БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники

и автоматизированных систем»

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.В. Полозков

(подпись) (инициалы и фамилия)

« \_\_\_\_ » \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ДИПЛОМНОГО ПРОЕКТА

**«ИНТЕРНЕТ-ПРИЛОЖЕНИЕ ФОРМИРОВАНИЯ УЧЕБНЫХ ТЕСТОВ С ФУНКЦИЯМИ РАСЧЕТА ПОКАЗАТЕЛЕЙ КАЧЕСТВА»**

Специальность 1-40 01 01 «Программное обеспечение информационных технологий»

Специализация 1-40 01 01 05 «Управление качеством и тестирование программного обеспечения»

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обучающийся | |  | | |  | |  | |  |  |
| группы | 10701214 | |  |  | | |  | |  | М.С. Ермолаев |
|  | (номер) |  | | |  | | (подпись, дата) | |  |  |
| Руководитель | |  | |  | | |  | |  | Ю.В. Полозков |
|  | |  | |  | | | (подпись, дата) | |  |  |
| Консультанты: | |  | |  | | |  | |  |  |
|  | |  | |  | | |  | |  |  |
| по компьютерному проектированию | | | | | |  | |  |  | Ю.В. Полозков |
|  | |  | | |  | | (подпись, дата) | |  |  |
| по разделу «Охрана труда» | | | | |  | |  | |  | А. М. Лазаренков |
|  | |  | | |  | | (подпись, дата) | |  |  |
| по разделу «Экономика» | | | | |  | |  | |  | И. В. Носонова |
|  | |  | | |  | | (подпись, дата) | |  |  |
| Ответственный за нормоконтроль | | | | |  | |  | |  | И.Ю. Васильева |
|  | |  | | |  | | (подпись, дата) | |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Объем проекта: | |  | |  |  | |  | |
| расчетно-пояснительная записка –зазаписзаписка – | | | |  | | страниц; | | | |
| графическая часть – |  | | листов; | | | | |
| магнитные (цифровые) носители – | | | |  | | единиц. | | | |

Минск 2018

Оглавление

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ 7](#_Toc485800957)

[ВВЕДЕНИЕ 8](#_Toc485800958)

[ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ ПО ТЕМЕ 9](#_Toc485800959)

[1 ~~ОБЩИЙ ОБЗОР ВОПРОСА~~ 10](#_Toc485800960)

[1.1 Обзор существующих проблем компьютерного контроля знаний 13](#_Toc485800962)

[2 ОБЗОР ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ 10](#_Toc485800960)

[2.1 Методы проведения контроля знаний 13](#_Toc485800962)

[2.1.1 Частично адаптивные методы проведения контроля знаний 13](#_Toc485800962)

[2.1.2 Адаптивные методы проведения контроля знаний 13](#_Toc485800962)

[2.2 Модель адаптивного контроля знаний 13](#_Toc485800962)

[2.3 Выбор необходимых показателей качества 13](#_Toc485800962)

[3 ВЫБОР ПРОГРАММНО-ТЕХНИЧЕСКИХ СРЕДСТВ РЕАЛИЗАЦИИ ВЭБ-ПРИЛОЖЕНИЯ ФОРМИРОВАНИЯ УЧЕБНЫХ ТЕСТОВ 26](#_Toc485800970)

[3.1 Выбор языка программирования для серверной части 28](#_Toc485800962)

[3.2 Выбор системы управления реляционными базами данных 27](#_Toc485800962)

[3.3 Выбор стека технологий для клиентской части 28](#_Toc485800962)

[4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ДЛЯ ХРАНЕНИЯ ИНФОРМАЦИИ О ПРОХОЖДЕНИИ УЧЕБНЫХ ТЕСТОВ 29](#_Toc485800971)

[4.1 Концептуальная модель базы данных 30](#_Toc485800972)

[4.2 Логическая модель базы данных 31](#_Toc485800976)

[4.3 Физическая модель базы данных 32](#_Toc485800980)

[4.4 Создание хранимых процедур в базе данных 34](#_Toc485800980)

[4.5 Создание тригеров в базе данных 35](#_Toc485800980)

[5 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ УЧЕБНЫХ КУРСОВ 37](#_Toc485800981)

[5.1 Серверная часть 38](#_Toc485800981)

[5.2 Проектирование клинетской части 42](#_Toc485800981)

[5.3 Проектирование интерфейса 45](#_Toc485800981)

[6 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 50](#_Toc485800981)

[7 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 55](#_Toc485800981)

[8 ОХРАНА ТРУДА 60](#_Toc485800988)

[9 ОПРЕДЕЛЕНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 65](#_Toc485800988)

[ЗАКЛЮЧЕНИЕ 70](#_Toc485800991)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУ 71](#_Toc485800998)

[ГРАФИЧЕСКАЯ ЧАСТЬ 72](#_Toc485800998)

[ПРИЛОЖЕНИЕ А 75](#_Toc485800993)

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

В настоящей пояснительной записке применяются следующие сокращения:

* ЛУЗ – Логит уровня знаний
* ЛТЗ – Логит трудности задания
* КЗ – Контроль знаний
* МС – Модель студента
* УМ – Учебный материал
* БЗ – База знаний.
* БД – База данных.
* СУБД - Система управления базами данных
* ПО – Программное обеспечение.
* ПК – Персональный компьютер.

# ВВЕДЕНИЕ

Электронное обучение – обучение, реализованное на базе всестороннего применения мультимедиа, удаленного доступа к распределенным образовательным ресурсам на основе веб-технологий, с автоматизированным контролем и анализом результатов обучения и широким использованием разнообразных сетевых средств взаимодействия обучаемых между собой и с преподавателем. Развитие электронного обучения вызвано недостаточной эффективностью традиционных систем обучения, таких как лекционные и практические занятия, вследствие их малой информативности и сложностями в выработке индивидуального подхода к обучаемому. Система электронного обучения, напротив, обеспечивает возможность адаптации к обучаемому с учетом его уровня знаний. Цель электронного обучения заключается в обучении приемам самостоятельной работы, самоконтроля, взаимоконтроля, умений обобщать и делать выводы.

Одним из свойств электронного обучения является адаптивность, которая заключается в возможности приспособления к действиям пользователя. При этом система может изменять свои параметры и структуру в зависимости от работы пользователя. С использованием технологии адаптивного обучения, обучаемый получает возможность самостоятельно работать с учебным материалом, проходить контроль знаний и анализировать его результаты. Задания для контроля знаний подбираются с учетом успеваемости обучаемого. Таким образом, вырабатывается индивидуальный подход к обучению. Создание технологии адаптивного обучения было вызвано рядом недостатков традиционной организации учебного процесса:

– пассивностью многих обучаемых;

– ограниченностью аудиторного времени процесса обучения;

– сложностью преподавателя учитывать индивидуальные особенности обучаемого;

- слостью управления самостоятельной работы обучаемого.

К достоинствам системы адаптивного обучения можно отнести:

- возможность дистанционного обучения;

- контроль успеваемости обучаемых и качества составления заданий;

- возможность корректировки заданий и лекционных материалов с учетом современных требований;

- автоматизация обработки данных и высвобождение трудовых ресурсов на анализ данных;

- гибкость и расширяемость системы;

- быстродействие системы;

- возможность вносить изменения в информационную систему без изменения приложения клиента.

К ее недостаткам можно отнести:

- сложности создания и поддержания системы обучения;

- сложности в обеспечении безопасности данных;

- необходимость в обеспечении совместимости интерфейсов клиентской и серверной частей.

В связи с этим актуальной пролемой является разработка приложений формирования электронных курсов с применением адаптивного обучения, которые помогут преподавателям легко подбирать тестовые задания для студентов с учетом их уровня знаний, а также производить быструю оценку знаний студентов за счёт автоматизация обработки данных. В основу разрабатываемого приложения должен быть положен автоматизированный анализ результатов обучения и качества составления тестов, основанный на таких показателях, как ЛУЗ и ЛТЗ.

# ОБЗОР ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ ПО ТЕМЕ

В [1-3] авторами описаны основные модели и методы адаптивного контроля знаний.

[1] Л.В. Зайцева, Н.О.Прокофьева. Обучение в среде Интернет // Scientific Proceedings of Riga Technical University. Computer Science. Applied Computer Systems. – Vol. 3. – Riga : RTU, 2000, – р. 33 - 45.

[2] Л.В. Зайцева, Н.О.Прокофьева. Статья для Рижского технического университета “Модели и методы адаптивного контроля знаний” 2004 г.

[3] Л.В. Зайцева, Боул Ц. Student models in Computer-based Education // Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies. ICALT 2003. – Athens, Greece, 2003, – p. 451.

[4] Растригин Л.А. Обучение с моделью // Вопросы кибернетики. Человеко-машинные обучающие системы. - М.: АН СССР, 1979, – c. 40 - 49.

[5] Общие сведения об основных понятиях платформы .NET Framework [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/previous-versions/visualstudio/visual-studio-2008/zw4w595w(v=vs.90), свободный. – Загл. с экрана.

[6] .NET Framework 4.5 и 4.6 [Электронный ресурс] – Режим доступа: https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx, свободный. – Загл. с экрана.

[7] .NET Framework [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/.NET\_Framework, свободный. – Загл. с экрана.

[8] Рихтер Дж. Программирование на платформе Microsoft .NET Framework.– M.: Русская Редакция, 2003

[9] Дейтел Х., Дейтел П., Листфилд Дж., Нието Т., Йегер Ш, Златкина М. С# в подлиннике.– СПб.: БХВ-Петербург, 2006.

[10] Язык C# и платформа.NET [Электронный ресурс] – Режим доступа: http://metanit.com/sharp/tutorial/1.1.php, свободный. – Загл. с экрана.

[11] Обзор SQL Server [Электронный ресурс] – Режим доступа: <https://technet.microsoft.com/ru-ru/library/ms166352(v=sql.90).aspx> , свободный. – Загл. с экрана.

[12] Владимир Драч статья “Сравнение современных СУБД ” [Электронный ресурс] – Режим доступа: <http://drach.pro/blog/hi-tech/item/145-db-comparison> , свободный. – Загл. с экрана

[13] HTML5 [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/HTML5, свободный. – Загл. с экрана

[14] CSS3 [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/CSS, свободный. – Загл. с экрана

[15] JavaScript [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/JavaScript, свободный. – Загл. с экрана

[16] Bootstrap [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Bootstrap\_(фреймворк), свободный. – Загл. с экрана

# ОБЩИЙ ОБЗОР ВОПРОСА

## **Обзор существующих проблем компьютерного контроля знаний**

Начиная с истоков применения компьютеров в учебном процессе, особое внимание уделялось контролю знаний. Технические средства обучения, а затем и компьютеры в первую очередь использовались именно для проверки знаний учащихся. И до настоящего времени, несмотря на бурное развитие обучающих систем и других форм компьютерного обучения, контролирующие (тестирующие) программы составляют половину [1] имеющихся в сети Интернет программ учебного назначения (универсальные и специализированные обучающие системы, электронные энциклопедии, обучающие игры и т.д.) и являются наиболее разработанными. Проблемы компьютерного КЗ обычно рассматриваются в двух аспектах: методическом и техническом [1]. К методическим аспектам относятся: планирование и организация проведения контроля; определение типов вопросов и отбор заданий для проверки знаний студентов; формирование набора вопросов и заданий для опроса; определение критериев оценки выполнения каждого задания и контрольной работы в целом и др. К техническим аспектам относятся: автоматическое формирование набора контрольных заданий на основе выбранного подхода; выбор и использование в системе контроля параметров КЗ; выбор алгоритмов для оценки знаний учащихся и др. Поэтому вопросы компьютерного КЗ интересуют многих ученых, как педагогов, так и специалистов в области информационных технологий.

Необходимо отметить, что существует ряд интересных разработок, посвященных различным аспектам КЗ и основанных на современных достижениях науки и компьютерной техники. В то же время, формирование набора заданий для КЗ осуществляется, обычно, случайным образом; иногда учитывая параметры заданий; и лишь в отдельных случаях используется адаптивная выдача контрольных заданий на базе модели студента. [2]

# ОБЗОР ТЕХНОЛОГИИ АДАПТИВНОГО ОБУЧЕНИЯ

## Методы проведения контроля знаний

Методы организации контроля знаний можно разделить на три класса рис 2.1.1. [2]

* неадаптивные методы;
* частично адаптивные методы;
* полностью адаптивные методы.

