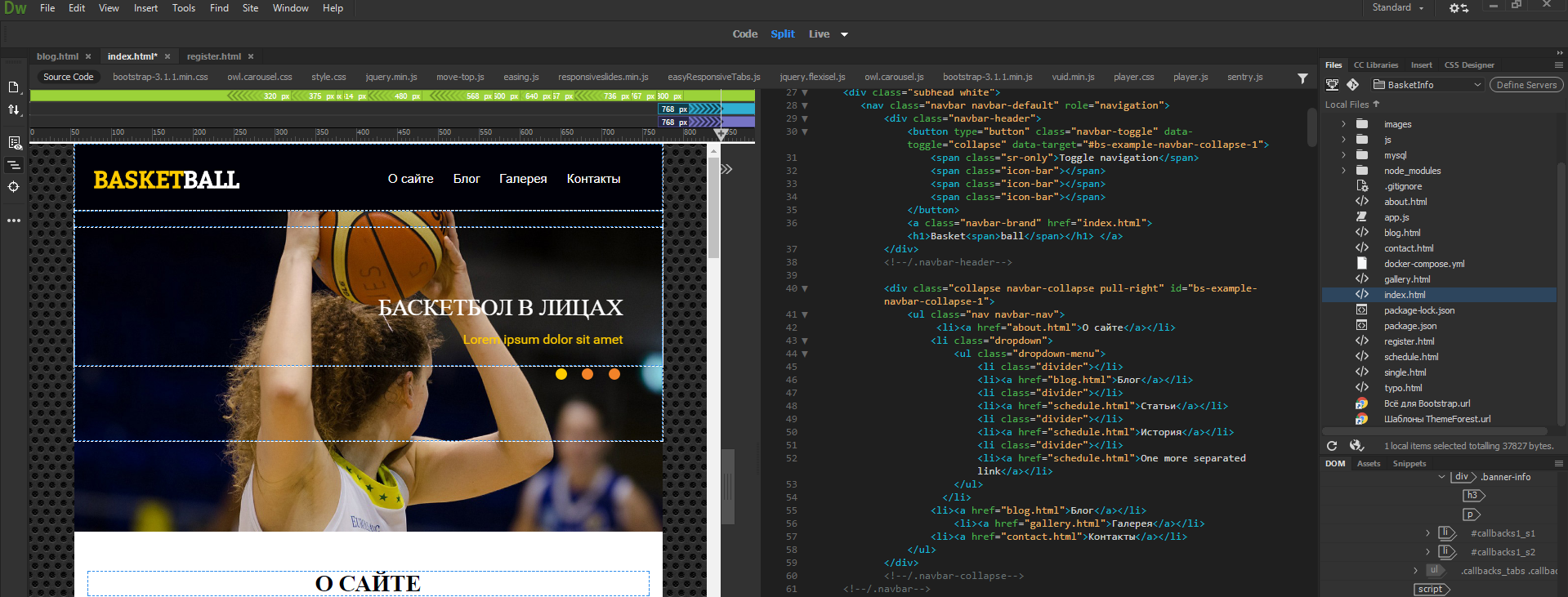


Рисунок 2.31 – Работа над макетом сайта

2.3.4 Отладка и тестирование разрабатываемого программного обеспечения

В проектах по разработке программного обеспечения помимо основной задачи по реализации заявленной функциональности существует не менее важная задача по обеспечению качества ПО. Качество ПО (Software quality) — это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

Процесс тестирования позволит определить скорость загрузки контента, качество адаптивности верстки, отказоустойчивость ресурсов под нагрузкой, метрику кода и корректность работы информационной системы.

Test Cases — это набор условий, при которых тестировщик будет определять, удовлетворяется ли заранее определённое требование. Чтобы определить, что требование полностью выполняется, может потребоваться много вариантов тестирования. Часто варианты тестирования группируют в тестовые наборы. UnitTest - процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок. В таблице 8 представлены результаты тестов проверки работы разработанного мобильного приложения.

Тестирование – составная часть отладки. Отладка имеет место тогда, когда сайт со всей очевидностью работает неправильно. Если же оказывается, что сайт работает, верно, то она тестируется. Часто случается так, что после прогона тестов сайта вновь подвергается отладке. Таким образом, тестирование устанавливает факт наличия ошибки, а отладка выявляет ее причину.

Отладка — это процесс поиска и устранения ошибок. Ошибки в программе разделяют на две группы: синтаксические (ошибки в тексте) и алгоритмические. Синтаксические ошибки — это наиболее легко устраняемые ошибки. Алгоритмические ошибки обнаружить труднее. Этап отладки можно считать законченным, если программа правильно работает на одном-двух наборах входных данных.

На этом этапе выявляются возможные ошибки, допущенные на предыдущих этапах. Синтаксические ошибки в тексте программы автоматически идентифицируются еще на этапе трансляции, и пользователь вносит соответствующие изменения в текст. Может случиться так, что текст программы записано, верно, а ошибка допущена при составлении алгоритма – программа работает, но выдает неправильные результаты. Если же ошибка допущена на этапе постановки задачи, то программа работает правильно, но решает другую задачу.

Итак, целью отладки является получение правильной программы, результатам работы которой можно было бы доверять.

Суть отладки заключается в том, что пользователь разрабатывает систему тестов, с помощью которой проверяется работа программы в различных возможных режимах. Каждый тест имеет набор входных данных, для которых известен результат. Тест стараются выбрать так, чтобы не только установить сам факт ошибки, но и локализовать ее, то есть выявить и сузить часть программы, содержащей ошибку.

В состав систем программирования включают специальные возможности отладки программ. Пользователь формулирует задачу, а система выполняет эту задачу и выдает пользователю необходимую информацию о том, как ведет себя программа. Такая информация значительно облегчает поиск и исправление ошибок. В таблице 2.6 представлены сведения проверки работоспособности системы.

Таблица 2.6 – Проверка работоспособности системы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Модуль | Подмодуль | Описание теста | Ожидаемый результат | Прохождение теста |
| 1 | Модуль авторизации | Авторизация в приложении | Вход с некорректными данными:  1.Ввести некорректные логин и пароль  2. Нажать кнопку ‘Войти’ | Отображается сообщение о том, что логин или пароль введён неверно | Тест пройден |
| 2 | Модуль авторизации | Авторизация в приложении | 1.Ввести корректные логин и пароль  2. Нажать кнопку ‘Войти’ | Открывается страница с выбором тестов | Тест пройден |
| 3 | Модуль оформления ститьи | Окно добавления статьи | Переход на страницу добавления статьи | Открытие окна с формой внесения данных статьи | Тест пройден |

Конфигурационное тестирование. При конфигурационном тестировании определено, что созданная программа успешно функционирует в различных браузерах под управлением разных операционных систем представлено в таблице 2.7.

Таблица 2.7 – Работа приложения в разных браузерах

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **Google Chrome** | **Mozilla Firefox** | **Safari** |
| Открытие главного модуля программы | + | + | + |
| Отображение меню программы | + | + | + |
| Выбор модуля программы | + | + | + |
| Отображение списка элементов из выбранного модуля | + | + | + |
| Добавление записей в выбранном модуле | + | + | + |
| Редактирование записей в выбранном модуле | + | + | + |
| Удаление записей в выбранном модуле | + | + | + |

При проверке удобства использования, определено, что программа понятна и удобна пользователя, при этом отвечает следующим параметрам:

* быстрое открытие модулей программы;
* подходящее стилевой оформление;
* удобное и понятное расположение пунктов навигации по системе;
* однотипные шрифты во всех модулях системы;
* отсутствие отвлекающих блоков и информации;
* удобное управление записями в модулях системы.

2.4 Описание применения программного обеспечения

Созданное приложение предназначено для знакомства пользователей с баскетболом Беларуси. Карта пользователя представлена на плакате

ДП-1090351602-2020-04-ПЛ.

На главной странице, представленной на рисунке 2.32, можно узнать общую информацию о баскетболе, где он зародился, о ведущих клубах страны.

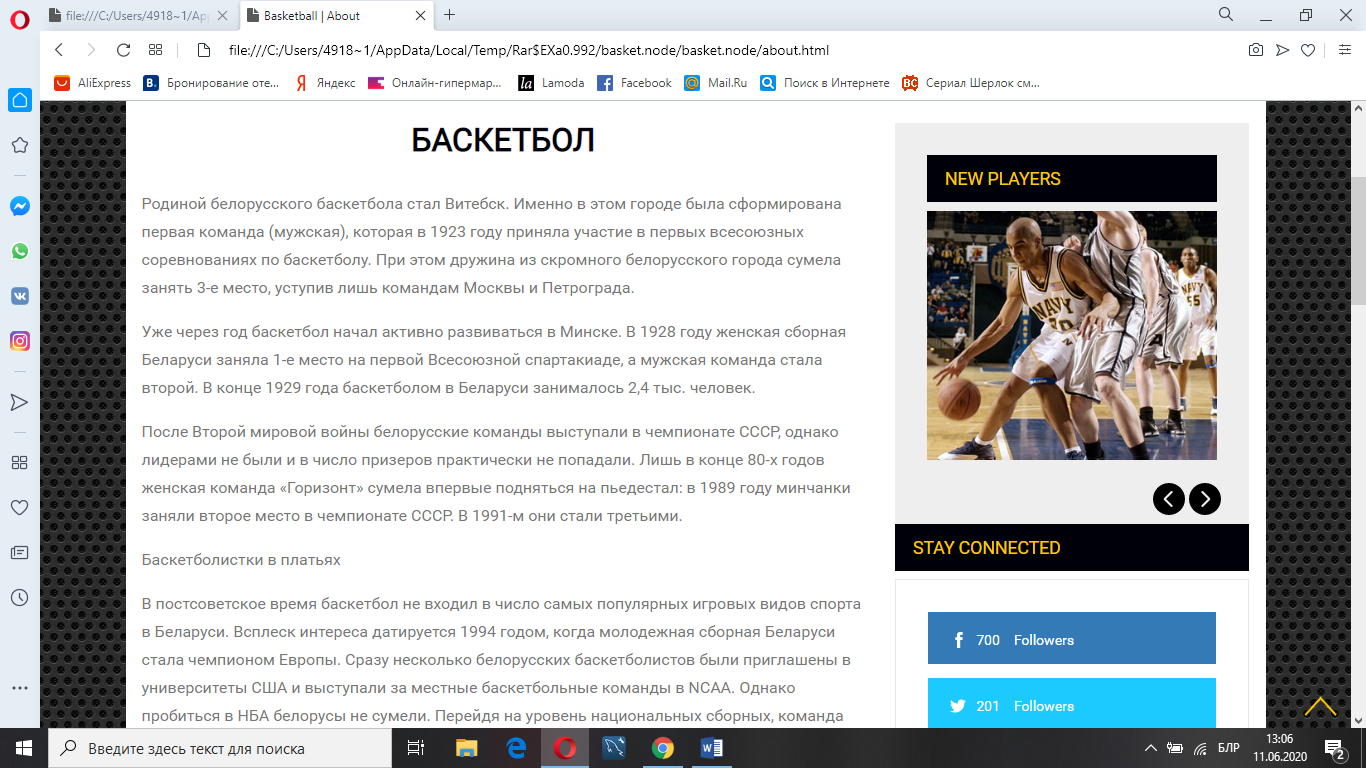


Рисунок 2.32 – Главная страница сайта

Для того, чтобы посмотреть информацию о выдающихся баскетболистах Беларуси нужно перейти на вкладку «Личности». На этой вкладке мы можем увидеть информацию о таких белорусских баскетболистах, как Владимир Веремеенко, Александр Кудрявцев и Артем Параховский, в каких клубах они выступали, их статистику и т.д., представлено на рисунках 2.33-2.35.



