

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ  
АКАДЕМИЯ»

«УТВЕРЖДАЮ»

Проректор по учебной работе

« 31 » марта 2021 г.

 Г.Ю. Нагорная



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Проектирование программного обеспечения

Уровень образовательной программы \_\_\_\_\_ бакалавриат \_\_\_\_\_

Направление подготовки \_\_\_\_\_ 01.03.04 Прикладная математика \_\_\_\_\_

Направленность (профиль) \_\_\_\_\_ общий \_\_\_\_\_

Форма обучения \_\_\_\_\_ очная \_\_\_\_\_

Срок освоения ОП \_\_\_\_\_ 4 года \_\_\_\_\_

Институт \_\_\_\_\_ Прикладной математики и информационных технологий \_\_\_\_\_

Кафедра разработчик РПД \_\_\_\_\_ Математика \_\_\_\_\_


Выпускающая кафедра \_\_\_\_\_ Математика \_\_\_\_\_

Начальник  
учебно-методического управления \_\_\_\_\_ 

Семенова Л.У.

Директор института ПМ и ИТ \_\_\_\_\_ 

Тебуев Д.Б.

Заведующий выпускающей кафедрой \_\_\_\_\_ 

Кочкаров А.М.

г. Черкесск, 2021 г.

## СОДЕРЖАНИЕ

<b>1</b>	<b>Цели освоения дисциплины</b>	3
<b>2</b>	<b>Место дисциплины в структуре образовательной программы</b>	4
<b>3</b>	<b>Планируемые результаты обучения по дисциплине</b>	5
<b>4</b>	<b>Структура и содержание дисциплины</b>	6
	4.1. Объем дисциплины и виды учебной работы	6
	4.2. Содержание дисциплины	7
	4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля	7
	4.2.2. Лекционный курс	8
	4.2.3. Лабораторный практикум	11
	4.2.4. Практические занятия	11
	4.3. Самостоятельная работа обучающегося	12
<b>5</b>	<b>Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине</b>	13
<b>6</b>	<b>Образовательные технологии</b>	16
<b>7</b>	<b>Учебно-методическое и информационное обеспечение дисциплины</b>	17
	7.1. Перечень основной и дополнительной учебной литературы	17
	7.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	18
	7.3. Информационные технологии, лицензионное программное обеспечение	19
<b>8</b>	<b>Материально-техническое обеспечение дисциплины</b>	20
	8.1. Требования к аудиториям (помещениям, местам) для проведения занятий	20
	8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся:	20
	8.3. Требования к специализированному оборудованию	20
<b>9</b>	<b>Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья</b>	21
	<b>Приложение 1. Фонд оценочных средств</b>	23
	<b>Приложение 1. Фонда оценочных средств</b>	52
	<b>Приложение 2. Аннотация рабочей программы</b>	53
	<b>Рецензия на рабочую программу</b>	54

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**Цель освоения дисциплины:** Получение обучающимися систематизированных сведений о технологиях разработки программного обеспечения; знакомство с Российскими и международными стандартами разработки программных средств и перспективными направлениями развития технологии разработки ПО; получение навыков по разработке сложных программных средств; приобретение обучающимися знаний по современным методам построения, основам функционирования программных средств различного назначения; получение первичных навыков системного подхода к их проектированию; изучение методов организации работы в коллективах разработчиков программного обеспечения.

### **Задачи дисциплины:**

- изложение основных положений технологии разработки ПО, формулировка практических рекомендаций по организации работы коллективов программистов, руководства такими коллективами;
- формирование у обучающихся знаний по дисциплине, связанных с процессом разработки ПО, включая связи с предметной областью, реализацию, организацию производства, контроль за сроками исполнения и качеством;
- ознакомление с техническими программными и технологическими решениями, используемыми при разработке ПО;
- формирование навыков проектирования, реализации, оценки качества и анализа эффективности программного обеспечения;
- приобретение практических навыков работы в коллективе программистов, умения находить правильные технологические решения по выбору структуры программного проекта, методов тестирования и контроля исполнения использование современных инструментальных и методологических средств;

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Дисциплина «Проектирование программного обеспечения» относится к части, формируемой участниками образовательных отношений блока Б1 Дисциплины (Модули), имеет тесную связь с другими дисциплинами.