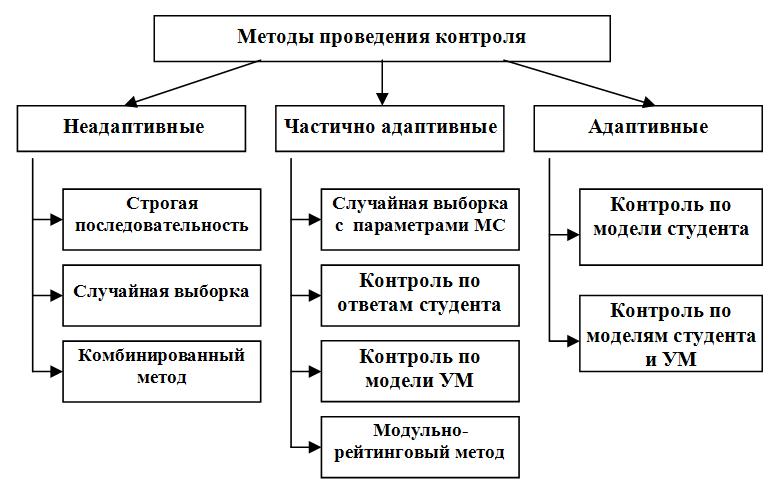


Рисунок 2.1.1. Методы организации контродя знаний

В контексте данного дипломного проекта логично будет рассматривать только частично адаптивные и адаптивные методы проведения контроля знаний.

## Частично адаптивные методы проведения контроля знаний

Частично адаптивные методы контроля предполагают использование информации из МС или УМ при формировании набора контрольных вопросов. К этому классу моделей относятся:

1) Случайная выборка с учетом отдельных параметров МС. Метод является развитием неадаптивных методов КЗ. Он аналогичен “Случайной выборке” и/или “Комбинированному методу”, т.е. набор заданий также формируется непосредственно перед контролем, но при генерации используются такие параметры МС, как общий уровень подготовленности, способность к обучению и, возможно, другие [3]. Таким образом, каждому студенту генерируется набор заданий, соответствующий его уровню подготовленности и способностям, что является главным преимуществом данного метода. Другое достоинство метода: студент, выполняя задания, соответствующие его способностям, не испытывает лишней психологической нагрузки во время контроля. В качестве недостатка данного метода можно отметить следующее: студенты получают задания различной трудности (это, безусловно, должно быть учтено при выставлении оценки), т.е. один выполняет только простые задания, а другой – трудные. Поэтому, генерируя вопросы студенту, соответствующие его способностям, целесообразно включить в набор и один - два задания повышенной трудности и значимости.

2) Контроль на основе ответов студента. В этом методе контроль осуществляется по заранее составленному сценарию или, другими словами, по разветвленной контролирующей программе. Пример такого сценария приведен на рис. 2.1.1.1, где вершины графа Вi соответствуют вопросам, предлагаемым студенту, а дуги указывают следующий выдаваемый вопрос в зависимости от правильности ответа: Пр – правильный ответ, Нт – неточный, Нп – неправильный ответ. Предварительная подготовка сценария КЗ дает возможность включить в программу вопросы разной степени трудности и значимости, расположив наиболее значимые и трудные задания в основной ветви программы (на рис. 2.1.1.1 это вопросы В1 и В6), а более простые – в разветвлениях. Таким образом, студенты получают разное число вопросов, а, следовательно, и время, затрачиваемое ими на контроль, различно, что является достоинством данного метода. Другое преимущество метода – простота обеспечения обратной связи (выдачи соответствующего комментария). Однако данный метод имеет существенный недостаток: всем студентам предлагаются одни и те же задания, однажды включенные в контролирующую программу. Устранить этот недостаток довольно просто – достаточно отделить сценарий КЗ от набора контрольных заданий. Для этого необходимо подготовить комплект однотипных вопросов для каждого Вi, включенного в сценарий контроля, т.е. Вi = {вi1, вi2, …, вiк}, а в процессе контроля случайным образом генерировать студенту вопрос из комплекта Вi.

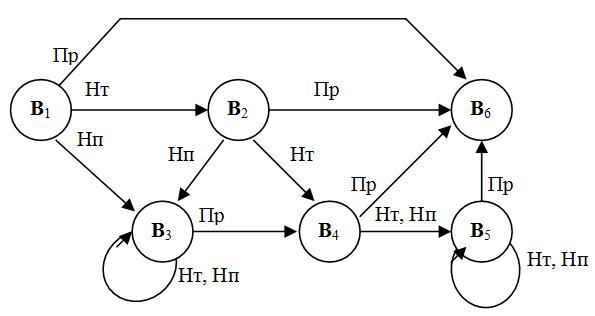


Рисунок 2.1.1.1. Методы организации контродя знаний

3) Контроль на основе модели учебного УМ. В данном методе формирование набора заданий для КЗ происходит на основе модели учебного материала (курса, темы, раздела темы), которая представляет собой ориентированный граф: множество вершин графа соответствует объектам изучения, а множество ребер – связям между ними. Изучение УМ, равно как и организация контроля, осуществляется в соответствии с оптимальной последовательностью изложения учебного материала, которая обычно есть ничто иное, как линейная последовательность объектов изучения. Таким образом, сначала генерируется задание для проверки знаний первого учебного объекта, затем – второго и т.д., т.е. последовательность выдачи заданий аналогична последовательности изучения учебного материала по модели УМ. При этом, если планируется проверить и знания, и умения, то одному учебному объекту могут соответствовать несколько вопросов.

4) Модульно-рейтинговый метод. Этот метод во многом аналогичен предыдущему. Учебный материал разделяется на отдельные составляющие – модули, для каждого из которых заранее подготавливается комплект контрольных заданий. В процессе КЗ студенту сначала предлагается вопрос из первого модуля. При этом после каждого ответа студента вычисляется его рейтинг. Переход к вопросам следующего модуля осуществляется при достижении определенного, заранее установленного рейтинга, причем студент с целью повышения своего рейтинга, а, следовательно, и оценки, может продолжить выполнение заданий текущего модуля и лишь затем перейти к следующему. [2]

## Адаптивные методы проведения контроля знаний

Адаптивные методы максимально используют информацию из моделей студента и/или учебного материала. К ним относятся:

1) Контроль по МС. В этом методе учитываются многие параметры МС, а именно:

* уровень подготовленности влияет на трудность предлагаемых заданий;
* вид репрезентативной системы обусловливает форму представления заданий (текст, визуальное изображение, использование звука);
* направленность личности влияет на формулировку текста выдаваемого задания;
* уровень беспокойства-тревоги определяет как наличие обратной связи, так и форму, и детальность комментариев;
* особенности памяти являются условием для определения времени выполнения задания и контрольной работы в целом;
* ответ студента, точнее, правильность ответа влияет на выбор следующего контрольного задания.

Сценарий контроля обычно формируется динамически в процессе КЗ, хотя набор сценариев для различных групп студентов может быть создан и заранее аналогично методу “Контроль по ответам студента”.

2) Контроль по МС и учебного материала. Данный метод является развитием предыдущего, т.е. при формировании контрольных заданий используются приведенные ранее параметры модели студента, но процесс КЗ строится на базе модели учебного материала, учитывая взаимосвязи между проверяемыми понятиями.[2]

## Модель адаптивного контроля знаний

Профессором Л.А. Растригиным [4] было предложено рассматривать процесс обучения как процесс управления сложной системой. Аналогично можно представить и процесс управления адаптивным контролем знаний рис. 2.2.1.

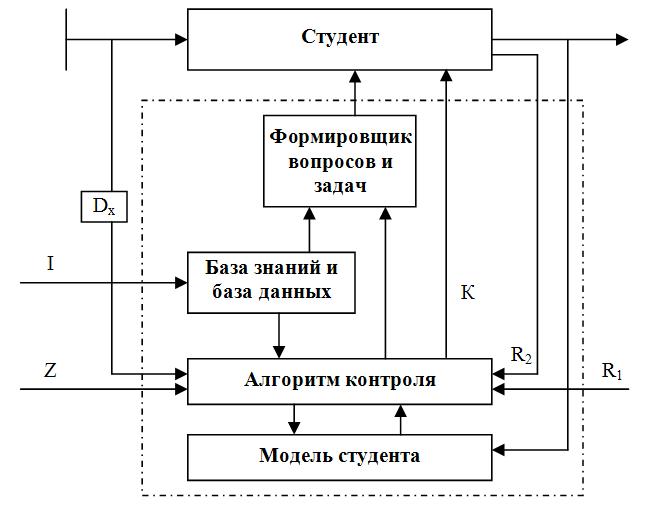


Рисунок 2.2.1. Модель адаптивного контроля знаний

Блок “Алгоритм контроля” выполняет следующие функции:

* анализ деятельности студента (проверка правильности его ответов и выполняемых действий);
* управление процессом контроля знаний на основе выбранного метода;
* определение результатов контроля, которое обычно сводится к выставлению оценки студенту.

БЗ содержит методы и/или модели процесса контроля, а также совокупность знаний предметной области. БД включает наборы вопросов и задач, предназначенных для проверки знаний студента и/или данные для формирования заданий. Контрольные задания могут также генерироваться автоматически на основе БЗ. БД и БЗ совместно с моделью студента образуют репозиторий системы контроля.

Модель студента включает разнообразную информацию о студенте: предыстория обучения; результаты текущей работы (тип выполненных заданий, время выполнения заданий, число обращений за помощью и т.д.); личностные психологические характеристики (тип и направленность личности, репрезентативная система, способность к обучению, уровень беспокойства-тревоги, особенности памяти и др.); общий уровень подготовленности и другие [3].

Формировщик вопросов и задач используется для формирования и выдачи студенту очередного задания (вопроса или задачи). Контроль знаний осуществляется следующим образом: студент выполняет предложенное задание, и результат его работы помещается в модель студента. Блок "Алгоритм контроля" на основе анализа ответа студента, целей контроля Z и используемого метода проведения контроля, учитывая внешние ресурсы R1 (например, возможности системы контроля) и внутренние ресурсы студента R2 (например, время контроля), а также состояние среды Dx, определяет параметры задания, которое должно быть предложено студенту. Формировщик вопросов и задач, получив от “Алгоритма контроля” данные о параметрах следующего задания, выбирает из БД и/или БЗ необходимую информацию I, формирует текст задания и выдает его студенту. В простейшем случае работа этого блока сводится к выбору нужного вопроса или задачи из базы данных. При некоторых видах контроля (например, при текущем КЗ или самопроверке) может быть предусмотрена обратная связь К, которая состоит в выдаче комментария на ответ студента. Таким образом, для управления адаптивным контролем знаний необходимо наличие:

* методов и моделей организации (проведения) контроля;
* моделей определения и оценки знаний, умений и навыков студента по результатам выполнения контрольных заданий.

[2]

## Выбор необходимых показателей качества

Эффективность обучения в адаптивной системе оценивается при помощи показателей качества данной системы. Показатель качества – это количественное выражение одного или нескольких характеристик или свойств объекта применительно к определенным условиям его создания и эксплуатации. При помощи показателей качества возможно оценить эффективность обучения и корректность составления тестовых заданий.

Чтобы определить эффективность обучения, необходимо представить выполнение задания в виде матрицы, где строки (i) содержат данные о решенных студентом задачах, столбцы (j) – результаты решения задач. Если представить, для примера, что четверо испытуемых отвечают на три задания, и что за каждый правильный ответ даётся один балл, а за неправильный- ноль, то результат тестирования можно представить в матрице Х4х3. Пример матрицы представлен в таблице 2.3.1.

Таблица 2.3.1 – Пример матрицы результатов тестового задания

|  |  |  |  |
| --- | --- | --- | --- |
| Испытуемые | Задания | | |
|  |  |  |
|  | 1 | 1 | 1 |
|  | 1 | 1 | 0 |
|  | 1 | 0 | 1 |
|  | 1 | 0 | 0 |

Для определения эффективности обучения необходимо определить показатели качества **отдельной** задачи и показатели успеваемости **отдельного** обучаемого. К основным показателям качества задачи относятся **трудность**, **доля правильных и неправильных ответов**, **логит трудности задания**. Уровень успеваемости обучаемых характеризуется **долей правильных и неправильных ответов**, **логитом уровня знаний**. На основании показателей качества задачи и показателей успеваемости имеется возможность рассчитать **вероятность решения задачи** обучаемым, что является критерием подбора заданий в адаптивной системе.