Рисунок 2.33 – Вкладка «Личности»

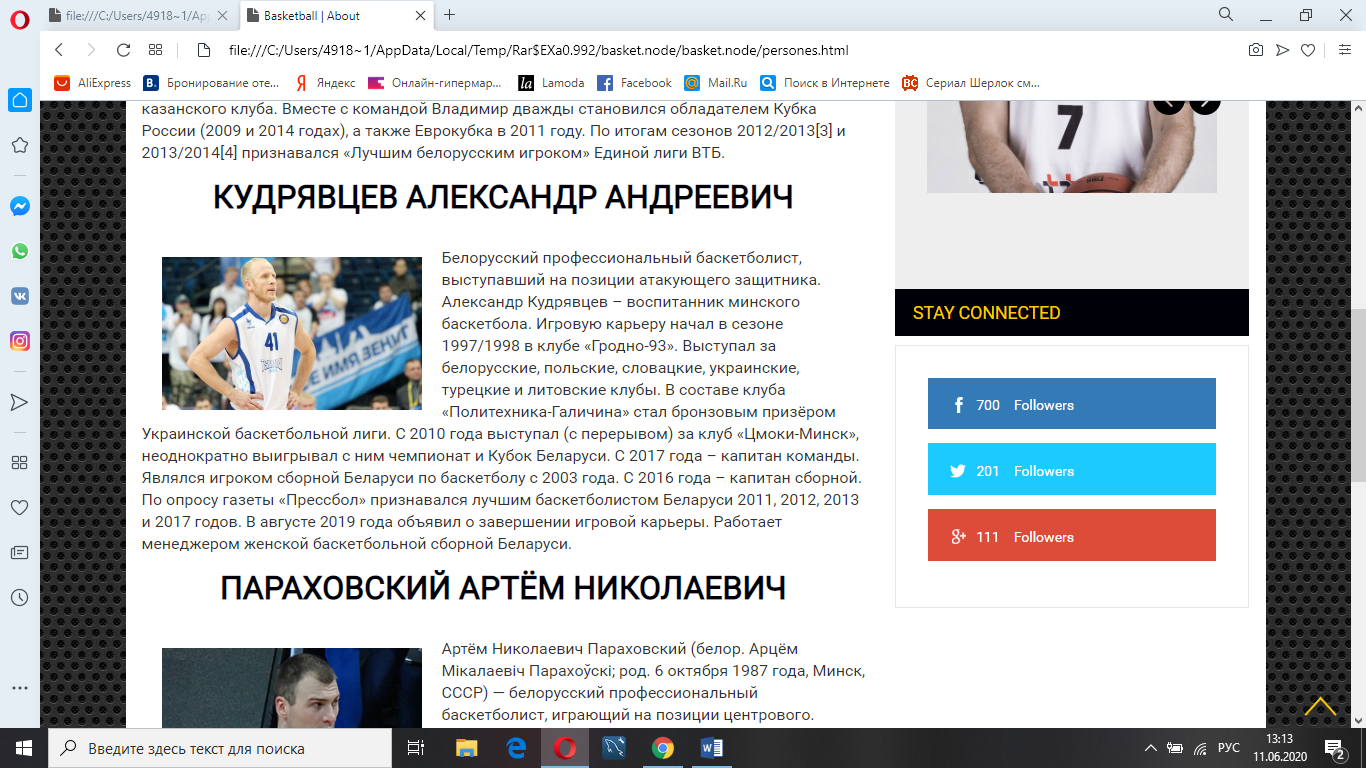


Рисунок 2.34 – Вкладка «Личности»

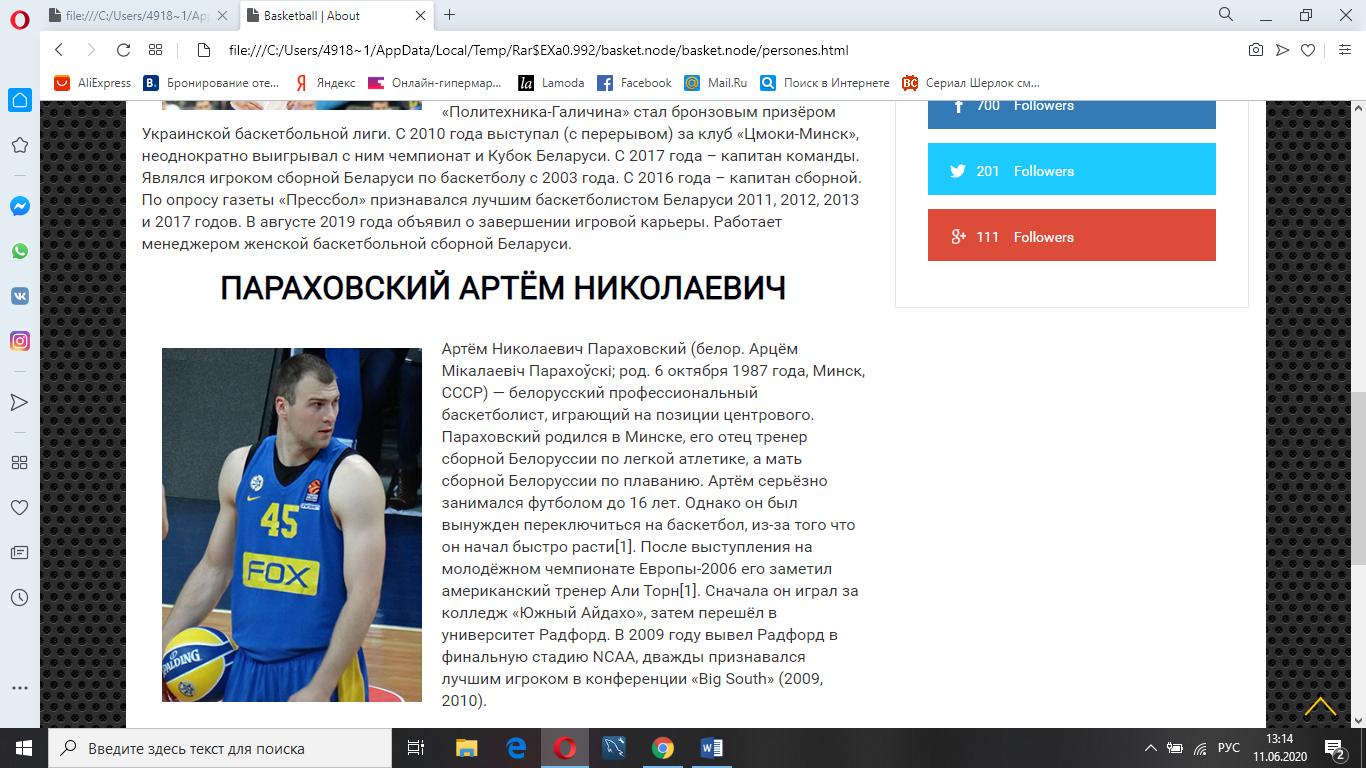


Рисунок 2.35 – Вкладка «Личности»

Так же на этой вкладке можно найти разного рода фотографии с баскетбольных матчей, представлено на рисунке 2.36.



Рисунок 2.36 – Фотографии с матчей

**3 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**3.1 Оценка конкурентоспособности объекта проектирования**

В экономической части дипломного проекта будет оценена экономическая целесообразность проектирования программного продукта web-сайта «Баскетбол Беларуси в лицах».

На начальном этапе будет проведен конкурентный анализ, путем сравнения проектируемого ПП с аналогичным web-сайтом «Цмоки-Минск».

Для оценки конкурентоспособности в результате анализа были выявлены следующие технические показатели: количество выполняемых функций, количество вспомогательных функций, время обработки запроса, количество переходов для завершения операции, количество вводимой информации. В конкурентоспособность также входят экономические параметры, которые присутствуют в программном продукте, а именно: цена приобретения, среднегодовые затраты на эксплуатацию.

Аналитическая таблица 3.1 заполнена по картам технического уровня и используется для сравнительных расчетов. Если для показателей не применена количественная оценка, то значения показателей определяются бальным методом (от 0 до 10).

Таблица 3.1. – Выбранные характеристики для сравнительных расчетов

|  |  |  |  |
| --- | --- | --- | --- |
| Показатели | Весомость показателя, % | Варианты | |
| Базовый | Проектный |
| Технические параметры | | | |
| Выполнение основных функций | 0,3 | 2 | 5 |
| Наличие вспомогательных функций | 0,2 | 1 | 3 |
| Время обработки запроса | 0,2 | 0,3 | 0,1 |
| Количество переходов для завершения операции | 0,2 | 2 | 3 |
| Количество вводимой информации | 0,1 | 2 | 5 |
| Экономические параметры | | | |
| Цена приобретения без НДС, руб |  | 520,10 | 543,35 |
| Среднегодовые затраты по эксплуатации, руб |  | 89,81 | 76,92 |
| Цена потребления при сроке службы 3 года, руб |  | 789, 53 | 774,11 |

На основе единичных показателей определяются групповой показатель конкурентоспособности с коэффициентом весомости, которые были установлены на предыдущем этапе, расчет Iтех производится по формуле (3.2):

Iтех = ∑ qтехi × aтехi, (3.2)

где qтехi– единичный показатель;

aтехi – единичный показатель с учетом весовой доли.

Рассчитать групповой показатель конкурентоспособности программы по экономическим параметрам Iэк можно по формуле (3.3):

Iэк = Ц/Цб, (3.3)

где Ц – значение цены проектного варианта, руб;

Цб – значение цены базового варианта, руб.

На основе групповых показателей конкурентоспособности рассчитывается интегральный показатель по формуле (3.4):

К = Iн.п × Iтех/Iэк, (3.4)

Результаты расчета единичных и групповых показателей представлены в таблице 3.2.

Таблица 3.2 – Единичные показатели конкурентоспособности проектируемого изделия

|  |  |  |  |
| --- | --- | --- | --- |
| Показатели | Весомость показателя, % | Единичный показатель | С учетом весовой доли |
| Технические параметры | | | |
| Выполнение основных функций | 0,3 | 5/2=2,5 | 0,75 |
| Наличие вспомогательных функций | 0,2 | 3/1=3 | 0,6 |
| Время обработки запроса | 0,2 | 0,1/0,3=0,33 | 0,06 |
| Количество переходов для завершения операции | 0,2 | 3/2=1,5 | 0,3 |
| Количество вводимой информации | 0,1 | 5/2=2,5 | 0,25 |
| Групповой показатель по техническим параметрам Iтех | | | 1,96 |
| Экономические параметры | | | |
| Групповой показатель по экономическим параметрам Iэк | | | 520,10/543,35=0,95 |
| Комплексный показатель К | | | 2,02 |

Расчет единичного показателя с учетом весовой доли:

Iтех1 = 2,5 × 0,3 = 0,75

Iтех2 = 3 × 0,2 = 0,6

Iтех3 = 0,33 × 0,2 = 0,06

Iтех4 = 1,5 × 0.2 = 0,3

Iтех5 = 2,5 × 0,1 = 0,25

Iтех = 0,75 + 0,6 + 0,06 + 0,3 + 0,25 = 1,96

Расчет комплексного показателя:

К = 1 × 1,96/0,97 = 2,02

Если комплексный показатель К больше 1, то проектируемый объект превосходит базовый вариант, если меньше 1 – уступает, если  
К = 1 – находится на одинаковом уровне.

По результатам проведенного расчета конкурентоспособности продукции с целью графической иллюстрации строится радар конкурентоспособности, который представлен на рисунке 3.1.

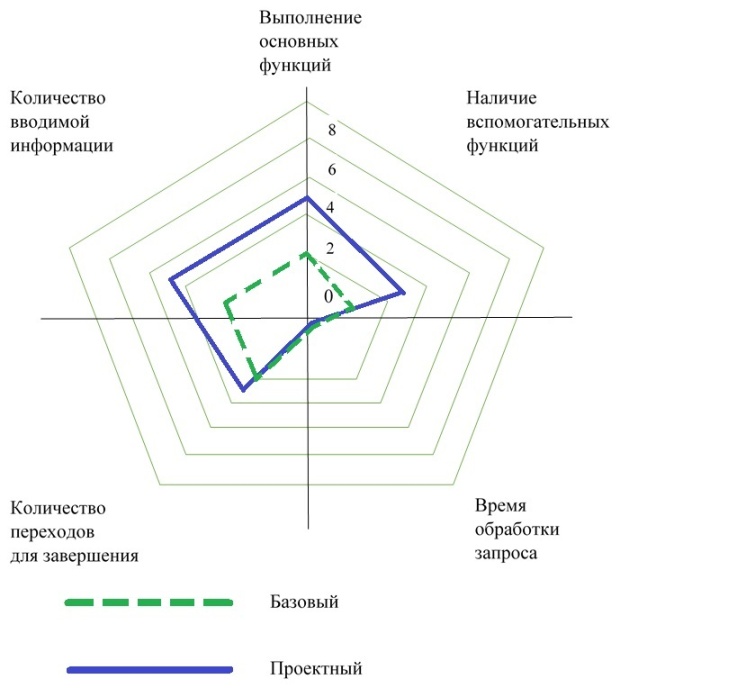


Рисунок 3.1 – Радар конкурентоспособности

Таким образом, так как комплексный показатель К больше 1, то проектируемый объект превосходит базовый и можно рекомендовать его производство.