2.2. В таблице приведены предшествующие и последующие дисциплины, направленные на формирование компетенций дисциплины в соответствии с матрицей компетенций ОП.

### **Предшествующие и последующие дисциплины, направленные на формирование компетенций**

№ п/п	Предшествующие дисциплины	Последующие дисциплины
	Экономика Основы проектной деятельности Мобильные сети и технологии Объектно – ориентированное программирование	Преддипломная практика

### 3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты освоения образовательной программы (ОП) – компетенции обучающихся определяются требованиями стандарта по направлению подготовки 01.03.04 Прикладная математика и формируются в соответствии с матрицей компетенций ОП

№ п/п	Номер/ индекс компетенции	Наименование компетенции (или ее части)	Индикаторы достижения компетенций
1	2	3	4
1.	<b>ПК-3</b>	Способен ориентироваться в современных алгоритмах компьютерной математики, обладать способностями к эффективному применению и реализации математически сложных алгоритмов в современных программных комплексах	ПК-3.1 Осуществляет выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов ПК-3.2 Способен практически реализовать алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи ПК-3.3 Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

##### 4.1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы		Всего часов	Семестры*
			6 семестр часов
1		2	3
<b>Аудиторная контактная работа (всего)</b>		68	68
В том числе:			
Лекции (Л)		34	34
Практические занятия (ПЗ), Семинары (С) в том числе, практическая подготовка		18	18
Лабораторные работы (ЛР) в том числе: индивидуальные и групповые консультации		16	16
<b>Контактная внеаудиторная работа</b> в том числе: индивидуальные и групповые консультации		2	2
индивидуальные и групповые консультации		2	2
<b>Самостоятельная работа обучающегося (СР) (всего)</b>		38	38
<i>Работа с книжными источниками</i>		5	5
<i>Работа с электронными источниками</i>		10	10
<i>Подготовка к коллоквиуму</i>		5	5
<i>Реферат</i>		5	5
<i>Подготовка к лабораторному практикуму</i>		5	5
<i>Подготовка к тестированию</i>		8	8
<b>Промежуточная аттестация</b>	экзамен (Э) <b>в том числе:</b>	Э(36)	Э (36)
	Прием экз., час	0,5	0,5
	Консультация, час.	2	2
	СРО, час.	33,5	33,5
<b>ИТОГО: Общая трудоемкость</b>	<b>Часов</b>	144	144
	<b>зач. ед.</b>	4	4

## 4.2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.2.1. Разделы (темы) дисциплины, виды учебной деятельности и формы контроля

№ п/ п	Наименование раздела (темы) дисциплины	Виды учебной деятельности, включая самостоятельную работу обучающихся (в часах)					Формы текущей и промежуточ ой аттестации
		Л	ЛР	ПЗ	СР О	все го	
1	2	3	4	5	6	7	8
<b>Семестр 6</b>							
1.	Раздел 1. Основные сведения о программном обеспечении	4	-	-	9	13	Коллоквиум, контрольные вопросы, индивидуальные задания к лабораторным работам
2.	Раздел 2. Жизненный цикл программного обеспечения	14	4	4	9	31	Коллоквиум, контрольные вопросы, индивидуальные задания к лабораторным работам
3.	Раздел 3. CASE-технологии при создании программного обеспечения	10	8	10	10	38	Коллоквиум, контрольные вопросы, индивидуальные задания к лабораторным работам, реферат, тестирование
4.	Раздел 4. Международные стандарты проектирования	6	4	4	10	24	Коллоквиум, контрольные вопросы, индивидуальные задания к лабораторным работам, тестирование
5	Контактная внеаудиторная работа					2	индивидуальные и групповые консультации
	<i>Промежуточная аттестация</i>					36	<i>Экзамен</i>
<b>ИТОГО:</b>		34	16	18	38		<b>144</b>