Одним из показателей качества задачи является ее **трудность Rj**. Авторы [Тархов С.В., Аванесов В.С.] предлагают расчитывать эмпирическую трудность заданий сложением элементов матрицы по строкам, что указывает на число правильных ответов, полученных по каждому j-му заданию. Таким образом, трудность задания определяется числом правильных на него ответов. Чем больше правильных ответов, тем оно легче для данной группы испытуемых. В силу простоты показатель Rj, удобен, но до тех пор, пока не появляются другие группы испытуемых с разным числом обучаемых N. Поэтому для получения сопоставительных характеристик Rj, делят на число испытуемых в каждой группе, результате чего рассчитывается **доля правильных ответов**:

|  |  |
| --- | --- |
| , | (2.3.1) |

где трудность задания, или общее количество правильных ответов на задание в группе испытуемых;

количество обучаемых.

Доля правильных ответов долго использовалась в качестве показателя трудности в так называемой классической теории тестов. Позже была осознана содержащаяся в ней смысловая неточность: ведь увеличение значения указывает не на возрастание трудности, а наоборот, на возрастание легкости, если можно применить такое слово. Поэтому в последние годы с показателем трудности заданий стали ассоциировать противоположную статистику – **долю неправильных ответов** :

|  |  |
| --- | --- |
| , | (2.3.2) |

где – количество неправильных ответов на задание в группе испытуемых,

количество обучаемых.

Также в технологиях адаптивного обучения и контроля используется другая мера трудности задания, называемая ***логитом трудности задания*** :

|  |  |
| --- | --- |
| , | (2.3.3) |

где – доля неправильных ответов, определяется по формуле (2.3.2);

– доля правильных ответов, определяется по формуле (2.3.1).

Этот показатель является отношением доли неправильных ответов к доле правильных ответов. Таким образом, чем выше значение данного показателя, тем труднее задание для группы испытуемых.

С точки зрения обучаемого существуют аналогичные характеристики – ***доля правильных и неправильных ответов испытуемого, логит уровня знаний***.

**Доля правильных ответов** испытуемого на тестовые задания определяется как отношение количества правильных ответов к общему количеству заданий:

|  |  |
| --- | --- |
| , | (2.3.4) |

где – количество правильных ответов испытуемого,

количество заданий.

Доля неправильных ответов определяется как доля неверных ответов к количеству заданий:

|  |  |
| --- | --- |
| , | (2.3.5) |

где – количество неправильных ответов испытуемого,

количество заданий.

**Логит уровня знаний** определяется как отношение доли правильных ответов к доле неправильных ответов испытуемого:

|  |  |
| --- | --- |
| , | (2.3.6) |

где – доля правильных ответов испытуемого, определяется по формуле (2.4);

– доля неправильных ответов испытуемого, определяется по формуле (2.3.5).

Нетрудно заметить, что логиты трудности задания (2.3.3) и уровня знаний (2.3.6) являются симметричными. Таким образом, имея значения данных показателей, возможно подобрать обучаемому задания согласно его уровню знаний. На основании логитов трудности задания и успеваемости студентов имеется возможность рассчитать вероятность решения задания обучающимся при помощи метрической системы Г. Раша [2]:

|  |  |
| --- | --- |
| , | (2.3.7) |

где – логит уровня знаний испытуемого, определяется по формуле (2.3.6),

– логит трудности задания, определяется по формуле (2.3.3).

Логиты трудности задания и уровня знаний являются одними из ключевых характеристик, поскольку позволяют подобрать задания обучаемому исходя их его уровня знаний.

Таким образом, при использовании адаптивного обучения достигается автоматизация контроля успеваемости обучаемых, качества составления тестовых заданий и методических материалов. При данной системе достигается гибкость обучения, поскольку обучаемому задания выдаются согласно его уровню знаний, а задания, не прошедшие проверки, полежат корректированию или удалению.

# ВЫБОР ПРОГРАММНО-ТЕХНИЧЕСКИХ СРЕДСТВ РЕАЛИЗАЦИИ ВЭБ-ПРИЛОЖЕНИЕ ФОРМИРОВАНИЯ УЧЕБНЫХ ТЕСТОВ

## Выбор серверного языка программирования

Языков программирования, используемых для серверной вебразработки, достаточно много: PHP, Ruby, Java, C#, Python, Perl и другие. В техническом плане для данного проекта нет какких-либо ограничений при выборе языка, то есть практически любой функционал сайта может быть успешно реализован на любом из них, значит главную роль при выборе языка для реализации серверной части будет играть опыт работы студента с языком, а также наличие удобных инструментов для веб-разработки и интеграции с другими продуктами.

Поэтому в качестве среды разработки была выбрана Microsoft Visual Studio 2015, платформа .NET, язык программирования С#.

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Она состоит из общеязыковой среды выполнения (среды CLR) и библиотеки классов .NET Framework.

Основой платформы .NET Framework является среда CLR, которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду. Среду выполнения можно считать ат гентом, который управляет кодом во время выполнения и предоставляет основные службы, такие как управление памятью, управление потоками и удаленное взаимодействие. При этом накладываются условия строгой типизации и другие виды проверки точности кода, обеспечивающие безопасность и надежность. Основной задачей CLR является управление кодом. [5]

Библиотека классов платформы .NET Framework представляет собой коллекцию типов, которые тесно интегрируются со средой CLR. Библиотека классов является объектно-ориентированной; предоставляя типы, из которых управляемый код пользователя может наследовать функции. Кроме того, компоненты независимых производителей можно легко объединять с классами платформы .NET Framework. Типы .NET Framework позволяют решать типовые задачи программирования, включая работу со строками, сбор данных, подключения к базам данных и доступ к файлам. В дополнение к этим обычным задачам библиотека классов содержит типы, поддерживающие многие специализированные сценарии разработки. [6]

Архитектура .NET Framework описана и опубликована в спецификации Common Language Infrastructure (CLI), разработанной Microsoft и утверждённой ISO и ECMA. В CLI описаны типы данных .NET, формат метаданных о структуре программы, система исполнения байт-кода и многое другое. [7,8]

Язык C# был создан в 2000 году корпорацией Microsoft, основным постулатом этого языка является высказывание: «всякая сущность есть объект». Текущей версией языка является версия C# 7.0, которая вышла 7 марта 2017 года вместе с Visual Studio 2017. Язык основан на строгой компонентной архитектуре и реализует передовые механизмы обеспечения безопасности кода. Он создан Microsoft специально для платформы .NET.

В языке программирования С# представлены важнейшие аспекты разработок в области вычислительных систем: объектно-ориентированное программирование, обработка графических объектов, компоненты графического пользовательского интерфейса (GUI), обработка исключений, многопоточная обработка, мультимедиа, обработка файлов, подготовленные структуры данных, работа с базами данных, разработка многозвенных программных приложений для Интернета, организация сетей, Web–службы и распределённые вычисления [9].

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших программ до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей [10]. Основной его задачей является практически реализовать компонентно-ориентированный подход к программированию, который способствует меньшей машинно-архитектурной зависимости результирующего программного кода, большей гибкости, переносимости и легкости повторного использования кода.

## Выбор системы управления реляционными базами данных

СУБД — это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями. Обычно СУБД различают по используемой модели данных. Так, СУБД, базирующиеся на использовании реляционной модели данных, называют реляционными СУБД. Системы управления базами данных помогают отсортировать информацию, а также связать базы данных между собой, при этом предоставив отчет об изменениях и зарегистрированных событиях.

Несмотря на то, что все системы управления базами данных выполняют одну и ту же основную задачу (т.е дают возможность пользователям создавать, редактировать и получать доступ к информации, хранящейся в базах данных), сам процесс выполнения этой задачи варьируется в широких пределах. Кроме того, функции и возможности каждой СУБД могут существенно отличаться. Различные СУБД документированы по-разному: более или менее тщательно. По-разному предоставляется и техническая поддержка.

На данный момент на рынке представлено большое множество СУБД, таких как: MySQL, MS SQL Server, PostgreSQL, MongoDB, MariaDB, DB2 и другие.

При выборе СУБД для реализации данного проекта нам потребуются только базовые функции и возможности, а также небольшое количество информации, размещаемой в БД. Также важным критерием при выборе СУБД является хорошая совместимость и взаимодействовие с .NET платформой, так как в качестве серверного языка программирования было решено использовать C#.

Идеальным решением является MS SQL Server, полностью удовлетворяющий наши потребности. SQL Server представляет «среды» для облегчения выполнения задач разработки и управления: среда SQL Server Management Studio и среда Business Intelli-gence Development Studio. В среде Management Studio можно разрабатывать решения компонента SQL Server Database Engine и решения по уведомлению и управлять ими, управлять развернутыми решениями служб Analysis Services, запускать пакеты служб Integration Services и управлять ими, управлять серверами, отчетами и моделями отчетов служб Reporting Services. В среде BI Development Studio можно разрабатывать решения бизнес-аналитики: проекты служб Analysis Services используются для разработки кубов, измерений и структур интеллектуального анализа данных; проекты служб Reporting Services — для создания отчетов; проекты моделей отчетов — для определения моделей отчетов и проекты служб Integration Services — для создания пакетов. Обе эти среды тесно взаимосвязаны с Microsoft Visual Studio и Microsoft Of-fice System [11]. MS SQL Server обладает следущими преимуществами и недостатками:

**Достоинства**

* Продукт очень прост в использовании
* Текущая версия работает быстро и стабильно
* Движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов.
* Он очень хорошо взаимодействует с другими продуктами Microsoft.
* Существует бесплатная экспресс версия(MS SQL Express.).

**Недостатки**

* Даже при тщательной настройке производительности корпорация SQL Server способен занять все доступные ресурсы.
* Существуют проблемы с использованием службы интеграции для импорта файлов.

[12]

## Выбор стека технологий для клиентской части

Выбор технологий для Front-end разработки является одним из самых простых, так как на рынке существуют явные фавориты – это HTML5, CSS3 и JavaScript.

HTML5 — язык для структурирования и представления содержимого всемирной паутины. Это пятая версия HTML. Цель разработки HTML5 — улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров. В HTML5 реализовано множество новых синтаксических особенностей. Например, элементы <video>, <audio> и <canvas>, а также возможность использования SVG и математических формул. Эти новшества разработаны для упрощения создания и управления графическими и мультимедийными объектами в сети без необходимости использования сторонних API и плагинов. [13]

CSS3 — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL. CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля. [14]

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности вебстраницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке. [15]

Также для удобства разработки было принято решение использовать библиотеку JQuery, которая помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими.

Для оформления веб-страниц было принято решение использовать Bootstrap3 библиотеку, так как она предоставляет возможность приятного оформления сайта и уже заложена в новый проект при создании его через Visual Studio.

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. [16]

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Главное окно модуля отрисовки графических объектов представлено на рисунке 4.1.

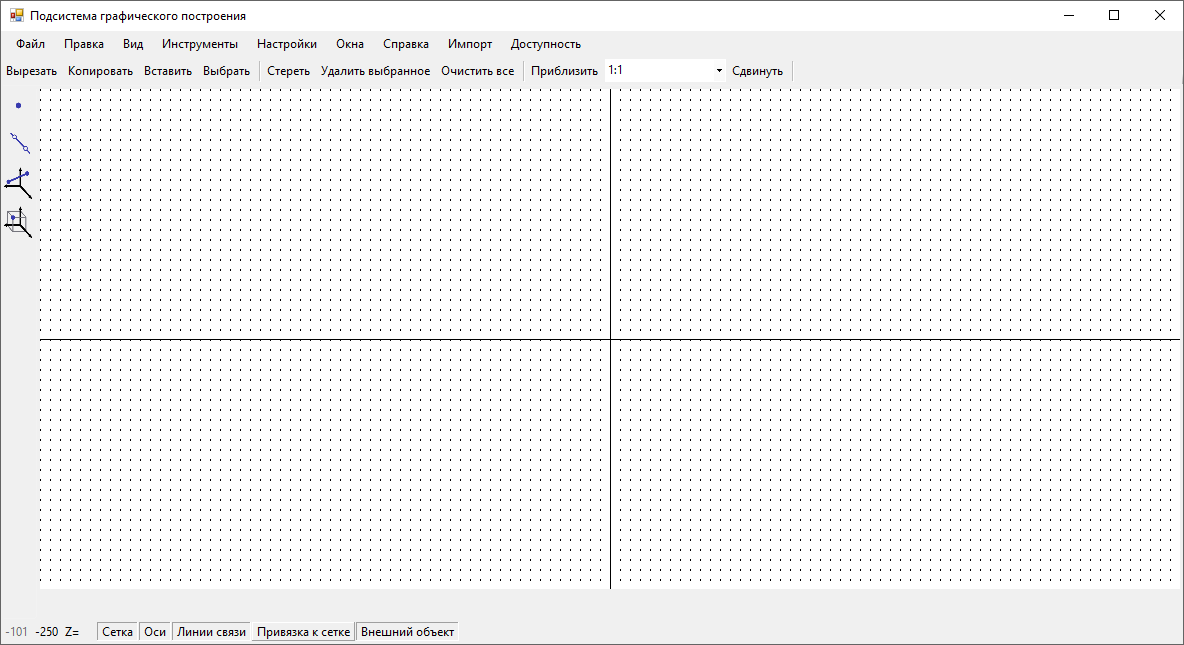


Рисунок 4.1 – Главное окно подсистемы

Доступ к основному функционалу предоставляется иерархической объектной моделью: Чертеж – Графическая область – Плоскость проекций. К наиболее востребованным при работе командам доступ реализован с помощью панелей инструментов: слева – рисования, вверху – стандартная, редактирования, снизу - управления рабочим окном, при построении примитива – панель параметров примитива. Все панели являются плавающими, предоставляется возможность закрепить их в любом месте экрана.