**3.2 Определение единовременных затрат на создание программного обеспечения**

Единовременные капитальные затраты представляют собой цену программного продукта. Все расчеты между покупателем и продавцом продукции, к числу которой относят и программные продукты, производятся на основе отпускных цен.

Определяющим фактором отпускной цены разработки является трудоемкость создания программного продукта.

Предусмотрено четыре степени новизны разрабатываемых задач:

* разработка задач, предусматривающая применение принципиально новых методов разработки, проведение научно-исследовательских работ;
* разработка типовых проектных решений, оригинальных задач и систем, не имеющих аналогов;
* разработка проекта с использованием типовых проектных решений при условии их изменения; разработка проектов, имеющих аналогичные решения.

Привязка типовых проектных решений.

Сложность алгоритма представлена тремя группами:

– алгоритмы оптимизации и моделирования систем и объектов;

– алгоритмы учета, отчетности, статистики и поиска;

– алгоритмы, реализующие стандартные методы решения, а также не предусматривающие применения сложных численных и логических методов.

Трудоемкость создания программного продукта определяется укрупненным методом. При этом необходимо воспользоваться формулой (3.5):

Трз = Тоа + Тбс + Тп + Тотл + Тдр + Тдо, (3.5)

где Тоа – трудоемкость подготовки описания задачи и исследования алгоритма решения, руб;

Тбс – трудоемкость разработки блок-схемы алгоритма, час;

Тп – трудоемкость программирования по готовой блок-схеме, час;

Тотл – трудоемкость отладки программы на ЭВМ, час;

Тдр – трудоемкость подготовки документации задачи в рукописи, час;

Тдо – трудоемкость редактирования, печати и оформления документации по задаче, час.

Трзп = 5,06 + 16,87 + 16,87 + 84,37 + 16,87 + 11,80 = 151,84 час

Составляющие приведенной формулы определяются, в свою очередь, через условное число операторов Q в разрабатываемом программном продукте по формуле (3.6):

Q = q × C × (1 + p), (3.6)

где q – число операторов в программе;

С – коэффициент сложности программы;

p – коэффициент коррекции программы в ходе ее разработки.

Qп = 280 × 3 × 0,3 = 270

Коэффициент сложности программы *С* характеризует относительную сложность программ задачи по отношению к так называемой типовой задаче, сложность которой принята за единицу. Значение коэффициента определяется на базе экспертных оценок [18].

Коэффициент коррекции программ *p* характеризует увеличение объема работ за счет внесения изменений в алгоритм и программу, изменения состава и структуры информации, а также уточнений, вносимых разработчиком программы для улучшения ее качества без изменения постановки задачи. Значение *p* может быть принято равным 0,15...0,5.

Составляющие трудоемкости разработки программы определятся по формулам (3.7,3.8,3.9,3.10,3.11,3.12):

Тоа = Q × W/((75...85) × K), (3.7)

Тбс = Q/((20…25) × K) (3.8)

Тп = Q/((20...25) × K), (3.9)

Тотл = Q/((4…5) × K), (3.10)

Тдр = Q/((15...20) × K), (3.11)

Тдо = 0.75 × Тдр, (3.12)

где W – коэффициент увеличения затрат труда вследствие недостаточного или некачественного описания задачи (W = 1.2...1.5);

К – коэффициент квалификации разработчика алгоритмов и программ (при стаже работы до двух лет К = 0.8, при стаже от двух до трех лет К = 1.0, при стаже от трех до пяти лет К = 1,1...1,2, при стаже от пяти до семи лет К = 1,3...1,4, при стаже свыше семи лет К = 1,5...1,6.).

Результат составляющих трудоемкости:

Проектный вариант:

Тоап = 270 × 1,2/(80 × 0,8) = 5,06 час

Тбсп = 270/(20 × 0,8) = 16,87 час

Тпп = 270/(20 × 0,8) = 16,87 час

Тотлп = 270/(4 × 0,8) = 84,37 час

Тдрп = 270/(20 × 0,8) = 16,87 час

Тдоп = 0,75×11,81 = 11,80 час

Для определения себестоимости создания программного продукта необходимо определить затраты на заработную плату разработчика по формуле (3.13):

Зрз = Трз × tчр × (1 + q) × (1 + a) × (1 + b), (3.13)

где Трз – трудоемкость разработки программного продукта, чел-ч.;

tчр – среднечасовая ставка работника, осуществлявшего разработку программного продукта, руб.

q – коэффициент, учитывающий процент премий в организации-разработчике (при отсутствии данных может быть принят 0,3...0,4), руб.

а – коэффициент, учитывающий дополнительную заработную плату (при отсутствии данных может быть принят 0,15), руб.

b – коэффициент, учитывающий отчисления от фонда заработной платы (отчисления в фонд социальной защиты населения и отчисления на обязательное медицинское страхование от несчастных случаев). Численное значение коэффициента b = 0,346), руб.

Зрзп = 151,84×1,34×(1+0,3)×(1+0,15)×(1+0,346) = 335,7 руб.

Среднечасовая ставка работника определяется исходя из Единой тарифной системы оплаты труда в Республике Беларусь по следующей формуле:

tчр  = (БС × kт) / 170, (3.14)

где БС ­­­– базовая ставка в РБ в 2020 году составляет 185 руб;

kт – тарифный коэффициент работника соответствующего разряда;

170 – среднее нормативное количество рабочих часов в месяце для 2020 года.

tчр = (185×1,27)/170 = 1,34

В себестоимость разработки программного продукта включаются также затраты на отладку программного продукта в процессе его создания. Для определения их величины необходимо рассчитать стоимость машиночаса работы электронно-вычислительной машины, на которой осуществлялась отладка. Данная величина соответствует величине арендной платы за один час работы электронно-вычислительной машины и определяется в соответствии с п. 3.3 настоящих методических указаний.

Затраты на отладку программы определяются по формуле (3.15):

Зот = Тотл × Sмч, (3.15)

где Тотл – трудоемкость отладки программы, час (определяется по нормам времени для программирования задач на электронно-вычислительных машинах или по формуле);

Sмч – стоимость машиночаса работы электронно-вычислительной машины, руб/час.

Зотп = 47,25×0,32 = 15,12 руб

Себестоимость разработки программного продукта определяется по формуле (3.16):

Спр = Зрз × F + Зот, (3.16)

где F – коэффициент накладных расходов проектной организации без учета эксплуатации электронно-вычислительной машины (при отсутствии данных может быть принят 1.15...1.2):

Спрп = 335,7×1,2+15,12 = 417,96 руб.

Отпускная цена складывается из себестоимости создания программного продукта и плановой прибыли на программу.

Отпускная цена программного продукта определяется по формуле (3.17):

Цо = Спр + Пр, (3.17)

где Пр – плановая прибыль на программу, руб.

Цоп = 417,96+125,39 = 543,35 руб.

Плановая прибыль на программу определяется по формуле (3.18):

Пр = Спр × Нп, (3.18)

где Спр – себестоимость программы;

Нп – норма прибыли проектной организации (при отсутствии данных может быть принята Нп = 0.25...0.3).

Прп = 417,96×0,3 = 125,39 руб.

Определение стоимости машиночаса работы электронно-вычислительной машины. Стоимость машиночаса работы электронно-вычислительной машины определяется по формуле (3.19):

Sмч = Сэ+ ((Аэвм + Рэвм + Апл +Рпл + Рар)/Фэвм), (3.19)

где Сэ – расходы на электроэнергию за час работы ЭВМ, руб;

Аэвм – годовая величина амортизационных отчислений на реновацию электронно-вычислительной машины;

Рэвм – годовые затраты на ремонт и техническое обслуживание электронно-вычислительной машины, руб.;

Апл – годовая величина амортизационных отчислений на реновацию производственных площадей, занимаемых электронно-вычислительной машины, руб;

Рпл – годовые затраты на ремонт и содержание производственных площадей, руб.;

Рар – годовая величина арендных платежей за помещение, занимаемое ЭВМ, руб.;

Фэвм – годовой фонд времени работы электронно-вычислительной машины, час.

Sмчп = 0,02+((30+39+14,4+20+403,92)/1680 = 0,32 час

Расходы на электроэнергию за час работы электронно-вычислительной машины определяются по формуле (3.20):

Сэ = Чэл × Цэ, (3.20)

где Цэ – стоимость 1 кВт-часа электроэнергии, руб.;

Чэл – среднечасовое потребление электроэнергии электронно-вычислительной машины, кВт.

Сэ = 0,1×0,19= 0,019

Годовая величина амортизационных отчислений на реновацию электронно-вычислительной машины определяется по формуле (3.21):

Аэвм = Цбэвм × Наэвм / 100, (3.21)

где Наэвм – норма амортизационных отчислений на электронно-вычислительную машину, % (Наэвм =10%);

Цбэвм – балансовая стоимость электронно-вычислительной машины, руб.

Аэвмп = 400×10/100 = 40 руб.

Годовые затраты на ремонт и техническое обслуживание электронно-вычислительной машины укрупненно могут быть определены по формуле (3.22):

Рэвм = Цбэвм× kро, (3.22)

где kро – коэффициент, учитывающий затраты на ремонт и техническое обслуживание электронно-вычислительной машины, в том числе затраты на запчасти, зарплату ремонтного персонала и др.(kро = 0.13).

Рэвмп = 400×0,13 = 52 руб.

Годовая величина амортизационных отчислений на реновацию производственных площадей, занятых электронно-вычислительной машиной определяется по формуле (3.23):

Апл = Sэвм × kд× Цпл × Напл / 100, (3.23)

где – норма амортизационных отчислений на производственные площади, % (=1,2%);

Sэвм – площадь, занимаемая электронно-вычислительной машиной, кв.м.;

kд – коэффициент, учитывающий дополнительную площадь (kд = 3);

Цпл – цена 1 квадратного метра производственной площади, руб.

Аплб = 1×3×500×1,2/100 = 18

Аплп = 1×3×500×1,2/100 = 18

Годовые затраты на ремонт и содержание производственных площадей укрупненно могут быть определены по формуле (3.24):

Рпл = Цбпл × kрэ, (3.24)

где kрэ – коэффициент, учитывающий затраты на ремонт и эксплуатацию производственных площадей (kрэ = 0,05).

Рплб = 500×0,05 = 25

Рплп = 400×0,05 = 20

Годовая величина арендных платежей за помещение, занимаемое электронно-вычислительной машиной, рассчитывается по формуле (3.25, 3.26):

Рар = Sэвм × kдп × kар × 12, (3.25)

Kар = Бст × km × kс × kц × kд, (3.26)

где Sэвм – площадь, занимаемая электронно-вычислительной машиной, кв.м;

kдп – коэффициент, учитывающий дополнительную площадь (kд = 3);

kар – ставка арендных платежей за помещение, руб;

Бст – базовая ставка за месяц, руб;

kc **–** коэффициент спроса от 0,5 до 3, руб;

kц – коэффициент цели использования имущества, руб;

km – повышающий коэффициент, учитывающий географическое размещение площади;

kд – дополнительный коэффициент.

Рар =2×3×5,61×12 = 403,92 руб.

Kар = 16,11×0,5×2,5×0,3×1,1= 6,64 руб.

Годовой фонд времени работы электронно-вычислительной машины определяется исходя из режима ее работы и может быть рассчитан по формуле (3.27):

Фэвм = tсс × Тсг, (3.27)

где tсс – среднесуточная фактическая загрузка электронно-вычислительной машины, час;

Тсг – среднее количество дней работы электронно-вычислительной мащины в год.