#### 4.2.2. Лекционный курс

№ п/п	Наименование раздела дисциплины	Наименование темы лекции	Содержание лекции	Всего часов
1	2	3	4	5
<b>Семестр 6</b>				
1.	Раздел 1. Основные сведения о программном обеспечении	Тема 1.1 Цели и задачи технологий разработки ПО.	История и предпосылки создания ПО. Цели ,задачи разработки программного обеспечения. Современные технологии создания ПО. Обзор	<b>2</b>
2.	Раздел 1. Основные сведения о программном обеспечении	Тема 1.2 Особенности современных крупных проектов ИС	Современные крупные проекты ИС характеризуются. Классификация, функциональные возможности. Факторы, которые способствовали развитию в области методологии программирования ИС. Основные различия в области программирования (структурного, визуального, модульного). Введение в процесс разработки программного обеспечения. Понятие программного продукта.	<b>2</b>
3.	Раздел 2. Жизненный цикл программного обеспечения	Тема 2.1 Программные средства. Программное обеспечение (ПО).	Программное обеспечение (Software) - полный набор или часть программ, процедур, правил и связанной с ними документации системы обработки информации. (ИСО/МЭК 2382-1 1993) Программные средства (Software product) -набор компьютерных программ, процедур и, возможно, связанных с ними документации и данных. Объем понятия, выражаемого термином "программные средства" включает в себя как частный случай объем понятия 'программное обеспечение".определяемого по ГОСТ 19781. [см. ГОСТ 28806-90. приложение 1 ] Программный продукт (Software product) - набор компьютерных программ, процедур и, возможно, связанных с ними документации и данных, предназначенных для передачи пользователю [ИСО/МЭК 12207]. Примечание. Продукты	<b>2</b>



			<p>включают промежуточные продукты и продукты, предназначенные для пользователей типа разработчиков и персонала сопровождения</p> <p>Проектирование программного обеспечения представляет собой процесс построения приложений реальных размеров и практической значимости, удовлетворяющих заданным требованиям функциональности и производительности, таких, например, как текстовый редактор, электронная таблица, операционная система или, скажем, программа контроля неисправностей космической станции. Программирование - это один из видов деятельности, входящих в цикл разработки программного обеспечения.</p>	
4.	Раздел 2. Жизненный цикл программного обеспечения	Тема 2.2 Жизненный цикл (ЖЦ) ПО.	<p>Понятие ЖЦ, нормативные документы регламентирующие ЖЦ программного продукта. Структура ЖЦ ПО. Основные структурные этапы ЖЦ ПО. Перечень эксплуатационных документов.</p> <p>Конфигурирование баз данных и рабочих мест. Модификация ПО.</p>	<b>4</b>
5.	Раздел 2. Жизненный цикл программного обеспечения	Тема 2.3 Процессы ЖЦ ПО	<p>Верификация. Вопросы идентификации ПО.</p> <p>Управление конфигурацией , процессы разработки и сопровождения ПО.</p> <p>Организация, систематический учет и контроль внесения изменений в ПО на всех стадиях ЖЦ. Общие принципы и рекомендации конфигурационного учета, планирования и управления конфигурациями ПО, которые отражены в проекте стандарта ISO 12207-2 [5].</p>	<b>4</b>
6.	Раздел 2. Жизненный цикл программного обеспечения	Тема 2.4 Модели ЖЦ ПО.	Каскадная модель ЖЦ ПО, спиральная модель. Основные понятия, методология	<b>4</b>