Стандартная панель предоставляет доступ к общему функционалу приложения – возможность создать, открыть, сохранить графический материал задачи. Настроить вид и положения панелей инструментов, осуществить контроль доступа пользователю к основным построениям, настроить внешний вид редактора и примитивов позволяет панель настроек, представленная на рисунке 4.2.

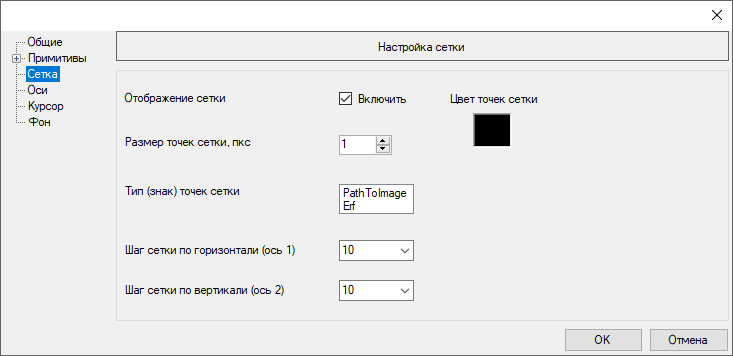


Рисунок 4.2 – Панель настроек

Панель рисования осуществляет доступ к графическим примитивам, реализованным в библиотеке графических примитивов. На данный момент для посторения доступны 2D точка, 3D точка и её проекции, 2D прямая, 3D прямая и её проекции, 2D отрезок, 3D отрезок и его проекции, 2D плоскость, 3D плоскость и её проекции. Возможно так же генерировать вышеописанные 3D примитивы из составных проекций соответствующей командой. Доступность вариантов построений зависит от требований задачи, регулируется администратором во вкладке доступа в настройках редактора. На примере на рисунке 4.3 построены 3D прямая, две 2D точки, проекция отрезка на плоскость XOY.

Панель редактирования позволяет изменять параметры чертежа и примитивов. Реализованы функции очистки полотна, выделения и удаления примитивов, их копирования и вставки.

Панель управления рабочим окном позволяет включать и отключать видимость осей, сетки, линий связи, привязку курсора к сетке, а так же отображает текущие локальные координаты курсора с учетом плоскости проекции.

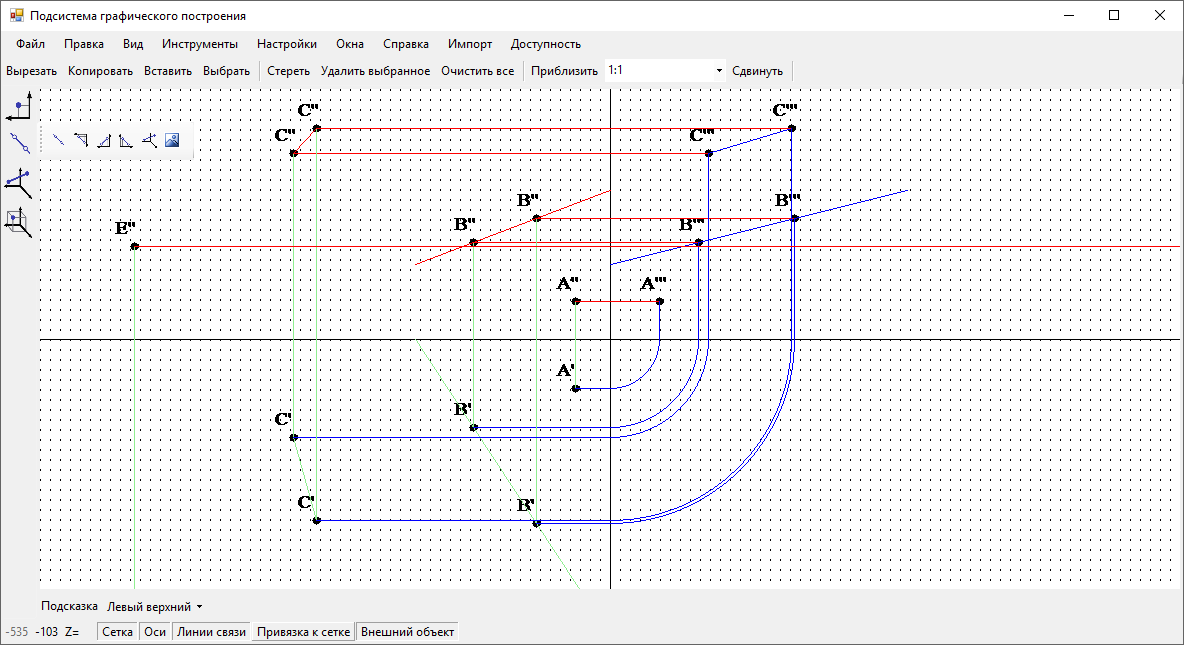


Рисунок 4.3 – Пример построения примитивов

# ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

## Определение единовременных затрат на создание программного продукта (разработку подсистемы)

Определение трудоемкости разработки ПП:

, (5.1)

где – трудоемкость подготовки описания задачи и исследования алгоритма решения;

– трудоемкость разработки блок-схемы алгоритма;

– трудоемкость программирования по готовой блок-схеме;

– трудоемкость отладки программы на ЭВМ;

– трудоемкость подготовки документации по задаче в рукописи;

– трудоемкость редактирования, печати и оформления документации по задаче.

Составляющие приведенной формулы определяются, в свою очередь, через условное число операторов Q в разрабатываемом ПП по формуле:

, (5.2)

где q - число операторов в программе;

С - коэффициент сложности программы, характеризует относительную сложность программ задачи по отношению к так называемой типовой задаче, сложность которой принята за единицу (принят равным 3);

p - коэффициент коррекции программы, 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 ( принят равным 0,2).

Таким образом,

4320.

Составляющие трудоемкости разработки программы определятся по формулам:

, (5.3)

, (5.4)

(5.5)

, (5.6)

, (5.7)

, (5.8)

где W - коэффициент увеличения затрат труда вследствие недостаточного или некачественного описания задачи (принят равным 1,3);

К - коэффициент квалификации разработчика алгоритмов и программ (так как стаж работы программистом до двух лет – принят 0,8).

Рассчитаем составляющие трудоемкости:

Тогда

.

### 5.1.2 **Определение себестоимости создания ПП**

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

, (5.9)

где – трудоемкость разработки программного продукта (1403,41 человек/час);

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

– коэффициент, учитывающий процент премий в организации-разработчике (принят 0,3);

– коэффициент, учитывающий дополнительную заработную плату (принят 0,15);

– коэффициент, учитывающий отчисления от фонда заработной платы (принят 0,346).

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

(5.10)

где – среднемесячная заработная плата работника 1 разряда (равна 31 рублю);

– тарифный коэффициент работника (принят равным 3,15);

170 – среднее нормативное количество рабочих часов в месяце для 2015 года.

Таким образом

=1637,95 (руб.).

В себестоимость разработки ПП включаются также затраты на отладку ПП в процессе его создания.

Затраты на отладку программы определяются по формуле:

, (5.11)

где – трудоемкость отладки программы (691,2 час.),

– стоимость машиночаса работы ЭВМ ( руб/час).

Получаем

.

Себестоимость разработки ПП определяется по формуле:

, (5.12)

где – коэффициент накладных расходов проектной организации без учета эксплуатации ЭВМ (принят равным 1,15).

Тогда себестоимость равна

.

### 5.1.3 **Определение оптовой и отпускной цены ПП**

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

Оптовая цена ПП определяется по формуле:

, (5.13)

где – плановая прибыль на программу, руб.

Плановая прибыль на программу определяется по формуле:

, (5.14)

где – себестоимость программы ( руб.);

– норма прибыли проектной организации (принимаем равной 0,25).

Тогда

,

.

Отпускная цена программы определяется по формуле:

, (5.15)

где – оптовая цена программы, руб;

– затраты на заработную плату разработчиков программы;

– размер плановой прибыли на программу;

НДС – ставка налога на добавленную стоимость (20%).

Рассчитаем отпускную цену программы:

.

### 5.1.4 **Определение стоимости машиночаса работы ЭВМ**

Стоимость машиночаса работы ЭВМ определяется по формуле:

, (5.16)

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

– годовая величина амортизационных отчислений на реновацию ЭВМ;

– годовые затраты на ремонт и техническое обслуживание ЭВМ, руб;

– годовая величина амортизационных отчислений на реновацию производственных площадей, занимаемых ЭВМ, руб;

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

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

– годовой фонд времени работы ЭВМ, час.

Расходы на электроэнергию за час работы ЭВМ определяются по формуле:

, (5.17)

где – среднечасовое потребление элетроэнегрии ЭВМ (0.09 кВт);

– стоимость 1 кВт-часа электроэнергии (0,19 руб.).

Тогда

.

Годовая величина амортизационных отчислений на реновацию ЭВМ определяется по формуле:

, (5.18)

Где – цена ЭВМ на момент ее выпуска (820 руб);

– коэффициент удорожания ЭВМ (используется цена текущего года, принимаем равным 1);

– коэффициент, учитывающий затраты на монтаж и транспортировку ЭВМ (kм = 1,05);

– норма амортизационных отчислений на ЭВМ, % (10%);

– балансовая стоимость ЭВМ, руб.

Получим

,

.

Годовые затраты на ремонт и техническое обслуживание ЭВМ укрупненно могут быть определены по формуле:

, (5.19)

где – коэффициент, учитывающий затраты на ремонт и техническое обслуживание ЭВМ, в том числе затраты на запчасти, зарплату ремонтного персонала и др. (= 0,13).

Тогда

.

Годовая величина амортизационных отчислений на реновацию производственных площадей, занятых ЭВМ определяется по формуле:

, (5.20)

где – балансовая стоимость площадей, руб;

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

– площадь, занимаемая ЭВМ (2 кв.м.);

– коэффициент, учитывающий дополнительную площадь (= 3);

– цена 1 квадратного метра производственной площади (300 руб.).

Тогда

,

.

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

, (5.21)

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

Тогда

.

Годовая величина арендных платежей за помещение, занимаемое ЭВМ, рассчитывается по формуле:

, (5.22)

где – площадь, занимаемая ЭВМ (2 кв.м);

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

– ставка арендных платежей за помещение (12 руб.);

– коэффициент комфортности помещения (принимаем равным 0,75);

– повышающий коэффициент, учитывающий географическое размещение площади (принимаем равным 0,9).

Тогда

.

Годовой фонд времени работы ЭВМ определяется исходя из режима ее работы и может быть рассчитан по формуле:

, (5.23)

где – среднесуточная фактическая загрузка ЭВМ (8 часов);

– среднее количество дней работы ЭВМ в год (равно количеству рабочх дней в году, исключая праздники – 220 дней).

Тогда

.

Рассчитаем стоимость машиночаса работы ЭВМ

.

## 5.2 **Определение ожидаемого прироста прибыли в результате внедрения ПП**

Внедрение ПП может обеспечить пользователю ожидаемый прирост прибыли за счет сокращения трудоемкости решения задачи, являющейся предметом автоматизации и, как результат, снижения текущих затрат, связанных с решением данной задачи.

Так как внедряемый ПП заменяет ручной труд, то производится сопоставление текущих затрат, связанных с решением задачи в ручном режиме и автоматизированном.