Фэвм = 7×240 = 1680 ч

**3.3 Определение цены потребления**

Обобщающим экономическим показателем является цена потребления, которая отражает полные затраты потребителя на приобретение, эксплуатацию продукции в течение периода полезного использования. Цена потребления определяется как совокупные единовременные (Цпр) и текущие затраты (Зт) на протяжении предполагаемого срока его полезного использования.

*Текущие затраты*включаютгодовые эксплуатационные расходы при обработке информации (Зр)и затраты, связанные с эксплуатацией электронно-вычислительной машины (За)

Годовые эксплуатационные расходы при обработке информации определяются по формуле (3.28):

Зр = Тр × к × tчр × (1 + q) × (1 + a) × (1 + b), (3.28)

где Тр – трудоемкость разового решения задачи, чел-ч.;

к – периодичность решения задачи в течение года, раз/год;

tчр – среднечасовая ставка работника, осуществляющего расчет задачи, руб;

q – коэффициент, учитывающий процент премий;

а – коэффициент, учитывающий дополнительную заработную плату;

b – коэффициент, учитывающий отчисления от фонда заработной платы.

Определение величин tчр, q, a, b ведется аналогично указаниям, приведенным в п.3.2.

Для расчета годовых текущих затрат, связанных с эксплуатацией программного продукта, необходимо определить время решения данной задачи на электронно-вычислительной машине.

Время решения задачи на электронно-вычислительной машине определяется по формуле (3.29):

Тз = (Твв + Тр + Твыв) × ((1+dпз)/60), (3.29)

где Твв – время ввода в электронно-вычислительную машину исходных данных, необходимых для решения задачи, мин;

Тр – время вычислений, мин;

Твыв – время вывода результатов решения задачи (включая время распечатки на принтере и графопостроителе), мин;

dпз – коэффициент, учитывающий подготовительно-заключительное время (dпз = 0.1...0.25).

Тзб = (0,8 + 0,1 + 2) × ((1 + 0,2)/60) = 0,058 ч

Тзп = (0,6 + 0,1 + 2) × ((1 + 0,2)/60) = 0,05 ч

Время ввода в электронно-вычислительную машину исходных данных может быть определено по формуле (3.30):

Твв = (Кz × Нz) / 100, (3.30)

где Кz – среднее количество знаков, набираемых с клавиатуры при вводе исходных данных;

Hz – норматив набора 100 знаков, мин (Hz = 6 ).

Тввб= (20 × 4)/100 = 0,8

Тввп= (20 × 3)/100 = 0,6

Зрб = 0,058 × 500 × 1,37 × (1+0,3) × (1+0,15) × (1+0,346) =79,95 руб

Зрп = 0,05 × 500 × 1,37 × (1+0,3) × (1+0,15) × (1+0,346) = 68,92 руб

В состав затрат, связанных с решением задачи, включаются также затраты, связанные с эксплуатацией электронно-вычислительной машины.

Затраты на оплату аренды электронно-вычислительной машины для решения задачи определяются по следующей формуле (3.32):

За = Тз × к × Sмч, (3.32)

где Sмч – стоимость одного машиночаса работы электронно-вычислительной машины, которая будет использоваться для решения задачи, руб (см. п. 1.4 настоящих указаний).

Заб = 0,058 × 500 × 0,34 = 9,86

Зап = 0,05 × 500 × 0,32 = 8

Годовые текущие затраты, связанные с эксплуатацией задачи, определяются по формуле (3.33):

Зт = Зп + За, (3.33)

где Зп – затраты на заработную плату пользователя программы;

За – затраты на оплату аренды электронно-вычислительной машины при решении задачи.

Зтб = 79,95+9,86 = 89,81 руб

Зтп = 68,92+8 = 76,92 руб

Расчет цены потребления определяется по формуле (3.1):

Цп = Цпр + Зт × t, (3.34)

где Цпр – цена приобретения без НДС,руб;

Зт – среднегодовые затраты по эксплуатации.руб;

t – срок службы лет программного продукта составляет 3 года.

Цпб = 520,10 + 89,81 × 3 = 789,53 руб

Цпп =543,35 + 76,92× 3 = 774,11 руб

Результаты расчета сводятся в таблицу 3.3 и представлены на плакате технико-экономические показатели проекта ДП-1090351602-2020-05-ПЛ.

Таблица 3.3 – Технико-экономические показатели проекта

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Варианты | |
| Базовый | Проектный |
| 1 Трудоемкость решения задачи, час | 0,058 | 0,05 |
| 2 Периодичность решения задачи, раз/год | 500 | 500 |
| 3 Годовые текущие затраты, связанные с решением задачи, руб. | 89,81 | 76,92 |
| 4 Отпускная цена программы, руб. | 520,10 | 543,35 |
| 5 Степень новизны программы | В | В |
| 6 Группа сложности алгоритма | 3 | 3 |
| 7 Цена потребления, руб | 789,53 | 774,11 |
| 8 Нормативный срок службы | 3 | 3 |
| 9. Групповой показатель по техническим параметрам Iтех | 1,96 | |
| 10Комплексный коэффициент конкурентоспособности продукта | 2,02 | |

Таким образом, приобретение нового программного продукта экономически целесообразно для пользователя, так как цена потребления в проектном варианте меньше на 15,42. Комплексный показатель конкурентоспособности продукта составит 2,02.

**4 ОХРАНА ТРУДА**

В данном разделе дипломного проекта рассматриваются вопросы охраны труда при работе с персональными электронно-вычислительными машинами (ПЭВМ).

ПЭВМ (персональная электронно-вычислительная машина) – настольная микро-ЭВМ, имеющая эксплуатационные характеристики бытового прибора и универсальные функциональные возможности.

На занятиях с использованием персональных электронно-вычислительных машин обучающиеся могут подвергаться воздействию различных опасных и вредных факторов, основными из которых являются:

* повышенная температура поверхностей ПК;
* повышенная или пониженная температура воздуха рабочей зоны;
* выделение в воздух рабочей зоны ряда химических веществ;
* повышенная или пониженная влажность воздуха;
* повышенный или пониженный уровень отрицательных и положительных аэроионов;
* повышенное значение напряжения в электрической цепи, замыкание;
* повышенный уровень статического электричества;
* повышенный уровень электромагнитных излучений;
* повышенная напряженность электрического поля;
* отсутствие или недостаток естественного света;
* недостаточная искусственная освещенность рабочей зоны;
* повышенная яркость света;
* повышенная контрастность;
* прямая и отраженная блесткость;
* зрительное напряжение;
* монотонность трудового процесса;
* нервно-эмоциональные перегрузки.
  1. **Производственная санитария**

Метеоусловия

В понятие метеорологических условий производственных помещений или микроклимата входят температура, влажность, скорость движения воздуха, тепловые излучения. Эти физические характеристики воздуха оказывают значительное влияние на протекание жизненных процессов в организме человека. Отрицательное влияние неблагоприятного микроклимата на организм проявляется в снижении производительности труда и изменении состояния здоровья.

Благоприятные (комфортные) метеоусловия на производстве являются важным фактором в обеспечении высокой производительности труда и профилактике заболеваний.

При несоблюдении гигиенических норм микроклимата не только снижается работоспособность человека, но и возрастает опасность возникновения травм и ряда заболеваний, в том числе профессиональных.

Рабочей зоной считается пространство высотой до 2-х метров над уровнем пола или площадки, на котором находится место постоянного или временного пребывания работающего.

Постоянным рабочим местом считается место, на котором работающий находится более 50% своего рабочего времени или более 2-х часов непрерывно. При выполнении работы в различных пунктах рабочей зоны постоянным рабочим местом считается вся рабочая зона.

Температура воздуха оказывает большое влияние на самочувствие человека и производительность труда.

Исследованиями установлено, что при температуре 180С человек может работать с полной отдачей.

Во время работы в «горячем» микроклимате производительность труда быстро падает и при температуре 300С снижается на 20-30%. Это касается как физической, так и умственной работы.

В организме человека непрерывно происходят окислительные процессы, связанные с образованием тепла, которое непрерывно отдается в окружающую среду.

Совокупность процессов, обуславливающих теплообмен между организмом и внешней средой, в результате которого температура тела поддерживается примерно на одинаковом уровне, называется терморегуляцией.

Теплоотдача организма во внешнюю среду зависит от температуры окружающей среды, от тяжести выполняемой работы, физического состояния и т.д.

Персональная электронно-вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении.

В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В таблице 4.1 указаны величины параметров микроклимата, создающие комфортные условия.

Труд программиста относится к категории физических работ Iа. Согласно СанПиН № 33 от 30.04.2013 «Требование к микроклимату рабочих мест в производственных и офисных помещениях».

Таблица 4.1 – Оптимальные значения параметров микроклимата на рабочих местах производственных и офисных помещений

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Период года | Категория работ по уровню энергозатрат, Вт | Температура воздуха, С | Температура поверхностей, С | Относительная влажность воздуха,  % | Скорость движения воздуха, м/с |
| Холодный | Iа | 22–24 | 21–25 | 60–40 | 0,1 |
| Теплый | Iа | 23–25 | 22–26 | 60–40 | 0,1 |

Таблица 4.2 – Оптимальные параметры температуры и относительной влажности воздуха

|  |  |  |
| --- | --- | --- |
| Температура, оС | Относительная влажность, % | Скорость движения воздуха, не более, м/с |
| 19 | 62 | 0,1 |
| 20 | 58 | 0,1 |
| 21 | 55 | 0,1 |

При повышении температуры воздуха в помещении кровеносные сосуды человека расширяются, происходит повышение притока крови к поверхности тела и теплоотдача в окружающую среду возрастает. При понижении температуры окружающей среды кровеносные сосуды сужаются и приток крови к поверхности тела, соответственно, замедляется, и теплоотдача уменьшается. Температуру в помещении следует регулировать с учетом тепловых потоков от оборудования. Предпочтение должно отдаваться оборудованию с малой электрической мощностью. Оборудование надо устанавливать так, чтобы тепловые потоки от него не были направлены на операторов. Следует также ограничивать количество вычислительной техники в помещении и избегать напольных отопительных систем.

Вредные химические вещества в воздухе рабочей зоны

Под вредным понимается вещество, которое при контакте с организмом человека вызывает производственные травмы, профессиональные заболевания или отклонения в состоянии здоровья.

Вредные вещества попадают в организм через органы дыхания, желудочно-кишечный тракт и кожный покров. Наиболее вероятно проникновение в организм веществ в виде газа, пара и пыли через органы дыхания (около 95% всех отравлений).

Непосредственными источниками выделения вредных веществ при плохом хранении могут быть подготовительные операции: размол и просеивание материалов, транспортирование сырья, травление, сушка.

Степень и характер вызываемых вредным веществом нарушений нормальной работы организма зависят от пути попадания в организм, дозы, времени воздействия, концентрации вещества, его растворимости. состояния воспринимающей ткани и организма в целом, атмосферного давления, температуры и других характеристик окружающей среды.

Следствием действия вредных веществ на организм могут быть анатомические повреждения, постоянные или временные расстройства и комбинированные последствия. Многие сильно действующие вредные вещества вызывают в организме расстройство нормальной физиологической деятельности без заметных анатомических повреждений, воздействий на работу нервной и сердечно-сосудистой систем, на общий обмен веществ и т.п.

Выделение вредных веществ в воздушную среду возможно при проведении технологических процессов и производстве работ, связанных с применением, хранением, транспортированием химических веществ и материалов, их добычей и изготовлением.

Отопление, вентиляция, кондиционирование

Отопление следует проектировать для обеспечения в помещениях расчетной температуры

воздуха, учитывая:

а) потери теплоты через ограждающие конструкции;

б) расход теплоты на нагревание инфильтрующегося наружного воздуха;

в) расход теплоты на нагревание материалов, оборудования и транспортных средств;

г) тепловой поток, регулярно поступающий от электрических приборов, освещения, технологического оборудования, коммуникаций, материалов, людей и других источников.