			создания. Международные стандарты создания моделей ЖЦ. Основные отличия этих моделей.	
7.	Раздел 3. CASE-технологии при создании программного обеспечения.	Тема 3.1 Классификация CASE- средств	Современные CASE – средства, назначение, виды, классификация	<b>6</b>
8.	Раздел 3. CASE-технологии при создании программного обеспечения.	Тема 3.2 Проектирование ПО в Case-средстве RATIONAL ROSE	Пакет RATIONAL ROSE. Начальные сведения. Интерфейс. Начальные этапы по работе с RATIONAL ROSE. Разработка клиент – серверной системы.	<b>4</b>
9.	Раздел 4. Международные стандарты проектирования прикладного обеспечения.	Тема 4.1 Международные стандарты проектирования.	Существующие на данный момент международные стандарты. Обзор . ISO/IEC JTC1/SC7 - международный технический комитет стандартов под контролем ИСО (ISO - Международная организация по стандартизации) и МЭК (IEC - Международная электротехническая комиссия). Стандарт проектирования устанавливает (ISO/IEC 2382-20: 1990, Information technology ~ Vocabulary - Part 20 : Systems development. (Разработка систем)	<b>4</b>
10.	Раздел 4. Международные стандарты проектирования прикладного обеспечения.	Тема 4.2 Разработка, оформление документации, пользовательского интерфейса ПО	Набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации; - правила фиксации проектных решений на диаграммах, в том числе: правила именования объектов (включая соглашения по терминологии), набор атрибутов для всех объектов и правила их заполнения на каждой стадии, правила оформления диаграмм, включая требования к форме и размерам объектов, и т. д. ; - требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы, настройки CASE-средств, общие настройки проекта. Механизм обеспечения совместной работы над проектом, в том числе: правила интеграции подсистем проекта, правила поддержания проекта в одинаковом для всех разработчиков состоянии (регламент обмена проектной информацией, механизм фиксации общих объектов и	<b>2</b>

			т.д.), правила проверки проектных решений на непротиворечивость . Стандарт оформления проектной документации ИСО 6592:1985. Обработка информации. Руководящие указания по документированию автоматизированных прикладных систем.	
<b>Итого часов</b>				<b>36</b>

#### 4.2.3. Лабораторный практикум

№ п/п	Наименование раздела дисциплины	Наименование лабораторной работы	Содержание лабораторной работы	Всего часов
1	2	3	4	5
<b>Семестр 6</b>				
1.	Раздел 2. Жизненный цикл программного обеспечения	Лабораторная работа №1 Создание интерфейса программного приложения в среде визуального программирования	Создание многооконного интерфейса, с различными элементами перехода от одного окна к другому.	<b>2</b>
2.	Раздел 2. Жизненный цикл программного обеспечения	Лабораторная работа № 2 Создание базы данных для своего приложения.	Формирование базы данных (сбор исходных данных по своим индивидуальным заданиям) или формирование вычислительной части ПО	<b>2</b>
3.	Раздел 3. CASE-технологии при создании программного обеспечения	Лабораторная работа №3,4 Описание процессов с помощью UML диаграмм.	Создание учебных проектов с помощью UML диаграмм, в визуальной среде Visual Paradigm UML 2.0 (Отдел информации и информационной безопасности. Администрация Усть – Джегутинского муниципального района)	<b>8</b>
4.	Раздел 4. Международные стандарты проектирования	Лабораторная работа №5 Подключение поисковой системы к ПО	Завершающая стадия над индивидуальным программным обеспечением	<b>4</b>
<b>ИТОГО часов в 6 семестре</b>				<b>16</b>

#### 4.2.4. Практические занятия

№ п/п	Наименование раздела дисциплины	Наименование практического занятия	Содержание практического занятия	Всего часов
1	2	3	4	5
<b>Семестр 6</b>				