### 5.2.1 Определение **годовых эксплуатационных расходов при ручном решении задачи**

Годовые эксплуатационные расходы при ручной обработке информации (ручном решении задачи) определяются по формуле:

, (5.24)

где – трудоемкость разового решения задачи вручную чел.-час;

к – периодичность решения задачи в течение года;

– среднечасовая ставка работника, осуществляющего ручное выполнение построений ( руб);

– коэффициент, учитывающий процент премий (принят 0,35);

– коэффициент, учитывающий дополнительную заработную плату (принят 0,15);

– коэффициент, учитывающий отчисления от фонда заработной платы (принят 0,346).

Для разового выполнения цикла обучения по одной задаче вручную требуется 2 человека (студент, выполняющий задачу, и преподаватель, проверяющий выполнение задачи), которым потребуется 8 часов. Цикл обучения включает в себя объяснение необходимого для решение задачи материала преподавателем, решение задачи студентом, проверка правильности решения, выявление ошибок и их анализ. Таким образом, трудоемкость решения задачи в год будет равна 4 чел.-час.

Учитывая то, что учебный год составляет примерно 45 недель, при этом задача решается 2 раза в день, а выполнение одного цикла решения задачи обучения по задаче составляет 1 день, периодичность решения задачи в течение года к – 440 раз.

Исходя из этого рассчитаем годовые эксплуатационные расходы при ручной обработке информации:

.

### 5.2.2 Определение годовых текущих затрат, связанных с эксплуатацией задачи

Для расчета годовых текущих затрат, связанных с эксплуатацией ПП, необходимо определить время решения данной задачи на ЭВМ.

В время решения задачи на ЭВМ входит время ввода информации (выполнение требований задачи по построению), время обработки результата (время, которое требуется программе для анализа правильности построения) и время на вывод результата проверки.

Время решения задачи на ЭВМ определяется по формуле:

, (5.25)

где – время ввода в ЭВМ исходных данных и выполнения необходимых для решения задачи построений, мин;

– время вычислений (30 сек);

– время вывода результатов проверки решения задачи (30 сек);

– коэффициент, учитывающий подготовительно-заключительное время (=0,25).

Время ввода в ЭВМ исходных данных может быть определено по формуле:

, (5.26)

где – среднее количество знаков, набираемых с клавиатуры при вводе исходных данных ( =20);

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

Таким образом

,

*.*

На основе рассчитанного времени решения задачи может быть определена заработная плата пользователя данного ПП. Затраты на заработную плату пользователя ПП определяются по формуле:

, (5.27)

где – время решения задачи на ЭВМ, час;

– среднечасовая ставка пользователя программы (0,57 руб),

– принимает равным 440 (эквивалентно ручному труду);

– коэффициент, учитывающий процент премий (принят 0,35);

– коэффициент, учитывающий дополнительную заработную плату (принят 0,15);

– коэффициент, учитывающий отчисления от фонда заработной платы (принят 0,346).

Тогда

.

В состав затрат, связанных с решением задачи включаются также затраты, связанные с эксплуатацией ЭВМ.

Затраты на оплату аренды ЭВМ для решения задачи определяются по следующей формуле:

, (5.28)

где – стоимость одного машиночаса работы ЭВМ, которая будет использоваться для решения задачи(0,53 руб).

Тогда

.

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

, (5.29)

где – затраты на заработную плату пользователя программы;

– затраты на оплату аренды ЭВМ при решении задачи.

Тогда

.

### 5.2.3 **Определение ожидаемого прироста прибыли в результате внедрения ПП**

Ожидаемый прирост прибыли в результате внедрения задачи взамен ручного ее расчета укрупненно может быть определен по формуле:

, (5.30)

где Снп - ставка налога на прибыль (18%).

Рассчитаем ожидаемый прирост прибыли:

.

## 5.3 **Расчет показателей эффективности использования программного продукта**

Для определения годового экономического эффекта от разработанной программы необходимо определить суммарные капитальные затраты на разработку и внедрения программы по формуле:

, (5.31)

где Кз - капитальные и приравненные к ним затраты;

Цпр - отпускная цена программы.

Капитальные и приравненные к ним затраты определяются для случая, если ЭВМ, на которой предполагается решать рассматриваемую задачу, отслужила к моменту расчета х лет, по формуле:

, (5.32)

Таким образом

*,*

.

Годовой экономический эффект от сокращения ручного труда при обработке информации определяется по формуле:

, (5.33)

где Е – коэффициент эффективности, равный ставке за кредиты на рынке долгосрочных кредитов (Е = 0,4).

Тогда

.

Срок возврата инвестиций определяется по формуле:

, (5.34)

Тогда

*.*

Технико-экономические показатели проекта приведены в таблице 5.1.

Таблица 5.1 – Технико-экономические показатели проекта

|  |  |  |
| --- | --- | --- |
| Наименование показателя | Варианты | |
| базовый | проектный |
| 1. Трудоемкость решения задачи, час | 4 | 0,05 |
| 2. Периодичность решения задачи, раз/год | 440 | 440 |
| 3. Годовые текущие затраты, связанные с решением задачи, руб |  |  |
| 4. Отпускная цена программы, руб |  |  |
| 5. Степень новизны программы |  | А |
| 6. Группа сложности алгоритма |  | 1 |
| 7. Прирост условной прибыли, руб |  |  |
| 8. Годовой экономический эффект, руб. |  |  |
| 9. Срок возврата инвестиций, лет |  |  |

# ОХРАНА ТРУДА

## Производственная санитария

Особенность современного производства, в частности увеличение скоростей и мощностей производственных процессов, автоматизация, применение ЭВМ, ЧПУ резко изменили характер трудовых процессов и роль в них человека.

Условия труда операторов ЭВМ характеризуются возможностью воздействия на них комплекса опасных и вредных производственных факторов:

* электрического тока и статического электричества;
* ионизирующих и неионизирующих излучений;
* шума;
* вибрации;
* нарушение параметров микроклимата
* тепловыделений и вредных веществ;
* специфических условий зрительной работы, параметров технологического оборудования и рабочего места.

При работе персонала за компьютером основными вредными производственными факторами, наряду с вышеперечисленными, являются: неправильная организация освещения, нарушения параметров микроклимата, присутствие вибрации, несоблюдение эргономических требований при проектировании рабочих мест операторов, психоэмоциональное напряжение.

К средствам защиты от вышеперечисленных вредных факторов относятся: вентиляция, искусственное освещение, звукоизоляция. Существуют нормативы, определяющие комфортные условия и предельно допустимые нормы запылённости, температуры воздуха, шума, освещённости. На постоянных рабочих местах обеспечивается соответствующие микроклиматические параметры, уровни освещенности, шума и состояние воздушной среды. [3]

### 6.1.1 Шум

По происхождению шум делят на механический, обусловленный колебаниями деталей машин и их взаимным перемещением, аэродинамический (гидродинамический), возникающий в упругих конструкциях в газе или жидкости, и шумы электрических машин. Для рабочих мест ВЦ характерно наличие всех видов шумов.

Основными источниками шума в помещениях с ВТ принтеры, плоттеры и оборудование для кондиционирования воздуха, а в самих ЭВМ – вентиляторы систем, трансформаторы и источники бесперебойного питания.

Шум, неблагоприятно воздействуя на организм человека, вызывает психические и физиологические нарушения, снижающие работоспособность и создающие предпосылки для общих и профессиональных заболеваний и производственного травматизма.

Уровни физических факторов (уровни звукового давления, уровни звука), создаваемые ВДТ, ЭВМ, ПЭВМ и периферийными устройствами, не должны превышать ПДУ, предусмотренные таблицей 6.1 и устанавливаются в зависимости от следующих категорий производимых работ:

категория I – выполнение основной работы на ВДТ в диспетчерских, операторских, расчетных кабинах и постах управления, залах вычислительной техники и другое, а также в помещениях с ВДТ, ЭВМ и ПЭВМ всех типов учреждений образования;

категория II – выполнение работы на ВДТ, ЭВМ и ПЭВМ в помещениях, где работают инженерно-технические работники, осуществляющие лабораторный, аналитический или измерительный контроль;

категория III – выполнение работы в помещениях операторов ЭВМ (без дисплеев);

категория IV – выполнение работы на ВДТ, ЭВМ и ПЭВМ в помещениях для размещения шумных агрегатов (алфавитно-цифровые печатающие устройства, принтеры и другое). [3]

Таблица 6.1 – Предельно-допустимые уровни звука, эквивалентные уровни звука и уровни звукового давления в октавных полосах частот при работе с ВДТ, ЭВМ и ПЭВМ и периферийными устройствами

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Категория нормы  Шума | Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц | | | | | | | | | Уровни звука и  эквивалентные уровни звука, дБА |
| 31.5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |  |
| I | 86 | 71 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |
| II | 93 | 79 | 70 | 63 | 58 | 55 | 52 | 50 | 49 | 60 |
| III | 96 | 83 | 74 | 68 | 63 | 60 | 57 | 55 | 54 | 65 |
| IV | 103 | 91 | 83 | 77 | 73 | 70 | 68 | 66 | 64 | 75 |

Нормированные уровни шума обеспечиваются использованием малошумящего оборудования, применением звукопоглощающих материалов для облицовки помещений, а также различных звукопоглощающих устройств (перегородки, кожухи, прокладки и т.д.). Наиболее шумное оборудование располагается в специально отведенных помещениях (гермозонах).

### 6.1.2 Вибрация

Уровень вибрации на рабочих местах операторов не превышает 75дБ (по виброскорости) согласно. [3]

Нормированные уровни вибрации обеспечиваются путем использования виброизоляторов из материалов с большим внутренним трением (резины, пробки, войлока, асбеста или стальных пружин).

При работе с вычислительной техникой (ВТ) уровень вибрации не превышает допустимых пределов, так как в ВТ нет вращающихся узлов и механизмов (за исключением вентилятора) (СанПиН № 132 от 26.12.2013г. [4]).

### 6.1.3 Микроклиматические условия

Одним из необходимых условий здорового и высокопроизводительного труда является обеспечение нормальных метеорологических условий и чистоты воздуха в рабочей зоне помещений. Выполняемая оператором ЭВМ работа по энергозатратам относится к категории "легкая - 1а" с наиболее низкими энергозатратами. Микроклиматические условия в помещениях с вычислительной техникой принимаем по Санитарные нормы и правила «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами» и Гигиенический норматив «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами», утвержденные постановлением МЗ РБ от 28.06.2013 г. № 59 [3] (по таблице 6.2).

Таблица 6.2 – Оптимальные параметры микроклимата для помещений с ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Категория работ | Температура воздуха, оС, не более | Относительная влажность воздуха, % | Скорость движения воздуха, м/с |
| Холодный | легкая-1а | 22-24 | 40-60 | 0,1 |
| легкая-1б | 21-23 | 40-60 | 0,1 |
| Теплый | легкая-la | 23-25 | 40-60 | 0,1 |
| легкая-1б | 22-24 | 40-60 | 0,2 |

Воздух рабочей зоны производственного помещения соответствует санитарно-гигиеническим требованиям по параметрам микроклимата, содержанию вредных веществ (газа, пара, аэрозоли) и частиц пыли, приведенным в ГОСТ 12.1.005–88 ССБТ Общие санитарно-гигиенические требования к воздуху рабочей зоны. [5]

В помещениях, оборудованных ВДТ, ЭВМ или ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы с ВДТ, ЭВМ или ПЭВМ.

Уровни положительных и отрицательных аэроионов, а также коэффициент униполярности в воздухе всех помещений, где расположены ВДТ, ЭВМ или ПЭВМ, соответствуют значениям, указанным в таблице 6.3.

Таблица 6.3 – Уровни ионизации и коэффициент униполярности воздуха помещений при работе с ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Уровни | Число ионов в 1 см3 воздуха | | Коэффициент униполярности (У) |
| n+ | n- |
| Минимально допустимые | 400 | 600 | 0,4 ≤ У < 1,0 |
| Оптимальные | 1500-3000 | 3000-5000 |
| Максимально допустимые | 50000 | 50000 |

Одним из мероприятий по оздоровлению воздушной среды является устройство вентиляции и отопления. Задачей вентиляции является обеспечение чистоты воздуха и заданных метеорологических условий на рабочих местах. Чистота воздушной среды достигается удалением загрязненного или нагретого воздуха из помещения и подачей в него свежего воздуха. Работа видеотерминалов сопровождается выделением тепла. Для поддержания нормального микроклимата необходим достаточный объем вентиляции, для чего в вычислительном центре предусматривается кондиционирование воздуха, осуществляющее поддержание постоянных параметров микроклимата в помещении независимо от наружных условий.