Потери теплоты через внутренние ограждающие конструкции помещений допускается не учитывать, если разность температур воздуха в этих помещениях равна 3 °С и менее.

Вентиляцию следует предусматривать для обеспечения допустимых параметров микроклимата и чистоты воздуха в обслуживаемой или рабочей зоне помещений (на постоянных и непостоянных рабочих местах).

Кондиционирование воздуха следует предусматривать для обеспечения нормируемых параметров микроклимата и чистоты воздуха в обслуживаемой или рабочей зоне помещения или отдельных его участков.

Кондиционирование воздуха следует принимать:

первого класса – для обеспечения параметров микроклимата, требуемых для технологического процесса, при экономическом обосновании или в соответствии с требованиями нормативных документов;

второго класса – для обеспечения параметров микроклимата в пределах оптимальных норм или требуемых для технологических процессов; допускается принимать скорость движения воздуха в обслуживаемой зоне, на постоянных и непостоянных рабочих местах, в пределах допустимых норм;

третьего класса – для обеспечения параметров микроклимата в пределах допустимых норм, если они не могут быть обеспечены вентиляцией в теплый период года без применения искусственного охлаждения воздуха, или оптимальных норм – при экономическом обосновании или на основании

задания на проектирование.

Учет тепловых потоков, расходов теплоносителя и конденсата для одного здания или группы зданий следует предусматривать в соответствии с требованиями СНиП 2.04.07.

В проектах следует предусматривать численность персонала по эксплуатации систем отопления, вентиляции и кондиционирования воздуха.

Согласно СНБ 4.02.01-03 «Отопление, вентиляция и кондиционирование воздуха» на предприятии в офисе отопление представлено в виде двух чугунных радиаторов. Используется естественная приточная вентиляция.

Шум

Шум – это совокупность звуков разной интенсивности и частоты, беспорядочно изменяющихся во времени, возникающих в производственных условиях и вызывающих у работников неприятные ощущения и объективные изменения органов и систем.

В зависимости от характера спектра шумы бывают постоянными и непостоянными.

Постоянный шум – это шум, при котором уровень звука за восьмичасовой день изменяется во времени не более чем на 5 дБ.

Непостоянный шум– это шум, при котором уровень звука за восьмичасовой рабочий день изменяется во времени не менее чем на 5 дБ.

В таблице 4.2 приведены предельно допустимые уровни звукового давления.

Согласно СанПиН №115 от 16.11.2011 «Шум на рабочих местах, в транспортных средствах, в помещениях жилых, общественных зданий и на территории жилой застройки» допустимый уровень шума представлен в таблице ниже.

Таблица 4.3 – Предельно допустимые уровни звукового давления

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вид трудовой деятельности, рабочее место | Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц | | | | | | | | | Уровни звука и эквивалентные по энергии уровни звука непостоянного шума, дБА |
| 31,5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |
| Научная деятельность, конструирование и проектирование, программирование, обучение и воспитание | 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |

Для снижения шума в помещении реализуется ряд мер: проводится акустическая обработка помещения: стены облицованы звукопоглощающим материалом для отделки помещений, разрешенным органами Государственного санитарного надзора Республики Беларусь. Организационные мероприятия включают своевременное и качественное проведение обслуживания и ремонта; выполнение правил технической эксплуатации.

Вибрация

Вибрация – это механические колебательные движения системы С упругими связями. Вибрация характеризуется спектром частот и такими кинематическими параметрами, как виброскорость и виброускорение или их логарифмическими уровнями в децибелах (дБ).

Вибрация относится к факторам, обладающим значительной биологической активностью. Характер, глубина и направленность функциональных сдвигов со стороны различных систем организма определятся прежде всего уровнем, спектральным составом и продолжительностью воздействия вибрации.

Нарушения здоровья работающего, обусловленные локальной или общей вибрацией, складываются из поражений нейрососудистой, нервно-мышечной систем, опорно-двигательного аппарата, изменений обмена веществ и др. При всех видах вибрационной болезни нередко наблюдаются изменения со стороны центральной нервной системы, которые связаны с комбинированным действием вибрации и интенсивного шума, постоянно сопутствующего вибрационным процессам.

По статистическим данным, 1/4 выявленных профессиональных заболеваний связано с воздействием вибрации и шума. Наиболее высокая заболеваемость вибрационной болезнью регистрируется в тяжелом, энергетическом, транспортном машиностроении, угольной промышленности и цветной металлургии.

Освещение

Необходимым условием для выполнения любого рода работ, является наличие необходимого уровня освещения, как естественного, так и искусственного. Недостаточное освещение может привести к перенапряжению глаз, к утомлению человека в целом. Это приводит к снижению внимания, ухудшению координации движений, развитию близорукости, расстройству нервной системы. Повышенная освещенность также оказывает негативное влияние на человека и вызывает слепящий эффект.

Работа за компьютером относится к работам высокой точности категории III. Рабочие столы размещены таким образом, чтобы экраны ЭВМ были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал слева.

Освещение в офисе представлены искусственным и естественным освещением.

В случае преимущественной работы с документами применяется комбинированное освещение, для освещения зоны расположения документов.

Выполнение расчета искусственного освещения можно проводить с помощью метода светового потока (метода коэффициента использования). Данный метод предназначен для определения средней освещенности. При расчете этим методом минимальная освещенность оценивается приблизительно, без выявления точек, в которых она имеет место. Данный метод целесообразен во всех случаях, когда расчет ведется по средней освещенности.

Произведем расчет освещенности в офисном помещении. Длина помещения – 8м, ширина – 6,5м, высота – 4м. В помещении проводятся работы высокой точности разряда IIIв. Тип светильника ЛПО2х40. Световой поток одной лампы равен 3120лм. Коэффициенты отражения от потолка – 70%; от стен – 50%, от пола – 30%.

Порядок расчета освещения по методу коэффициента использования светового потока. Определим расчетную высоту Нр, тип и количество светильников в офисном помещении по формуле (4.1).

h = H – hp – hc, (4.1)

где Н – высота помещения, м;

hc – расстояние светильника от перекрытия, м;

hр – высота рабочей поверхности над полом (обычно hр = 0,8), м.

Выполним расчеты:

h = 4 – 0 – 0,8 = 3,2м

Индекс офисного помещения i (он учитывает зависимость коэффициента использования светового потока от параметров помещения) определяется по формуле (4.2).

i = (A × B) / (h × (A + B), (4.2)

где А и В – ширина и длина помещения, м.

Определим индекс помещения:

i = (8 × 6,5) / (3,2 × (8 + 6,5) = 52 / 46,4 = 1,12

Определим коэффициент использования светового потока η в зависимости от типа светильника. В нашем случае коэффициент использования светового потока η = 0,55. Количество светильников определяется по формуле (4.3).

N= (E× k × S × Z)/(F ×n ×η) , (4.3)

где Е – минимальная нормированная освещенность, лк;

k – коэффициент запаса (для люминесцентных и ламп k=1,3);

S – освещаемая площадь, м2;

Z – коэффициент минимальной освещенности (при расчете освещения от светильников с люминесцентными лампами Z = 1,1);

F – световой поток одной лампы, лм;

n – число ламп в светильнике;

η – коэффициент использования светового потока в долях единицы.

В нашем случае минимальная нормированная освещённость равна 300лк. Рассчитаем количество ламп по формуле (4.3):

N = (300 × 1,3 × 52 × 1,1) / (3120 × 2 × 0,55) = 22,308/3432 = 7

В нашем случае необходимо 7 светильников.

Ультрафиолетовое излучение

Ультрафиолетовое излучение – это электромагнитное излучение в оптической области, примыкающее со стороны коротких волн к видимому свету и имеющее длины волн в диапазоне (200-400) нм.

Естественным источником ультрафиолетового излучения является солнце, искусственными – газоразрядные источники света, ртутно-кварцевые лампы, электрические дуги, лазеры и др.

Ультрафиолетовое излучение необходимо для нормальной жизнедеятельности человека.

При его длительном отсутствии в организме развиваются неблагоприятные явления, получившие название «светового голодания» или «ультрафиолетовой недостаточности».

Недостаток ультрафиолетового излучения вызывает авитаминоз, при котором нарушается фосфорно-кальциевый обмен и процесс костеобразования, а также происходит снижение защитных свойств организма от других заболеваний.

С другой стороны, длительное воздействие больших доз ультрафиолетового излучения может вызвать кожные заболевания, головную боль, тошноту, повышенную утомляемость, нервное возбуждение, повышение температуры тела, заболевания глаз и др. Длительное воздействие больших доз ультрафиолетового излучения может привести к развитию рака кожи.

Электромагнитное излучение

Электромагнитные поля низкой частоты, связанные с работой схем развертки электронного луча ЭЛТ-монитора, схем системного блока, а также электромагнитные поля промышленной частоты, образованные неэкранированными проводами (удлинителями) системы электрического питания ПЭВМ, являются одним из основных вредных воздействий компьютера на организм человека.

Воздействие электромагнитных излучений на организм человека приводит к нарушению нервной и сердечно-сосудистой систем, к из­менениям в составе крови.

Степень воздействия зависит от диапазона частот, интенсивности, продолжительности излучения.

Интенсивные сверхчастотные излучения (выше 300 МГц) вызывают патологию различных органов.

Рентгеновское излучение

Рентгеновское излучение возникает при торможении электронного луча на внутренней поверхности кинескопа.

Необходимо отметить, что по результатам замеров уровня реактивного излучения современных мониторов, изготовленных в соответствии с гигиеническими требованиями международных стандартов, уровень рентгеновского излучения на рабочем месте пользователя не превышает уровня естественного радиационного фона вследствие того, что значительная часть этого излучения поглощается специальным покрытием экрана видеотерминала.

* 1. **Техника безопасности**

Электробезопасность

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту людей от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Работодатель в зависимости от местных условий может предусматривать дополнительные меры безопасности труда, не противоречащие действующим правилам по охране труда при эксплуатации электро­установок. Эти меры безопасности должны быть внесены в соответствующие инструкции по охране груда, доведены до персонала в виде распоряжений, указаний, инструктажа.

Работники, принимаемые для выполнения работ в электроустановках, должны иметь профессиональную подготовку, соответствующую характеру работы. Электротехнический (электротехнологический) персонал обязан пройти проверку знаний норм и правил работы в электроустановках в пределах требований, предъявляемых к соответствующей должности или профессии, и иметь соответствующую группу по электробезопасности. Работнику, прошедшему проверку знаний по охране груда при эксплуатации электроустановок, выдается удостоверение установленного образца, в которое вносятся результаты проверки знаний.

Механические повреждения являются следствием резких непроизвольных судорожных сокращений мышц под действием тока, приходящего через тело человека, в результате могут произойти разрывы кожи, кровеносных сосудов и нервной ткани, вывихи суставов и даже переломы костей

К электротравмам общего характера относятся

1) электрический удар – возбуждение живых тканей организма проходящим через него электрическим током, сопровождающееся непроизвольными судорожными сокращениями мышц.

2) клиническая («мнимая») смерть – ‘переходный процесс от жизни к смерти, наступающий с момента прекращения деятельности сердца и легких. У человека отсутствуют все признаки жизни: он не дышит, сердце не работает, болевые раздражения не вызывают никаких реакций, зрачки глаз расширены и не реагируют на свет. Однако почти во всех тканях продолжаются обменные процессы на очень низком уровне, но достаточном для поддержания жизнедеятельности. Первыми начинают погибать очень чувствительные к кислородному голоданию клетки коры головного мозга, с деятельностью которых связаны сознание и мышление, поэтому длительность клинической смерти определяется временем с момента прекращения сердечной деятельности и дыхания до начала гибели клеток коры головного мозга, и составляет от 4-5 до 7-8 мин. После этого происходит множественный распад клеток коры головного мозга и других органов;

3) биологическая (истинная) смерть – необратимое явление, характеризующееся прекращением биологических процессов в клетках и тканях организма и распадом белковых структур; она наступает по истечении периода клинической смерти.

* 1. **Пожарная безопасность**

Категорирование помещения по взрывной и пожарной опасности

Категорирование помещений, зданий и наружных установок осуществляется в целях определения мер по обеспечению их взрывопожарной и пожарной безопасности.