1.	Раздел 2. Жизненный цикл программного обеспечения	<i>Практическое занятие 2.1.</i> различные интерфейсы прикладного программного обеспечения.	Практические приемы создания различных интерфейсов в визуальном программировании (Отдел информации и информационной безопасности. Администрация Усть – Джугутинского муниципального района)	<b>2</b>
2.	Раздел 2. Жизненный цикл программного обеспечения	<i>Практическое занятие 2.2</i> Различные способы создания баз данных.	Изучение практических подходов связывания разнородной информации.	<b>2</b>
3.	Раздел 3. CASE-технологии при создании программного обеспечения	<i>Практическое занятие 3.1</i> Изучение практических возможностей визуальной среды Visual Paradigm UML	Интерфейс. Основные возможности. Варианты описание бизнес – процессов.	<b>10</b>
4.	Раздел 4. Международные стандарты проектирования	<i>Практическое занятие 4.1</i> Связывание всех частей программного обеспечения	Компиляция, тестирование созданного программного приложения по индивидуальным заданиям	<b>4</b>
<b>ИТОГО часов в 6 семестре:</b>				<b>18</b>

#### 4.3. САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ

№ п/п	Наименование раздела (темы) дисциплины	№ п/п	Виды СРО	Всего часов
1	2	3	4	5
<b>Семестр 6</b>				
1.	Раздел 1. Основные сведения о программном обеспечении	1.1	Работа с книжными источниками. Работа с электронными учебниками. Подготовка к тестированию. Подготовка к лабораторному практикуму	<b>9</b>
2.	Раздел 2. Жизненный цикл программного обеспечения	2.1	Работа с книжными источниками. Работа с электронными учебниками. Подготовка к тестированию. Подготовка к лабораторному практикуму	<b>9</b>
3.	Раздел 3. CASE-технологии при создании программного обеспечения		Работа с книжными источниками. Работа с электронными учебниками. Подготовка к тестированию. Подготовка к лабораторному практикуму	<b>10</b>
4.	Раздел 4. Международные стандарты проектирования		Работа с книжными источниками. Работа с электронными учебниками. Подготовка к тестированию. Подготовка к лабораторному практикуму	<b>10</b>
<b>ИТОГО часов в 6 семестре:</b>				<b>38</b>

## **5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

### **5.1. Методические указания для подготовки обучающихся к лекционным занятиям**

Изучение дисциплины «Проектирование программного обеспечения» требует систематического и последовательного накопления знаний по теории проектирования ПО, следовательно, пропуски отдельных тем не способствуют глубокому пониманию и освоению этого предмета. Именно поэтому необходим систематический контроль преподавателя над вниманием и работой обучающихся во время лекции.

Обучающийся, готовясь к лекционному занятию, включает выполнение всех видов заданий размещенных в каждой лекции, т.е. задания выполняются еще до лекционного занятия по соответствующей теме. В ходе лекционных занятий, обучающийся должен:

- вести конспектирование лекционного материала;
- обращать внимание на термины и определения, а также формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.
- обучающийся должен оставить в рабочих конспектах поля, на которых делает пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений;
- обучающийся должен задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, а так же разрешения спорных ситуаций;
- дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой. Обучающийся должен уметь проводить параллель текущей лекции с предшествующей.

### **5.2. Методические указания для подготовки обучающихся к лабораторным занятиям**

Ведущей дидактической целью лабораторных занятий является систематизация и обобщение знаний по изучаемой теме, приобретение практических навыков по тому или другому разделу курса, закрепление полученных теоретических знаний. Лабораторные работы сопровождают и поддерживают лекционный курс.

Обучающийся должен подготовиться к выполнению лабораторных работ строго в соответствии с содержанием курса.

В начале каждого лабораторного занятия обучающийся должен ознакомиться с теоретическим материалом, необходимым для выполнения текущей лабораторной работы.

Подготовить ответы на контрольные вопросы, которые представляются в конце каждой лабораторной работы.

Каждая лабораторная работа содержит список индивидуальных заданий, на выполнение которых обучающийся должен быть готовым.

Полученные результаты выполнения лабораторной работы обучающийся должен уметь оформить и быть готовым к устной защите.

### **5.3. Методические указания для подготовки обучающихся к практическим занятиям**

В процессе подготовки и проведения практических занятий обучающиеся закрепляют полученные ранее теоретические знания, приобретают навыки их практического применения, опыт рациональной организации учебной работы, готовятся к сдаче зачета, экзамена.