### 6.1.4 Освещение

Важное место в комплексе мероприятий по охране труда и оздоровлению условий труда работающих с ЭВМ занимает создание оптимальной световой среды, т.е. рациональная организация естественного и искусственного освещения помещения и рабочих мест.

В соответствии с ТКП 45-2.04-153-2009 [8] устанавливаются нормируемые показатели освещения по таблице 6.4. Нормированный уровень освещенности для работы с ЭВМ принят 400 лк. Для работы в дневное время предусмотрено естественное освещение, боковое одностороннее, отвечающее требованиям ТКП 45-2.04-153-2009 [8].

В помещениях, оборудованных ЭВМ, предусматриваются меры для ограничения слепящего воздействия светопроемов, имеющих высокую яркость (800 кд/м2 и более), и прямых солнечных лучей для обеспечения благоприятного распределения светового потока в помещении и исключения на рабочих поверхностях ярких и темных пятен, засветки экранов посторонним светом. Это достигается путем соответствующей ориентации светопроемов, правильного размещения рабочих мест и использования солнцезащитных средств.

В тех случаях, когда одного естественного освещения недостаточно, устроено совместное освещение. При этом в поле зрения работающих обеспечены оптимальные соотношения яркости рабочих и окружающих поверхностей, исключена или максимально ограничена отраженная блеклость от экрана и клавиатуры в результате отражения в них световых потоков от светильников и источников света.

Расчет искусственного освещения в помещении оборудованном ЭВМ произведем методом коэффициента использования светового потока [9]. При расчете этим методом учитывается как прямой свет от светильника, так и свет, отраженный от стен и потолка:

где – нормированная минимальная освещённость, лк; = 200лк [1, таблица Г.1]; S– площадь освещаемого помещения, ; K - коэффициент запаса; – коэффициент минимальной освещенности (для люминесцентных ламп равен 1,1) [8];  – коэффициент использования осветительной установки; n – необходимое число ламп.

Исходные характеристики помещения вычислительного центра:

* длина (a = 8 м);
* ширина (b = 6 м);
* высота (H = 3 м).

Площадь помещения:

Для помещения оборудованном ЭВМ имеет смысл использовать прямоугольное симметричное расположение светильников, что будет обеспечивать одинаковое освещение оборудования, рабочих мест и проходов.

Определяем показатель помещения:

где а, b — соответственно длина и ширина помещения, м;

– высота расположения светильника над освещаемой поверхностью, м.

где – общая высота помещения, м; – высота от потолка до нижней части светильника, м; – высота от пола до освещаемой поверхности, м.

Чтобы уменьшить ослепляющее действие светильников общего освещения, высоту подвеса их над уровнем пола устанавливают не менее 2,5 – 4 м при лампах мощностью до 200 Вт и не менее 3 – 6 м при лампах большей мощности.

Необходимое число светильников (ламп):

где S – площадь помещения, ; L – расстояние между светильниками, м.

Расстояние между светильниками L может быть найдено из соотношения . В зависимости от типа светильника это соотношение при расположении светильников прямоугольником может быть принято равным 1,4 – 2,0 [2, стр. 127].

Таблица 6.4 – Нормируемые показатели освещения (ТКП 45-2.04-153-2009 [8])

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Помещения | Плоскость нормирова-ния освещено-сти и ее высота над полом, м | Разряд и подраз-ряд  зрите-льной работы | Искусственное освещение | | | | | Естественное освещение КЕО ен, % | | Совмещенное освещение КЕО ен, % | |
| Освещенность рабочих поверхностей, лк | | Цилиндриче-ская освеще-нность, лк | Показа-тель дискомфорта, не более | Коэффициент пульсации освещенности, %, не более, при верхнем или комбиниро-ванном освещении | при верхнем или комби-нирован-ном  освеще-нии | при боко-вом осве-ще-нии | при верхнем или комби-ниро-ванном  освеще-нии | при боко-вом осве-ще-нии |
| при комбини-рованном освещении | при общем осве-щении |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Административные здания (министерства, ведомства, комитеты, префектуры, муниципалитеты, управления,  конструкторские и проектные организации, научно-исследовательские учреждения и т. п.) | | | | | | | | | | | |
| 12. Помещения для работы с дисплеями и видеотерминалами, дисплей­ные залы | В — 1,2 (на экране дисплея) | Б-2 | – | 200 | – | – | – | – | – | – | – |
| Г — 0,8  (на рабочих столах) | А-2 | 500/300 | 400 | – | 15 | 5 | 3,5 | 1,2 | 2,1 | 0,7 |

По найденному показателю помещения и коэффициентам отражения потолка и стен определяем коэффициент использования светового потока  осветительной установки ([2, таблица 4]) (отношение светового потока, падающего на расчетную поверхность к световому потоку источников света). Его величина зависит от КПД и кривой распределения силы света светильника, высоты его подвеса , показателя помещения , коэффициента отражения потолка и стен .

Коэффициент запаса, учитывающий снижение освещенности в процессе эксплуаации осветительной установки, принимаем равным 1,5 ([6, таблица 3]).

При коэффициенте отражения потолка и стен, равном 0,7 и 0,5 соответственно, коэффициент использования светового потока в зависимости от показателя помещения имеет следующие значения (таблица 6.5).

Таблица 6.5 – Коэффициент использования осветительной установки ([2, таблица 4])

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Показатель помещения | 0,5 | 1 | 1,5 | 2 | 3 | 4 | 5 |
|  | 0,22 | 0,37 | 0,44 | 0,48 | 0,54 | 0,59 | 0,61 |

 = 0,44.

Получив все исходные данные, определяем световой поток одной лампы.

(6.5)

По найденному значению светового потока каждой лампы определяем ее мощность по ГОСТ 6825 – 91 и ГОСТ 2239 – 79.

Для выбранного помещения вычислительного центра используются 16 светильников ЛД40 со светоотдачей 58,5 лм/Вт, расположенные прямоугольником.

### 6.1.5 Организация рабочего места

Выделение рабочих мест операторов в отдельный раздел обусловлено целым рядом особенностей их деятельности. При оборудовании рабочего места ПЭВМ согласно [18] учитываются границы поля зрения оператора, которые определяются движениями глаз и головы, они представлены на рисунке 6.1.

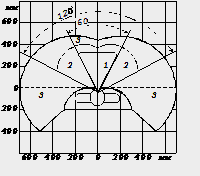


Рисунок 6.1 – Зоны выполнения ручных операций и размещения органов управления

Различают зоны зрительного наблюдения в вертикальной плоскости, ограниченные определенными углами, в которых располагаются дисплей ЭВМ (40-60ºС), пюпитр (35-45ºС) и клавиатура. При организации рабочего места учитываются антропометрические данные операторов, а также предусматривается соответствующее размещение элементов оборудования в зависимости от характера выполняемой работы.

При постоянной работе экран расположен в центре поля обзора, документы слева на столе или на специальной подставке (рисунок 6.2). [18]

Рисунок2

Рисунок 6.2 – Зоны зрительного наблюдения в вертикальной плоскости

Рабочий стол имеет стабильную конструкцию. Плоскость стола выбирают в зависимости от размера документов. При больших размерах документов она достигает 160х90 см. Плоскость стола, а также сидение оператора регулируются по высоте. Высоту плоскости стола регулируют в диапазоне 65-85 см. При этом высота от горизонтальной линии зрения до рабочей поверхности стола при выпрямленной рабочей позе 45-50 см.

Высота сидений от пола регулируется в пределах 42-55 см. По желанию оператора устанавливается подставка для ног размером 40х30х15 см и углом наклона 0-20ºС с нескользящим покрытием и неперемещаемая по полу. [18]

Планировка рабочего места должна удовлетворять требованиям удобства выполнения работ и экономии энергии оператора, рационального использования площадей и удобства обслуживания устройств ЭВМ.

### 6.1.7 Излучение

При работе с дисплеем могут возникнуть следующие опасные факторы: электромагнитные поля, электростатические поля, рентгеновское излучение, ультрафиолетовое и инфракрасное излучение.

Уровни физических факторов, создаваемые ВДТ, ЭВМ, ПЭВМ и периферийными устройствами, не превышают предельно-допустимые уровни: электромагнитных и электростатических полей (таблицы 6.6, 6.7), ультрафиолетового (таблица 6.6), установленных Гигиеническим нормативом «Предельно-допустимые уровни нормируемых параметров при работе с видеодисплейными терминалами и электронно-вычислительными машинами». [3]

Таблица 6.6 – Предельно допустимые уровни электромагнитных полей от экранов ВДТ, ЭВМ и ПЭВМ

|  |  |  |
| --- | --- | --- |
| Наименование параметра | | Предельно-допустимые уровни |
| 1 | | 2 |
| Напряженность электрического поля в диапазоне частот: | |  |
|  | 5 Гц-2 кГц | не более 25,0 В/м |
|  | 2-400 кГц | не более 2,5 В/м |
| Плотность магнитного потока магнитного поля в диапазоне частот: | |  |
|  | 5 Гц-2 кГц | не более 250 нТл |
|  | 2-400 кГц | не более 25 нТл |
| Напряженность электростатического поля | | не более 15 кВ/м |

Таблица 6.7 – Предельно допустимые уровни электромагнитных полей при работе с ВДТ, ЭВМ, ПЭВМ от клавиатуры, системного блока, манипулятора «мышь», беспроводных системам передачи информации и иных периферийных устройств

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Диапазоны частот | 0,3-300  кГц | 0,3-3  МГц | 3-30  МГц | 30-300  МГц | 0,3-300  ГГц |
| Предельно допустимые уровни | 25 В/м | 15 В/м | 10 В/м | 3 В/м | 10 мкВт/см2 |

Таблица 6.8 – Предельно допустимые уровни интенсивности излучения в ультрафиолетовом диапазоне на расстоянии 0,5 м со стороны экрана ВДТ, ЭВМ и ПЭВМ

|  |  |  |  |
| --- | --- | --- | --- |
| Диапазоны длин волн | 200-280 нм | 280-315 нм | 315-400 нм |
| Предельно допустимые уровни | не допускается | 0,0001 Вт/м2 | 0,1 Вт/м2 |

Наиболее эффективным и часто применяемым методом защиты от электромагнитных излучений является установка экранов. Экранируют либо источник излучения, либо рабочее место. Часто экран устанавливают непосредственно на монитор.

## 6.2 Техника безопасности

Эксплуатация вычислительной техники связана с применением электрической энергии. Опасность поражения электрическим током возникает при прикосновении к открытым токоведущим частям с нарушенной изоляцией или к оборудованию, находящемуся под напряжением при отсутствии или нарушении изоляции. По степени поражения людей электрическим током вычислительный центр относится к классу помещений без повышенной опасности.

Основными нормативными документами по защите от поражения электротоком являются «Правила устройства электроустановок, ПУЭ» [10], «Правила техники безопасности при эксплуатации электроустановок» [11].

Основные меры защиты от поражения током:

* защитное заземление (Rз=4 Ом);
* зануление;
* защитное отключение;
* изоляция токоведущих частей и устройств контроля;
* ограждение и недоступность токоведущих частей;
* электрическое разделение сети с помощью специальных разделяющих трансформаторов;
* применение малого напряжения; использование двойной изоляции.

Для участка персональных компьютеров наиболее приемлемым вариантом является защитное заземление, т.к. корпуса компьютеров и периферии обычно выполнены не из токопроводящих материалов, а также имеются специальные клеммы для подключения заземления. [11]

Опасность возникновения статического электричества проявляется в возможности образования электрической искры и вредном действии его на организм человека.

ПДУ напряженности ЭП (50 Гц) на рабочем месте в течение всей смены устанавливается равным 5 кВ/м. Пребывание работников в ЭП (50 Гц) с уровнем напряженности, не превышающим 5 кВ/м, допускается в течение всего рабочего дня (СанПиН № 69 от 21.06.2010 г). [13]

Организацию работ по управлению оборудованием и надзор за их выполнением осуществляет административно-технический персонал участка, где выполняются эти работы. Группа по ТБ этого персонала не ниже IV.

Группа I по электробезопасности присваивается неэлектрическому персоналу – техники и операторы ЭВМ. Группа II – электромонтеры, электрослесари, математики-программисты, механики ЭВМ, практиканты институтов. Группа III, IV,V – старшие инженеры, электромонтеры, оперативный и ремонтный персонал вычислительного центра, в зависимости от стажа работы [14].