Категории взрывопожарной и пожарной опасности помещений и зданий определяются для наиболее неблагоприятного в отношении пожара или взрыва периода исходя из вида находящихся в аппаратах и помещениях горючих веществ и материалов, их количества и пожароопасных свойств, особенностей технологических процессов.

В зависимости от количества и пожаро-взрывоопасных свойств обращающихся (находящихся) веществ и материалов с учетом особенностей технологических процессов намечаются мероприятия по обеспечению взрывопожарной и пожарной безопасности людей.

По взрывопожарной и пожарной опасности:

1) помещения подразделяются на категории А, Б, В1-В4, Г1, Г2, Д;

2) здания подразделяются на категории А, Б, В, Г и Д;

3) наружные установки подразделяются на категории Ан, Бн, Вн, Гн, Дн.

Степень огнестойкости здания

Помещение предприятия согласно ТКП 45-2.02-315-2018 [4] имеет III степень огнестойкости и Ф5.1 класс пожарной опасности.

При III степени огнестойкости зданий и объектов несгораемыми должны быть только несущие стены, каркас, колонны, а перегородки, междуэтажные и чердачные перекрытия могут быть из трудносгораемых материалов или из сгораемых, но оштукатуренных или обработанных огнезащитным составом.

К классу Ф5 относятся здания, сооружения и помещения производственного и складского назначения (для помещений этого класса характерно наличие постоянного контингента работающих, в том числе круглосуточно), а к Ф5.1 – производственные здания и сооружения, производственные и лабораторные помещения, мастерские.

Возможные причины пожара

Согласно ППБ 1.04.2014 «Правила противопожарной безопасности» опасными факторами, воздействующими на людей и на материальные ценности, являются: пламя и искры, повышенная температура окружающей среды, токсичные продукты горения и термического разложения, дым, пониженная концентрация кислорода, угарный газ, опасность взрыва.

К вторичным проявлениям опасных факторов пожара, воздействующим на людей и на материальные ценности, относятся: осколки, части разрушившихся аппаратов, агрегатов, установок, конструкций; радиоактивные и токсичные вещества и материалы, вышедшие из различных аппаратов и агрегатов; электрический ток, возникший в результате выноса высокого напряжения на токопроводящие части конструкций, аппаратов; огнетушащие вещества.

Возможными причинами пожаров в офисе являются: неисправность электрооборудования, неправильное устройство и эксплуатация оборудования, короткое замыкание, перегрузки.

Средства пожаротушения

Средства тушения пожара – это вещества и пожарная техника, способные прекратить процесс горения различных веществ и материалов.

Выбор огнетушащего средства определяется условиями возникновения и развития пожара и типом горючего вещества. При выборе средств тушения пожара необходимо учитывать эффективность тушения того или иного горючего материала (вещества), возможную порчу материальных ценностей. Кроме самих огнетушащих веществ для борьбы с огнем используется широкий спектр технических средств, разработанных как для защиты зданий, строений, сооружений, так и для доставки необходимого специального оборудования, инструмента, средств защиты.

Пути эвакуации людей

Большое значение уделено вопросу эвакуации при пожаре. Процесс эвакуации начинается одновременно из всех помещений и протекает в двух направлении в сторону выходов. Предусмотрено два эвакуационных выхода. В качестве организационно-технических мероприятий по обеспечению пожарной безопасности проводится обучение работников и обучающихся правилам пожарной безопасности, изготовление и применение средств наглядной агитации. Разработан порядок действий и эвакуации работников при пожаре.

Выходы относятся к эвакуационным, если они ведут из помещений:

* первого этажа – наружу непосредственно;
* любого надземного этажа – непосредственно на лестницу;
* подвального или цокольного этажа – наружу непосредственно;
* в соседнее помещение на том же этаже, обеспеченное выходами.

Количество и суммарная ширина эвакуационных выходов определяется в зависимости от максимально возможного числа эвакуирующихся через них людей и предельно допустимого расстояния от наиболее удаленного места их возможного пребывания до ближайшего эвакуационного выхода.

Пожарная сигнализация

Пожарная связь и сигнализация играют важную роль по предупреждению пожаров и способствуют своевременному обнаружению и вызову пожарных подразделений к месту возникшего пожара, обеспечивают управление и оперативное руководство работающими на пожаре. Пожарная сеть обеспечивается такими техническими средствами, как телефонная связь, радиосвязь, а также различными системами электрической пожарной сигнализации.

Все виды электрической пожарной сигнализации, независимо от ее системы, состоят из трех основных частей: извещателей, подающих сигнал о пожаре; приемной станции, предназначенной для приема поданных от извещателей сигналов о пожаре и автоматической подачи тревоги; системы проводов, соединяющих извещатели с приемной станцией. Системы электрической пожарной сигнализации устанавливаются как автоматического, так и ручного действия.

В зависимости от влияющего на них срабатывающего фактора автоматические извещатели подразделяются на группы: дымовые извещатели реагируют на появление дыма; тепловые – на повышение температуры воздуха в окружающей среде; световые – на появление и излучение открытого пламени; комбинированные – на тепло и дым.

# **Заключение**

В результате выполнения работы достигнута поставленная цель, которая состояла в изучении технологий обучения на основе использования образовательных сервисов Web 2.0 в ходе изучения темы учебного предмета «Информатика» на примере создания сайта «Баскетбол Беларуси в лицах».

При достижении цели решены следующие задачи:

* исследованы технологии обучения на основании использования образовательных сервисов web 2.0 в ходе изучения темы учебного предмета «Информатика»;
* разработано программное обеспечение – веб-сайт «Баскетбол Беларуси в лицах»;
* рассчитано технико-экономическое обоснование разработки веб-сайта;
* рассмотрены вопросы охраны труда.

Основными технологиями создания веб-сайта «Баскетбол Беларуси в лицах»: язык гипертекстовой разметки HTML, который служит для наглядного и хорошо структурированного представления информации; JavaScritp - скриптовый язык программирования общего назначения, интенсивно применяющийся для разработки веб-приложений; NodeJS – серверная платформа на языке JavaScritpt; SQL - универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных; phpMyAdmin - веб-приложение с открытым кодом, написанное на языке PHP, представляющее собой веб-интерфейс для администрирования СУБД MySQL.

**Список использованной литературы**

1. Брайт И. Секреты прибыльных интернет-магазинов / И. Брайт // Маркетинг в России и за рубежом. 2012. - № 6. - С. 97–101.
2. Григорьев Ю. А. Теория и практика проектирования систем на основе баз данных: учеб. пособие для вузов / Ю. А. Григорьев, А. Д. Плутенко. - Благовещенск: Изд-во Амур. гос. ун-та, 2008. - 394 с.
3. Дронов В. А. JavaScript в Web-дизайне/ В. А. Дронов. – СПб.: Питер, 2001. – 127 с.
4. Жуков Р. А. СУБД с открытым исходным кодом: возможность применения алгоритмов распараллеливания / Р. А. Жуков // Технические науки. – 2015. – № 1–2 (35–36). – С. 20 – 21.
5. Закон РБ «Об образовании» Главы 6-13 (Глава 6. Профессионально-техническое образование);
6. ЗАКОН РБ 1 июля 2007 г. N 252-З «О высшем образовании» (Статья 3. Государственная политика в сфере высшего образования).
7. ЗАКОН РБ от 5 июля 2006 г. № 141-З «Об общем среднем образовании» (Статья 3. Государственная политика в сфере общего среднего образования);
8. Зашеловский А. Е. Среды разработки программного обеспечения, ориентированные на процессы [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. – Режим доступа: http://www.scienceforum.ru/2015/pdf/15128.pdf (дата обращения 08.05.2016).
9. Институт социально-политических исследований при администрации Президента Республики Беларусь «Национальная система образования – важнейший стратегический ресурс государства (социальные гарантии и перспективы развития)», Минск 2005 г.;
10. Институт социально-политических исследований при администрации Президента Республики Беларусь «Национальная система образования в XXI веке», Минск 2006 г.;
11. Инструктивно-методическое письмо Министерства образования Республики Беларусь к 2009/2010 учебному году (Учреждения, обеспечивающие получение дошкольного образования);
12. Коцюба И. Ю. Основы проектирования информационных систем. Учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков. – СПб: Университет ИТМО, 2015. – 206 с.
13. Круг С. Как сделать сайт удобным. Юзабилити по методу Стива Круга / С. Круг – СПб.: Питер, 2010. – 208 с.
14. Курзыбова Я. В. Средства создания динамических web-сайтов: учеб. пособие / Я. В. Курзыбова. – Иркутск: Изд-во ИГУ, 2011. – 121 с.
15. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафлин. – СПб.: Питер, 2014. – 544 с.
16. Масляк Т. А. Анализ методов разработки сайтов / Т. А. Масляк, Т. А. Колесникова // Информационные технологии: наука, техника, технология, образование, здоровье: Тезисы докладов XXIII Международной научно-практической конференции, Ч.IV (20-22 мая 2015 г., Харьков). – С. 230.
17. Муравьев А. Технические аспекты веб-разработки [Электронный ресурс] / А. Муравьев, А. Григорьев. – Режим доступа: http://netology.ru/files/2930/Технические аспекты веб-разработки.pdf (дата обращения 06.05.2020).
18. Приступа В. В. Подход к выбору технологии создания динамического сайта [Электронный ресурс] / В. В. Приступа. - Режим доступа: http://www.repository.hneu.edu.ua/jspui/bitstream/  
    123456789/6097/29/sect3\_Prystupa.pdf (дата обращения 09.05.2020).

**Чертеж**

**Чертеж**

**Плакат**

**Плакат**

**Плакат**

**Плакат**

**Плакат**

**Приложение А**

(обязательное)

**План-конспект учебного занятия**

Тема: «Аппаратное и программное обеспечение компьютера»

Дата проведения: 22.03.2020

Цели занятия:

Обучающая:

* ознакомить с понятиями «архитектура компьютера», «аппаратное и программное обеспечение», классификацией программного обеспечения;
* дать общее представление об архитектуре компьютера;

Развивающая:

* способствовать развитию усидчивости и внимательности, умений правильно обобщать данные и делать выводы;

Воспитательная:

* содействовать воспитанию культуры общения, самостоятельности в учебно-трудовой деятельности, чувства исполнительности и ответственности за поведение и выполняемую работу.

Задачи:

* изучить аппаратное обеспечение компьютера, классификацию программного обеспечения.

Тип занятия: формирования новых знаний.

Материально-техническое оснащение: компьютер, системный блок как средство обучения.

Ход занятия

1. Организационный момент.

Приветствие.

Проверка присутствующих.

Объявление темы и задач урока.

Объяснение хода занятия.

1. Сообщение новых знаний

Компьютеры появились очень давно в нашем мире, но только в последнее время их начали так усиленно использовать во многих отраслях человеческой жизни. Ещё несколько лет назад было редкостью увидеть какой-нибудь персональный компьютер – они были, но были очень дорогие, и даже не каждая фирма могла иметь у себя в офисе компьютер. Теперь в каждом доме есть компьютер, который уже глубоко вошёл в жизнь самих обитателей дома.

Компьютер – это многофункциональное электронное устройство, предназначенное для накопления, обработки и передач» информации.

Архитектура компьютера определяется конструкцией и структурной организацией его функциональных блоков (компонентов), описанием принципов их работы и взаимодействия на аппаратном и программном уровнях. Основное внимание при рассмотрении архитектуры уделяется главным функциям, выполняемым компьютером: обработке, хранению и обмену информацией.

Работа любого компьютера осуществляется благодаря взаимосвязи двух компонентов: аппаратной части (компьютер и все его устройства) и программного обеспечения (программы, предназначенные для нормального функционирования аппаратной части).

Изучение аппаратной части компьютера сопровождается слайдшоу, разработанным в сервисе «SmileboxEditor» [15]. Памятка по использованию сервиса Web 2.0 «SmileboxEditor» представлена в приложении B.

Главной составляющей любого компьютера является системный блок. Системный блок – часть компьютера, содержащая практически все основные его устройства.

Корпус – внешняя оболочка системного блока персонального компьютера, которая защищает внутренние элементы от физического воздействия. Предназначен для компактного расположения и фиксации всех остальных комплектующих ПК;

Блок питания – питает электроэнергией все комплектующие системного блока (преобразовывает сетевое переменное напряжение в постоянное и подать его компонентам компьютера);

Материнская (системная) плата – устройство для сопряжения всех комплектующих ПК и обеспечение совместного функционирования. На ней расположены разъемы (слоты) для подключения разных устройств: процессора, модулей памяти, адаптеров и контроллеров, соединенных системной шиной;

Центральным устройством любого компьютера является процессор, который располагается на материнской плате, размещенной в системном блоке.

Процессор представляет собой устройство, предназначенное для обработки данных, которые находятся в его регистрах, в оперативной памяти, а также данные, размещенные во внешних портах процессора. От быстродействия процессора во многом зависит производительность компьютера.

При этом он интерпретирует часть данных, как непосредственно данные, часть – как адресные данные, а часть – как команды. Регистры процессора представляют собой память одинакового небольшого размера (32, 64 бита) и хранят некоторые выполняемые команды программы, адреса этих команд и промежуточные данные.

Работа центрального процессора может быть представлена в виде алгоритма из нескольких основных действий:

* найти и выбрать очередную команду программы из памяти компьютера и перенести ее в регистр команд;
* определить адрес последующей (будущей) команды;
* распознать тип команды, т.е. декодировать ее;
* если команде необходимы данные для выполнения, найти эти данные в памяти и перенести их в регистры;
* выполнить команду и вернуться к шагу 1 для выполнения следующей команды.

Характеристики процессора:

* разрядность (число одновременно обрабатываемых битов, машинных слов (8, 16, 32, 64 бита));
* тактовая частота (количество выполняемых операций в секунду);
* производительность (быстродействие компьютера, зависит от разрядности и тактовой частоты).

Сейчас на рынке присутствуют два основных брэнда: Intel и AMD.

Одним из важнейших компонентов компьютера является его память. В памяти компьютера хранятся программы и данные. Существуют различные способы классификации памяти Компьютерная память может быть разделена по скорости доступа к памяти, по времени хранения данных. Память делится на 2 части – внутреннюю и внешнюю.

Внутренняя память – это запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных, участвующих в вычислениях. Обращение к внутренней памяти ЭВМ осуществляется с высоким быстродействием, но она имеет ограниченный объем, определяемый адресацией машины. Внутренняя память делится на оперативную и постоянную.

Основной функцией внешней памяти ПК является способность долговременно хранить большой объем информации (на накопителях или дисководах).

Виды внутренней памяти:

* ОЗУ (Оперативная память):
* ПЗУ (BIOS).

Оперативная память – ОЗУ. Это набор микросхем, в которых хранятся данные, необходимые для работы ПК на текущий момент. Устанавливаются в специальные гнёзда на материнской плате. Она предназначена для временного и быстро доступного хранения данных в течение всего времени работы компьютера, для передачи на обработку в процессор. роме этого в оперативную память после включения компьютера записываются некоторые программы операционной системы. Эта часть оперативной памяти во время сеанса работы не доступна пользовательским программам.

Постоянная память (ПЗУ, ROM) отличается тем, что запись информации в ПЗУ осуществляется только 1 раз на заводе-изготовителе. И в дальнейшем из этой памяти возможно только чтение. Эта память энергонезависима, т.е. при выключении компьютера содержимое памяти не исчезает. Используется для хранения наиболее важных и часто используемых служебных программ, присутствие которых постоянно нужно компьютеру. Обычно это компоненты ОС (программа первоначальной загрузки), программы контроля оборудования.

Для ускорения работы компьютера разработчиками была создана быстрая по времени доступа кэш-память. Идея использования каш-памяти строится на том` что в ней хранятся часто используемые данные.

Каким бы ни был объем внутренней памяти компьютера, его всегда не хватает. В связи с этим большие объемы информации хранятся на внешней памяти. Внешняя память – предназначена для размещения больших объемов информации и обмена ею с оперативной памятью. Для внешней памяти используют энергонезависимые носители. Емкость внешней памяти практически не имеет ограничений, а для обращения к ней требуется больше времени, чем к внутренней.

В настоящее время в качестве внешней памяти в основном используются гибкие магнитные, жесткие магнитные, оптические диски.

Жёсткий диск – это устройство долговременной памяти, данные при выключении питания не стираются, скорость работы намного ниже, чем у оперативной памяти, а емкость намного выше. Содержит данные, используемые при работе с компьютером, которые при его отключении сохраняются. Существуют три вида жёстких дисков:

* HDD – обладают относительно малой скоростью чтения/записи, небольшой ценой и издают шум. При поломке данные подлежат частичному или полному восстановлению;
* SSD – обладают пониженной восприимчивостью к встряскам и ударам, относительно большой скоростью чтения/записи и высокой ценой, не издают шума;
* H-HDD – гибридный вариант двух видов.

Видеокарта – это устройство, преобразующее графический образ, хранящийся как содержимое памяти компьютера, в форму, пригодную для дальнейшего вывода на экран монитора.

Сетевая карта – это дополнительное устройство, позволяющее компьютеру взаимодействовать с другими устройствами сети. В настоящее время сетевые платы как правило интегрируются в материнские платы.

Звуковая карта – дополнительное оборудование персонального компьютера и ноутбука, позволяющее обрабатывать звук (выводить на акустические системы и/или записывать);

Дисковод (оптический привод) – устройство компьютера, позволяющее осуществить чтение и запись информации на съёмный носитель информации.

Система охлаждения (кулер и радиатор) – это система из вентиляторов, которая служит для отвода теплого воздуха с компонентов системного блока и подачи холодного воздуха из внешней среды (один монтируется на процессор, второй, на блок питания);

Современный компьютер имеет различные внешние типовые (периферийные) устройства, которые связаны основными устройствами с помощью разъемов (на задней стенке системного блока). Они могут использоваться для ввода. вывода информации или обеспечивать выполнение сразу двух этих функций.

Устройства ввода информации – приборы для занесения данных в компьютер.

Устройства ввода графической информации: сканер, графический планшет, видео- и Веб-камера, цифровой фотоаппарат.

Устройства ввода звука: микрофон, цифровой диктофон.

Устройства ввода текстовой информации: клавиатура.

Указательные (координатные) устройства с относительным указанием позиции (перемещения): мышь, трекбол, тачпад, джойстик.

Указательные (координатные) устройства с возможностью указания абсолютной позиции: графический планшет, световое перо, аналоговый джойстик.

Игровые устройства ввода: джойстик, педаль, геймпад, руль.

Устройства вывода информации – используются для извлечения результатов работы компьютера. Устройство вывода преобразуют информацию из двоичного кода в вид, понятный человеку.

Устройства для вывода визуальной информации: монитор (дисплей), проектор, принтер, графопостроитель.

Устройства для вывода звуковой информации: встроенный динамик, колонки, наушники.

Основу компьютера наряду с его аппаратным обеспечением составляет программное обеспечение. Программное обеспечение (ПО) – это совокупность всех программ и соответствующей документации, обеспечивающая использование ЭВМ в интересах каждого ее пользователя. Являясь неотъемлемой частью компьютерной системы, ПО определяет сферу применения конкретного компьютера. В современных компьютерах постоянно запускается и активно функционирует большое количество ПО. Одни занимаются арифметическими расчетами, другие строят диаграммы, рисуют или помогают оставаться на линии с собеседниками через почту. Программы выполняют функцию обработки информации. При этом она тесно взаимодействует с аппаратными устройствами компьютера.

Изучение классификации программного обеспечение проходит с помощью ментальной карты, которая представлена на рисунке А.1. Карта разработана в сервисе «Mindomo» [16].

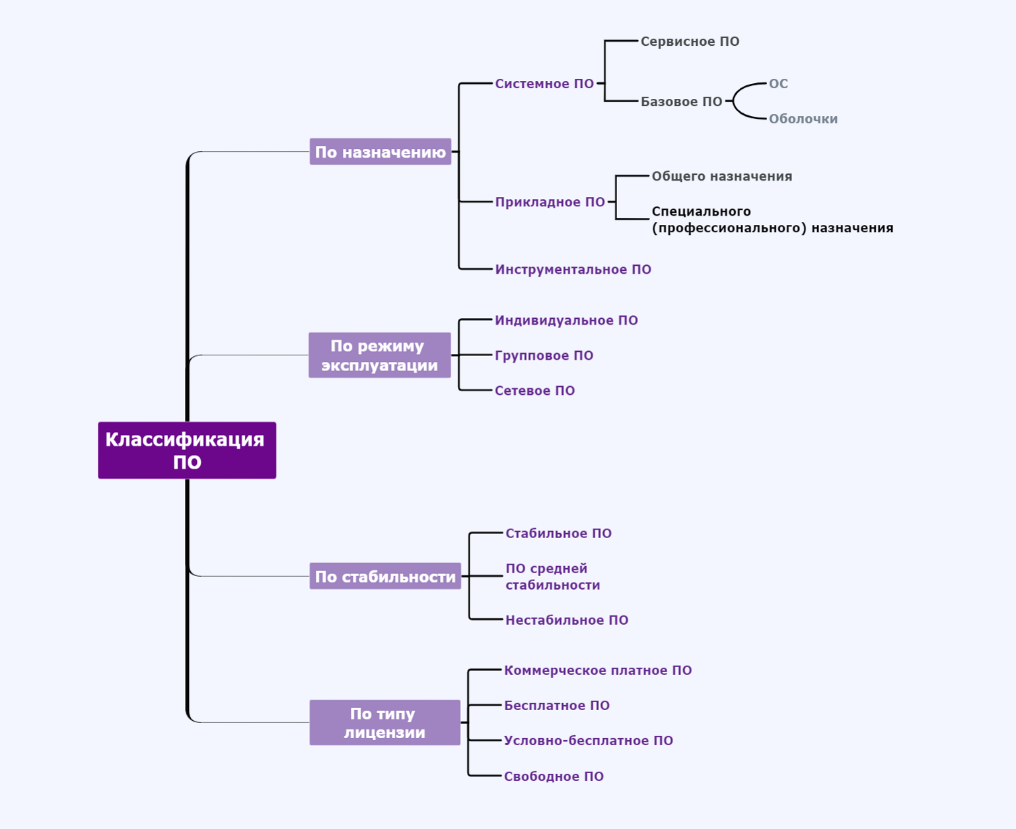


Рисунок А.1 – Ментальная карта, разработанная в сервисе «Mindomo»

Программное обеспечение, можно условно разделить:

1 По назначению:

* системное ПО;
* инструментальное ПО (системы программирования);
* прикладное ПО

Системное программное обеспечение (СПО) – это программы, управляющие работой компьютера и выполняющие различные вспомогательные функции, например, управление ресурсами компьютера, создание копий информации, проверка работоспособности устройств компьютера, выдача справочной информации о компьютере. Системное ПО подразделяют на базовое, сервисное. К базовому (общему) программному обеспечению относятся:

* [операционные системы](https://studopedia.ru/10_187335_operatsionnie-sistemi-servisnie-programmi.html) – обеспечивают функционирование и взаимосвязь всех компонентов компьютера, а также предоставляет пользователю доступ к его аппаратным возможностям),
* оболочки – интерпретатор команд ОС, обеспечивающий интерфейс для взаимодействия пользователя с функциями системы.

Сервисное ПО расширяют возможности ОС по обслуживанию системы, обеспечивают удобство работы пользователя.

Сервисное ПО включает в себя программы (утилиты): диагностики, антивирусные, обслуживания носителей, архивирования, обслуживания сети.

Прикладное ПО (приложения) – это комплекс программ для решения задач определённого класса конкретной предметной области. Прикладное ПО работает только при наличии системного ПО.

Прикладное ПО общего назначения – совокупность программ для решения общих универсальных задач. Эти программы используются большинством пользователей компьютера.

Прикладное ПО специального (профессионального) назначения – совокупность программ для решения узких и профессиональных задач различных предметных областей (архитектуры, строительства).

Инструментальное ПО – программное обеспечение, предназначенное для использования в ходе проектирования, разработки и сопровождения программ (компиляторы, отладчики, переводчики высокого уровня, редакторы).