Основные способы защиты от статического электричества:

* заземление оборудования;
* увеличение поверхностей проводимости диэлектриков;
* увлажнение окружающего воздуха;
* ионизация воздуха или среды нейтрализаторами статического электричества.

Для оказания первой помощи пострадавшему от электрического тока необходимо быстро отключить оборудование, которого касается пострадавший, определить состояние пострадавшего и выбрать меры первой помощи.

## 6.3 Пожарная безопасность

В современных ЭВМ имеется высокая плотность размещения элементов электронных схем. При протекании по ним электрического тока выделяется значительное количество теплоты. Для отбора избыточной теплоты от ЭВМ служат системы вентиляции и кондиционирования воздуха

Так как в процессе работы используются горючие вещества и материалы (бумага, порошковые картриджи для множительной и оргтехники), то помещения с ЭВМ относим к категории Д пожароопасности (ТКП 474-2013). [15]

Учитывая высокую стоимость электронного оборудования, а также категорию пожарной опасности помещения, здания для вычислительных центров и части зданий другого назначения, в которых предусмотрено размещение ЭВМ, относятся ко второй степени огнестойкости в соответствии с ТКП 45-2.02-279-2013. [17]

Для предотвращения распространения огня во время пожара с одной части здания на другую устраивают противопожарные преграды в виде стен, перегородок, дверей, окон, люков, клапанов. Все виды кабелей прокладываются в металлических коробах до распределительных щитов или стоек питания.

Для ликвидации пожаров в начальной стадии применяются первичные средства пожаротушения: внутренние пожарные водопроводы, огнетушители типа ОВП-10, ОУ-2, сухой песок, асбестовые одеяла и др. В здании вычислительного центра краны устанавливают в коридорах, на площадках лестничных клеток, у входа, т.е. в доступных и защитных местах. На каждые 100 квадратных метров пола обычно требуется один-два огнетушителя.

Эвакуационный путь из рабочего помещения проходит через дверь (ширина двери не менее 1 м) по коридору (ширина не менее 1.8 м) через лестничную клетку (ширина марша не менее двух метров) наружу, что отвечает требованиям ТКП 45-2.02-22-2006. [19]

# **ЗАКЛЮЧЕНИЕ**

В данном дипломном проекте разработана подсистема графического построения прмитивов для автоматизации обучения начертательной геометрии. Для этого был произведен обзор литературы по теме, выделены особенности автоматизации обучения начертательной геометрии, проанализированы аналогичные программные продукты, выстроены требования к функционалу графического редактора, спроектированы и реализованы модули графического редактора и библиотеки графических примитивов, рассчитана экономическая эффективность.

На начальном этапе проектирования были выделены задачи, которые должна выполнять подсистема. Далее были определены требования необходимые для реализации поставленных задач.

В соответствии с поставленными задачами разработка была разделена на этапы. Разработанная подсистема предоставляет следующие возможности:

- дружественный и интуитивно понятный пользовательский интерфейс;

- выбор различных задач и вариантов для выполнения работы;

- построение стандартных элементов и разработанных для подсистемы;

- контроль над целостностью, корректностью и непротиворечивостью вводимых данных, а также возможности предотвратить попытку ввода некорректных данных;

- проверка выполненной задачи на ошибки;

- возможность легкого расширения и наращивания функциональности, благодаря трехуровневой архитектуре проектирования.

В ходе выполнения апробации программный продукт показал достаточно стабильные результаты работы. Было установлено, что все реализованные функции подсистемы успешно работаю7т в стандартном режиме эксплуатации, а также при различных несанкционированных действиях пользователя. Таким образом, поставленная задача выполнена в полном объеме.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Полозков, Ю. В., Концепция интерактивного программного комплекса обучения и контроля знаний по начертательной геометрии / Ю. В. Полозков, В. И. Луцейкович // Вестник ПГУ. Сер. Е, Педагогические науки. – 2015. – № 15. – С. 48–56.
2. Замжицкий, О.С. Обеспечение импорта графической информации для автоматизации проверки заданий электронной рабочей тетради по начертательной геометрии / О.С. Замжицкий, Ю. В. Полозков // 47 Международная научно-техническая конференция преподавателей и студентов университета : тезисы докладов Международной науч.-техн. конф., Витебск, 2014 г. / УО “ВГТУ”; редкол.: Е. В. Ванкевич [и др.]. – Витебск, 2014. – С. 143°– 144.
3. Карабчевский, В. В. Визуальная среда разработки систем геометрического моделирования / В. В. Карабчевский, И. В. Хлепитько // Наукові праці Донецького національного технічного університету. Сер. «Інформатика, кібернетика та обчислювальна техніка». – Донецк, 2009. – Вып. 10 (153). – С. 146-151.
4. Бобровских, А. С. Разработка обучающей системы для алгоритмов начертательной геометрии с ис-пользованием технологии автоматного программирования / А. С. Бобровских // Вестник компьютер-ных и информационных технологий [Электронный ресурс]. / Ярославский гос. ун-т им. П. Г. Демидова. – Ярославль, 2011. – № 1. – С. 38 – 43. – Режим доступа: http://www.ugatu.ac.ru/ddo/reclama/elz-ng-1. - Дата доступа: 25.11.2012.
5. Подберезкин, А.С. Автоматизация проверки выполнения графических задач в CAD системе / Подберезкин А.С., Колядко С.В. Полозков Ю.В. // Материалы 72-й студенческой науч.-техн. конф., Минск, 2016 г. / БНТУ; редкол.: Е.Е. Трофименко [и др.]. [Электронный ресурс] : – Минск, 2016. – С. 146 – 152.
6. Isotani, S. An algorithm for automatic checking of exercises in a dynamic geometry system: iGeom / S. Isotani, L.O. Brandao // Computers & Education. – 2008 – Vol. 51. – P. 1283–1303.
7. Pham, T.M. A Combination of a Dynamic Geometry Software With a Proof Assistant for Interactive For-mal Proofs / T.M. Pham, Y. Bertot // Electronic Notes in Theoretical Computer Science. – 2012. – Vol. 285. – P. 43–55.
8. Задруцкий, С. А. Решение задач по начертательной геометрии графоаналитическим способом с применением ПЭВМ : практикум для студентов всех специальностей БГУИР / С. А. Задруцкий, А. А. Резанко, В. А. Столер. – Минск : БГУИР, 2003. – 49 с.
9. Козинец, Д. Г. Программно-методический комплекс поддержки процесса графической подготовки и контроля знаний по начертательной геометрии / Д. Г. Козинец, Ю. В. Полозков // Инновационные технологии в инженерном образовании : материалы международной научно-практической конференции, Минск, 27-28 апреля 2011 г. / Белорус. нац. техн. ун-т ; под ред. В. Л. Соломахо. – Минск, 2011. – С. 247 – 249.
10. .NET Framework 4.5 и 4.6 [Электронный ресурс] – Режим доступа: https://msdn.microsoft.com/ru-ru/library/w0x726c2(v=vs.110).aspx, свободный. – Загл. с экрана.
11. .NET Framework [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/.NET\_Framework, свободный. – Загл. с экрана.
12. Дейтел Х., Дейтел П., Листфилд Дж., Нието Т., Йегер Ш, Златкина М. С# в подлиннике.– СПб.: БХВ-Петербург, 2006.
13. Язык C# и платформа.NET [Электронный ресурс] – Режим доступа: <http://metanit.com/sharp/tutorial/1.1.php>, свободный. – Загл. с экрана.
14. Рихтер Дж. Программирование на платформе Microsoft .NET Framework.– M.: Русская Редакция, 2003.
15. В.П. Семич, А.В. Охрана труда при работе на персональных электронно-вычислительных машинах и другой офисной технике: Мн.: ЦОТЖ, 2004.
16. СанПиН №59 2013. Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами. Минск, 2013.
17. СанПиН №33 2013г Гигиенические требования к микроклимату производственных и офисных помещений.
18. СанПиН №115 2011. Шум на рабочих местах, в транспортных средствах, в помещениях жилых, общественных зданий и на территории жилой застройки. Минск, 2011.
19. СанПиН № 132. Требования к производственной вибрации, вибрации в жилых помещениях, помещениях административных и общественных зданий. Минск, 2013.
20. ТКП 45-2.04-153-2009 Естественное и искусственное освещение. Строительные нормы проектирования.
21. ГОСТ 12.2.032-78. ССБТ. Рабочее место при выполнении работ сидя.
22. ГОСТ 12.1.045-84 2011. ССБТ. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля. Минск, 2011.
23. Правила устройства электроустановок. Минэнерго СССР – М.: Электроатомиздат, 1986.
24. Правила технической эксплуатации электроустановок потребителей и правила техники безопасности при эксплуатации электроустановок потребителей. Минэнерго СССР – М.: Энергоатомиздат, 1986.
25. ТКП 474-2013 Категорирование зданий, помещений и наружных установок по взрывопожарной и пожарной опасности.
26. СниП 2.01.02-85. Противопожарные нормы. – М.: Стройиздат, 1985.
27. ТКП 45-2.02-142-2011 Здания, строительные конструкции, материалы и изделия. Правила пожарно-технической классификации.
28. ТКП 45-2.02-279-2013 (02250) Здания и сооружения. Эвакуация людей при пожаре. Строительные нормы проектирования.

# ПРИЛОЖЕНИЕ 1. Класс Storage

public class Storage

{

public Storage()

{

Objects = new List<IObject>();

SelectedObjects = new List<IObject>();

CopiedObjects = new List<IObject>();

PastedObjects = new List<IObject>();

DeletedObjects = new List<IObject>();

TempObjects = new List<IObject>();

TempLinesOfPlane = new List<IObject>();

}

public Storage(IList<IObject> objects)

{

Objects = objects;

SelectedObjects = new List<IObject>();

CopiedObjects = new List<IObject>();

PastedObjects = new List<IObject>();

DeletedObjects = new List<IObject>();

TempObjects = new List<IObject>();

TempLinesOfPlane = new List<IObject>();

}

public void ClearAllCollections()

{

Objects.Clear();

SelectedObjects.Clear();

CopiedObjects.Clear();

PastedObjects.Clear();

DeletedObjects.Clear();

TempObjects.Clear();

}

public void AddToCollection(IObject source)

{

if (source != null)

{

Objects.Add(source);

}

}

/// <summary>

/// Отрисовывает все коллекции объектов

/// </summary>

/// <param name="st">Настройки</param>

/// <param name="frameCenter">Центр системы координат</param>

/// <param name="g">Целевой Graphics</param>

public void DrawObjects(Settings st, Point frameCenter, Graphics g)

{

foreach(var ob in Objects)

{

ob.Draw(st.DrawSettings, frameCenter, g);

}

foreach (var ob in TempObjects)

{

ob.Draw(st.DrawSettings, frameCenter, g);

}

foreach (var ob in SelectedObjects)

{

ob.Draw(st.SelectedDrawSettings, frameCenter, g);

}

foreach (var ob in DeletedObjects)

{

ob.Draw(st.SelectedDrawSettings, frameCenter, g);

}

}

/// <summary>

/// Отрисовывает последний добавленный объект в коллекцию объектов

/// </summary>

/// <param name="st"></param>

/// <param name="frameCenter"></param>

/// <param name="g"></param>

public void DrawLastAddedToObjects(DrawSettings st, Point frameCenter, Graphics g)

{

Objects.Last().Draw(st, frameCenter, g);

}

/// <summary>

/// Отрисовывает последний добавленный объект в коллекцию временных объектов

/// </summary>

/// <param name="st"></param>

/// <param name="frameCenter"></param>

/// <param name="g"></param>

public void DrawLastAddedToTempObjects(DrawSettings st, Point frameCenter, Graphics g)

{

TempObjects.Last().Draw(st, frameCenter, g);

}

public void ClearTempCollections()

{

TempObjects.Clear();

TempLinesOfPlane.Clear();

}

public IList<IObject> Objects { get; }

public IList<IObject> SelectedObjects { get; }

public IList<IObject> CopiedObjects { get; }

public IList<IObject> PastedObjects { get; }

public IList<IObject> DeletedObjects { get; }

public IList<IObject> TempObjects { get; }

public IList<IPointOfPlane> TempPointsOfPlane { get; }

public IList<IObject> TempLinesOfPlane { get; }

# }

# ПРИЛОЖЕНИЕ 2. Группа основных интерфейсов библиотеки графических примитивов

public interface IObject

{

Name Name { get; set; }

void Draw(DrawSettings st, Point frameCenter, Graphics graphics);

bool IsSelected(Point mousecoords, float ptR, Point frameCenter, double distance);