2 По типу лицензии – право на использование.

* коммерческое платное – официально приобретенные или зарегистрированные у официальных распространителей программы (основные программные продукты фирм Microsoft, Adobe, 1C, Windows).
* Бесплатное. Исходный код закрыт, нельзя модифицировать программу. Часто разработчик выпускает домашнюю бесплатную, и корпоративную платную версии программы. Пример: антивирусы.
* условно-бесплатное. Программа работает определенное число запусков или небольшой промежуток времени. После этого она или перестает работать вообще, или отключаются некоторые функции/появится реклама. Пример: WinRar, CorelDraw.
* свободное. Программа не только бесплатная, но и идет с открытым исходным кодом. Это означает свободу модификации программ. Примеры: ОС Linux, Gimp, браузер Chromium, среда программирования PascalABC.NET.

3 По режиму эксплуатации:

* индивидуальное – использовать его будет только один пользователь, который владеет логином и паролем;
* групповое – доступ имеют несколько человек;
* сетевое – программы доступны всем, кому по сети раздали копии.

4 По стабильности:

* стабильное – внесение изменений маловероятно;
* средней стабильности – перемены вносятся дискретно;
* нестабильное – постоянные замены.

Операционная система (ОС) – комплекс программ системного уроню предназначенный для функционирования всех устройств компьютера и поддержи работы его программ. ОС выполняет дне основные функции: обеспечение пользователю-программисту удобств при работе с компьютером и повышение эффективности использования компьютера путем рационального управления его ресурсам. В настоящее время наиболее распространенными являются следующие ОС Windows, Маc OC, Linux.

Просмотр видеоролика «Какой путь прошли компьютеры и история ОС Windows продолжительностью 25 минут.

Закрепление изученного материала.

Фронтальный опрос:

Благодаря взаимосвязи каких двух компонентов осуществляется работа компьютера?

Назовите основные элементы системного блока.

Дайте понятие «программное обеспечение».

Назовите и охарактеризуйте виды ПО по назначению, типу лицензии, режиму эксплуатации, стабильности.

Также может быть предложено разгадать кроссворд (при наличии времени).

Заключительный этап.

Сегодня на занятии вы познакомились с понятием – архитектура компьютера. Узнали принципы построения первых ПК и устройство современных компьютеров, а также классификациями ПО. Что вы узнали нового? Что вызвало интерес?

Выставление отметок за урок. Постановка домашнего задания.

**Приложение Б**

(обязательное)

**Технологическая карта учебного занятия**

Специальность: Профессиональное обучение (Информатика).

Преподаватель: Близнюк Владислав Дмитриевич.

Тема учебного занятия: «Аппаратное и программное обеспечение компьютера.

Тип учебного занятия: комбинированный.

Цели учебного занятия:

Обучающая:

* ознакомить с понятиями «архитектура компьютера», «аппаратное и программное обеспечение», классификацией программного обеспечения;
* дать общее представление об архитектуре компьютера;

Развивающая:

* способствовать развитию усидчивости и внимательности, умений правильно обобщать данные и делать выводы;

Воспитательная:

* содействовать воспитанию культуры общения, самостоятельности в учебно-трудовой деятельности, чувства исполнительности и ответственности за поведение и выполняемую работу.организованность, аккуратность, четкость

Сведения об учебных материалах и материально-техническом оснащении представлены в таблицах Б.1 и Б.2.

Таблица Б.1 – Учебные материалы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид МТО | №  п/п | Шифр по паспорту кабинета | Наименование | Количество, шт |
| 1 | 2 | 3 | 4 | 5 |
| Иллюстративные СО | 1 | ЭС1 | Электронный слайд «Секреты Word» | 1 |
| 2 | ЭС2 | Электронный слайд «Вставка даты и времени» | 1 |
| 3 | ЭС3 | Электронный слайд «Быстрая смена регистра» | 1 |
| 4 | ЭС4 | Электронный слайд «Выделение фрагментов текста, расположенных в разных местах» | 1 |
| 5 | ЭС5 | Электронный слайд «Буфер обмена» | 1 |
| 6 | ЭС6 | Электронный слайд «Расстановка переносов» | 1 |
|  | 7 | ЭС7 |  | 1 |

Продолжение таблицы Б.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
|  | 8 | ЭС8 | Электронный слайд «Настройка ленты » | 1 |
| 9 | ЭС9 | Электронный слайд «Вертикальное выделение текста» | 1 |
| 10 | ЭС10 | Электронный слайд «Защита документа паролем» | 1 |

Таблица Б.2 – Материально-техническое оснащение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид МТО | № п/п | Шифр по паспорту кабинета | Наименование | Количество, шт |
| Технические средства обучения | 1 | ТСО1 | Ноутбук с поддержкой кабеля WGA либо HDMI | 1 |
| 2 | ТСО2 | Мультимедиа-проектор | 1 |
| 3 | ТСО3 | ПЭВМ для обучающихся | 10 и более |

Виды методов обучения по характеру познавательной деятельности

* объяснительно-иллюстративный (О-И);
* репродуктивный (Р);
* проблемное изложение (ПИ);
* частично-поисковый (эвристический) (Ч-П);
* исследовательский (И).

Ход учебного занятия представлен в таблице Б.3.

Таблица Б.3 – Ход учебного занятия

| Этапы (структурные элементы) учебного занятия | Время, мин | Промежуточные цели (по этапам) учебного занятия | Содержание деятельности | | Технология обучения | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Преподавателя | Обучающихся | Методы обучения | | СО (ТСО) |
| По характеру познава-тельной деятель-  ности | По источнику информации (степени взаимодействия педагог-учащийся) |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. **Организационная часть (3-5 мин)** | | | | | | | |
| 1.1 Проверка присутствующих учащихся | 2 | Проверить присутствующих учащихся | Слушает доклад дежурного. Отмечает отсутствующих учащихся в журнале | Дежурный докладывает о наличии учащихся к занятию | - | - | - |
| 1.2 Проверка готовности учащихся к занятию | 1 | Проверить готовность учащихся к занятию | Оценивает готовность учащихся | Дежурный докладывает о готовности учащихся к занятию | - | - | - |
| 1.3 Сообщение темы и цели занятия | 2 | Сообщить тему и цель занятия | Называет тему урока, формулирует цель занятия | Слушают, записывают в конспект тему | О-И | Словесный (рассказ)  Практический (конспектирование) | - |

| Продолжение таблицы Б.3 | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1. **Теоретический материал (40-42 мин)** | | | | | | | |
| 2.1 Создание, редактирование и обновление оглавления | 11 | Рассмотреть необходимость оглавления  Рассмотреть создание, редактирование и обновление оглавления | Рассказывает о необходимости оглавления в сложном документе  Демонстрирует на экране создание, редактирование и обновление оглавления | Записывают основные действия создания, редактирование и обновление оглавления | О-И | Словесный (рассказ)  Иллюстративный (демонстрация экрана)  Практический (конспектирование) | ТСО2 |
| 2.2 Создание закладки в Word | 7 | Рассмотреть понятие закладки, её необходимость.  Рассмотреть создание закладки | Рассказывает о понятии закладки, её необходимости в Word.  Демонстрирует на экране создание закладки в Word | Записывают понятие и основные действия создания закладки в Word | О-И | Словесный (рассказ)  Иллюстративный (демонстрация экрана)  Практический (конспектирование) | ТСО2 |
| 2.3 Создание сноски в Word | 7 | Рассмотреть понятие сноски, её необходимость.  Рассмотреть создание сноски | Рассказывает о понятии сноски, её необходимости в Word.  Демонстрирует на экране создание сноски в Word | Записывают понятие и основные действия создания сноски в Word | О-И | Словесный (рассказ)  Иллюстративный (демонстрация экрана)  Практический (конспектирование) | ТСО2 |

Продолжение таблицы Б.3

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2.4 Работа с таблицами | 11 | Рассмотреть исспользова-ние таблиц в Word.  Рассмотреть создание и редактирование таблицы | Рассказывает о необходимости таблиц в Word.  Демонстрирует на экране создание и редактирование таблицы | Записывают основные действия создания и редактирования таблицы | О-И | Словесный (рассказ)  Иллюстративный (демонстрация экрана)  Практический (конспектирование) | ТСО2 |
| **3. Закрепление материала (35-40 мин)** | | | | | | | |
| Выполнение задания | 35-40 | Выполнение заданий по пройденной теме | Раздаёт листики с заданиями.  Кратко рассказывает план действия. | Выполняют задания | И | Практический (упражнение) |  |
| **4. Подведение итогов (5-10 мин)** | | | | | | | |
| 4.1 Сообщение о достижении цели | 5 | Оценивание заданий | Просматривает и оценивает выполненные задания по пройденной теме.  Выставление оценок. | Демонстрируют выполненные задания. |  |  | - |
| 4.2 Сообщение домашнего задания | 3 | Рассказать домашнее задание | Объясняет домашнее задание | Задают вопросы | О-И | Словесный (беседа) | - |

**Приложение В**

(обязательное)

**Фрагмент исходного кода программы**

CREATE TABLE IF NOT EXISTS `newsartsek`.`admins` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(50) NOT NULL,

`hashed\_password` VARCHAR(60) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`subjects` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`menu\_name` VARCHAR(30) NOT NULL,

`position` INT(3) NOT NULL,

`visible` TINYINT(1) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`pages` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`subject\_id` INT(11) NOT NULL,

`menu\_name` VARCHAR(30) NOT NULL,

`position` INT(3) NOT NULL,

`visible` TINYINT(1) NOT NULL,

`content` TEXT NULL DEFAULT NULL,

`subjects\_id` INT(11) NOT NULL,

PRIMARY KEY (`id`),

INDEX `subject\_id` (`subject\_id` ASC),

INDEX `fk\_pages\_subjects1\_idx` (`subjects\_id` ASC),

CONSTRAINT `fk\_pages\_subjects1`

FOREIGN KEY (`subjects\_id`)

REFERENCES `newsartsek`.`subjects` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 11

DEFAULT CHARACTER SET = utf8

ROW\_FORMAT = DYNAMIC;

CREATE TABLE IF NOT EXISTS `newsartsek`.`images` (

`idimages` INT NOT NULL AUTO\_INCREMENT,

`imagescol` BLOB NULL,

`pages\_id` INT(11) NOT NULL,

PRIMARY KEY (`idimages`, `pages\_id`),

INDEX `fk\_images\_pages\_idx` (`pages\_id` ASC),

CONSTRAINT `fk\_images\_pages`

FOREIGN KEY (`pages\_id`)

REFERENCES `newsartsek`.`pages` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`comment` (

`idcomment` INT NOT NULL AUTO\_INCREMENT,

`pages\_id` INT(11) NOT NULL,

`comment` VARCHAR(250) NULL,

PRIMARY KEY (`idcomment`, `pages\_id`),

INDEX `fk\_comment\_pages1\_idx` (`pages\_id` ASC),

CONSTRAINT `fk\_comment\_pages1`

FOREIGN KEY (`pages\_id`)

REFERENCES `newsartsek`.`pages` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`role` (

`idrole` INT NOT NULL AUTO\_INCREMENT,

`namerole` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idrole`))

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`user` (

`iduser` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(50) NOT NULL,

`password` VARCHAR(50) NOT NULL,

`role\_idrole` INT NOT NULL,

PRIMARY KEY (`iduser`),

INDEX `fk\_user\_role1\_idx` (`role\_idrole` ASC),

CONSTRAINT `fk\_user\_role1`

FOREIGN KEY (`role\_idrole`)

REFERENCES `newsartsek`.`role` (`idrole`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `newsartsek`.`userphoto` (

`iduserphoto` INT NOT NULL AUTO\_INCREMENT,

`user\_iduser` INT NOT NULL,

`photo` BLOB NULL,

PRIMARY KEY (`iduserphoto`, `user\_iduser`),

INDEX `fk\_userphoto\_user1\_idx` (`user\_iduser` ASC),

CONSTRAINT `fk\_userphoto\_user1`

FOREIGN KEY (`user\_iduser`)

REFERENCES `newsartsek`.`user` (`iduser`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;