}

public interface IObjectOfPlane1X0Y : IObject

{

double X { get; }

double Y { get; }

}

public interface IObjectOfPlane2X0Z : IObject

{

double X { get; }

double Z { get; }

}

public interface IObjectOfPlane3Y0Z : IObject

{

double Y { get; }

double Z { get; }

}

# ПРИЛОЖЕНИЕ 3. Класс Calculate

public static class Calculate

{

#region Distance

public static double Distance(Point mscoords, Point pt)

{

return Math.Sqrt(Math.Pow((mscoords.X - pt.X), 2) + Math.Pow((mscoords.Y - pt.Y), 2));

}

public static double Distance(Point mscoords, Point2D pt)

{

return Math.Sqrt(Math.Pow((mscoords.X - pt.X), 2) + Math.Pow((mscoords.Y - pt.Y), 2));

}

public static double Distance(Point mscoords, float ptR, Point frameCenter, PointOfPlane1X0Y pt)

{

var dpt = DeterminePosition.ForPointProjection(pt, ptR, frameCenter);

return Distance(mscoords, dpt);

}

public static double Distance(Point mscoords, float ptR, Point frameCenter, PointOfPlane2X0Z pt)

{

var dpt = DeterminePosition.ForPointProjection(pt, ptR, frameCenter);

return Distance(mscoords, dpt);

}

public static double Distance(Point mscoords, float ptR, Point frameCenter, PointOfPlane3Y0Z pt)

{

var dpt = DeterminePosition.ForPointProjection(pt, ptR, frameCenter);

return Distance(mscoords, dpt);

}

public static double[] Distance(Point mscoords, float ptR, Point frameCenter, Point3D pt)

{

return new[] { Distance(mscoords, ptR, frameCenter, pt.PointOfPlane1X0Y),

Distance(mscoords, ptR, frameCenter, pt.PointOfPlane2X0Z),

Distance(mscoords, ptR, frameCenter, pt.PointOfPlane3Y0Z)};

}

#endregion

#region Crossing

public static PointF CrossingPoint(Line2D ln1, Line2D ln2)

{

var y = (ln2.Point0.Y \* ln2.kx \* ln1.ky - ln1.Point0.Y \* ln2.ky \* ln1.kx + ln2.ky \* ln1.ky \* (ln1.Point0.X - ln2.Point0.X)) /

(ln2.kx \* ln1.ky - ln1.kx \* ln2.ky);

var x = ln1.kx \* (y - ln1.Point0.Y) / ln1.ky + ln1.Point0.X;

return new PointF((float)x, (float)y);

}

public static PointF CrossingPoint(Line2D ln1, LineOfPlane1X0Y ln, Point frameCenter)

{

var ln2 = DeterminePosition.ForLineProjection(ln, frameCenter);

var y = (ln2.Point0.Y \* ln2.kx \* ln1.ky - ln1.Point0.Y \* ln2.ky \* ln1.kx + ln2.ky \* ln1.ky \* (ln1.Point0.X - ln2.Point0.X)) /

(ln2.kx \* ln1.ky - ln1.kx \* ln2.ky);

var x = (ln1.Point0.X \* ln2.kx \* ln1.ky - ln2.Point0.X \* ln1.kx \* ln2.ky + ln2.kx \* ln1.kx \* (ln2.Point0.Y - ln1.Point0.Y)) /

(ln1.ky \* ln2.kx - ln1.kx \* ln2.ky);

return new PointF((float)x, (float)y);

}

public static PointF CrossingPoint(Line2D ln1, LineOfPlane2X0Z ln, Point frameCenter)

{

var ln2 = DeterminePosition.ForLineProjection(ln, frameCenter);

var y = (ln2.Point0.Y \* ln2.kx \* ln1.ky - ln1.Point0.Y \* ln2.ky \* ln1.kx + ln2.ky \* ln1.ky \* (ln1.Point0.X - ln2.Point0.X)) /

(ln2.kx \* ln1.ky - ln1.kx \* ln2.ky);

var x = (ln1.Point0.X \* ln2.kx \* ln1.ky - ln2.Point0.X \* ln1.kx \* ln2.ky + ln2.kx \* ln1.kx \* (ln2.Point0.Y - ln1.Point0.Y)) /

(ln1.ky \* ln2.kx - ln1.kx \* ln2.ky);

return new PointF((float)x, (float)y);

}

public static PointF CrossingPoint(Line2D ln1, LineOfPlane3Y0Z ln, Point frameCenter)

{

var ln2 = DeterminePosition.ForLineProjection(ln, frameCenter);

var y = (ln2.Point0.Y \* ln2.kx \* ln1.ky - ln1.Point0.Y \* ln2.ky \* ln1.kx + ln2.ky \* ln1.ky \* (ln1.Point0.X - ln2.Point0.X)) /

(ln2.kx \* ln1.ky - ln1.kx \* ln2.ky);

var x = (ln1.Point0.X \* ln2.kx \* ln1.ky - ln2.Point0.X \* ln1.kx \* ln2.ky + ln2.kx \* ln1.kx \* (ln2.Point0.Y - ln1.Point0.Y)) /

(ln1.ky \* ln2.kx - ln1.kx \* ln2.ky);

return new PointF((float)x, (float)y);

}

#endregion

}

# ПРИЛОЖЕНИЕ 4. Класс Plane2D

public class Plane2D : IPlaneOfPlane

{

public IObject[] Objects;

private Name \_name;

public Plane2D(Point2D[] pts)

{

Objects = new IObject[pts.Length];

Array.Copy(pts, Objects, pts.Length);

\_name = new Name();

}

public Plane2D(Line2D ln1, Point2D pt1)

{

Objects = new IObject[] {ln1, pt1};

\_name = new Name();

}

public Plane2D(Line2D ln1, Line2D ln2)

{

Objects = new IObject[] { ln1, ln2 };

\_name = new Name();

}

public Plane2D(Segment2D sg1, Point2D pt1)

{

Objects = new IObject[] { sg1, pt1 };

\_name = new Name();

}

public Plane2D(Segment2D sg1, Segment2D sg2)

{

Objects = new IObject[] { sg1, sg2 };

\_name = new Name();

}

public void Draw(DrawSettings settings, Point frameCenter, Graphics graphics)

{

foreach (var obj in Objects)

{

obj.Draw(settings, frameCenter, graphics);

}

}

public bool IsSelected(Point mousecoords, float ptR, Point frameCenter, double distance)

{

return Objects.Any(obj => obj.IsSelected(mousecoords, ptR, frameCenter, distance));

}

public Name Name

{

get

{

return \_name;

}

set

{

\_name = value;

foreach (var t in Objects)

{

t.Name = \_name;

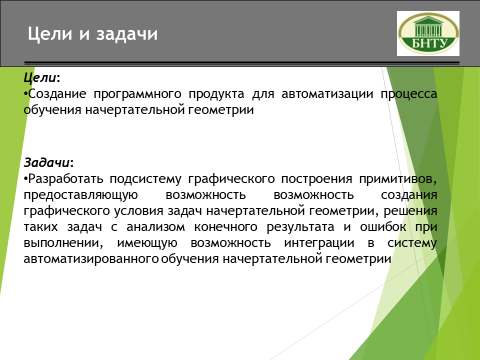
}

}

}

# ГРАФИЧЕСКИЙ МАТЕРИАЛ





Изм.

Лист

№ докум.

Подпись

Дата

Лист

1

ДП – 10702412/10 – ДО - 2017

Разраб.

Колядко С.В.

Руковод.

Шваякова Е.А.

Н. Контр.

Шваякова Е.А.

Зав. Каф.

Бородуля А.В.

Подсистема графического построения примитивов для автоматизации обучения начертательной геометрии

Лит.

Листов

6

БНТУ, г. Минск

Изм.

Лист

№ докум.

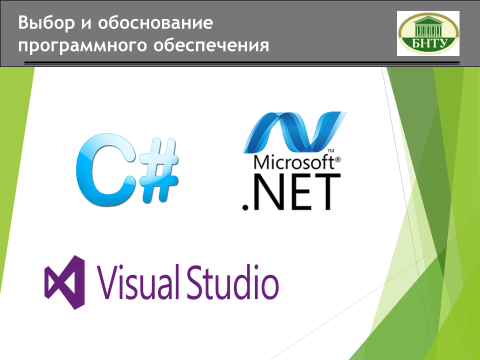
Подпись

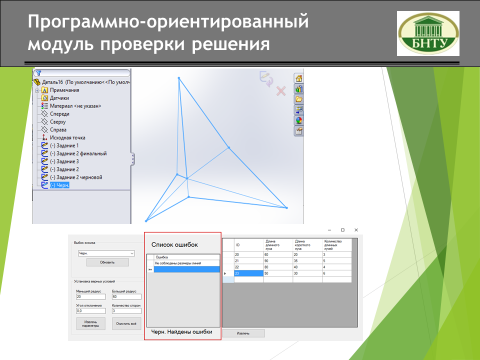
Дата

Лист

2

ДП – 10702412/10 – ДО - 2017





Изм.

Лист

№ докум.

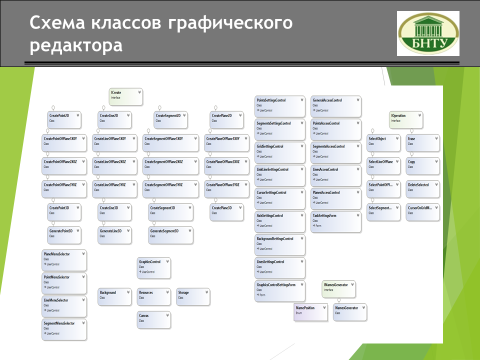
Подпись

Дата

Лист

3

ДП – 10702412/10 – ДО - 2017



Изм.

Лист

№ докум.

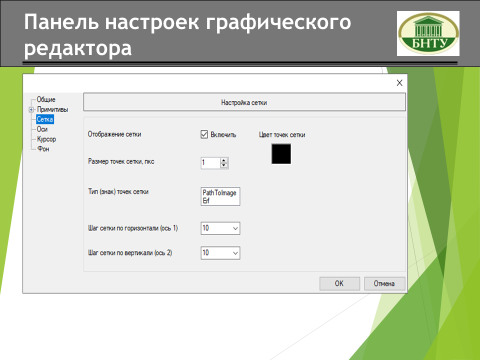
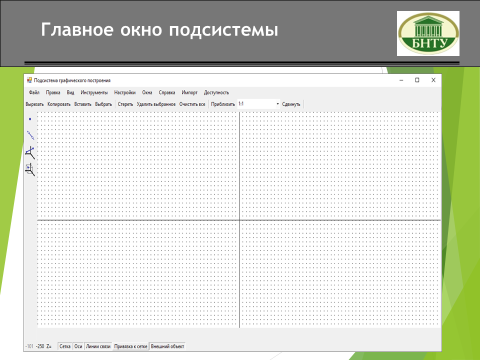
Подпись

Дата

Лист

4

ДП – 10702412/10 – ДО - 2017



Изм.

Лист

№ докум.

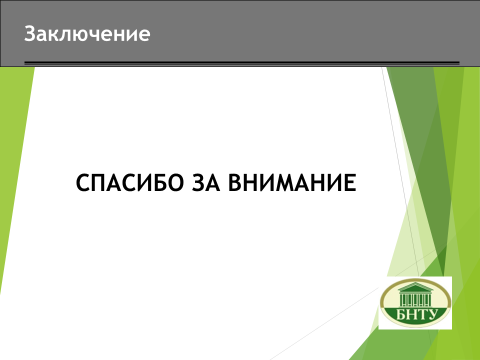
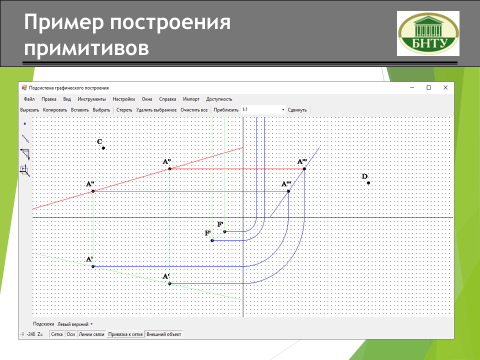
Подпись

Дата

Лист

5

ДП – 10702412/10 – ДО - 2017



Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

ДП – 10702412/10 – ДО - 2017