В начале семестра обучающиеся получают сводную информацию о формах проведения занятий и формах контроля знаний. Тогда же обучающимся предоставляется список тем лекционных и практических заданий, а также тематика рефератов. Каждое практическое занятие по соответствующей тематике теоретического курса состоит из вопросов для подготовки, на основе которых проводится устный опрос каждого обучающегося. Также после изучения каждого раздела обучающиеся для закрепления пройденного материала решают тесты, сдают коллоквиумы и делают реферативные работы по дополнительным материалам курса.

Поскольку активность обучающегося на практических занятиях является предметом внутрисеместрового контроля его продвижения в освоении курса, подготовка к таким занятиям требует от студента ответственного отношения.

При подготовке к занятию обучающиеся в первую очередь должны использовать материал лекций и соответствующих литературных источников. Самоконтроль качества подготовки к каждому занятию обучающиеся осуществляют, проверяя свои знания и отвечая на вопросы для самопроверки по соответствующей теме.

Входной контроль осуществляется преподавателем в виде проверки и актуализации знаний обучающихся по соответствующей теме.

Выходной контроль осуществляется преподавателем проверкой качества и полноты выполнения задания.

Типовой план практических занятий:

1. Изложение преподавателем темы занятия, его целей и задач.
2. Выдача преподавателем задания обучающимся, необходимые пояснения.
3. Выполнение задания обучающимися под наблюдением преподавателя.

Обсуждение результатов. Резюме преподавателя.

4. Общее подведение итогов занятия преподавателем и выдача домашнего задания.

Коллоквиум

Форма проверки и оценивания знаний обучающихся в системе образования, представляет собой проводимый по инициативе преподавателя промежуточный контроль знаний по определенным разделам для оценки текущего уровня знаний обучающихся, а также для повышения знаний обучающихся. После окончания семинарского занятия обучающимися выставляются оценки за работу.

Обучающийся при подготовке к практическому занятию может консультироваться с преподавателем и получать от него наводящие разъяснения.

Задания для самостоятельной работы и формы контроля за их выполнением

Формы самостоятельной работы обучающегося по освоению дисциплины

1. Усвоение текущего учебного материала;
2. Конспектирование первоисточников;
3. Работа с конспектами лекций;
4. Подготовка по темам для самостоятельного изучения;
5. Написание докладов и реферативных работ по заданным темам;
6. Изучение специальной, методической литературы;
7. Подготовка к экзамену.

Дидактические цели практического занятия: углубление, систематизация и закрепление знаний, превращение их в убеждения; проверка знаний; привитие умений и навыков самостоятельной работы с книгой; развитие культуры речи, формирование умения аргументировано отстаивать свою точку зрения, отвечать на вопросы слушателей; умение слушать других, задавать вопросы.

Задачи: стимулировать регулярное изучение программного материала, первоисточников; закреплять знания, полученные на уроке и во время самостоятельной работы; обогащать знаниями благодаря выступлениям товарищей и учителя на занятии, корректировать ранее полученные знания.

## 5.4 Подготовка презентации и реферата

Для подготовки презентации рекомендуется использовать: PowerPoint, MS Word, Acrobat Reader, LaTeX-овский пакет beamer. Самая простая программа для создания презентаций – Microsoft PowerPoint. Для подготовки презентации необходимо собрать и обработать начальную информацию.

Последовательность подготовки презентации:

1. Четко сформулировать цель презентации: вы хотите свою аудиторию мотивировать, убедить, заразить какой-то идеей или просто формально отчитаться.
2. Определить каков будет формат презентации: живое выступление (тогда, сколько будет его продолжительность) или электронная рассылка (каков будет контекст презентации).
3. Отобразить всю содержательную часть для презентации и выстроить логическую цепочку представления.
4. Определить ключевые моменты в содержании текста и выделить их.
5. Определить виды визуализации (картинки) для отображения их на слайдах в соответствии с логикой, целью и спецификой материала.
6. Подобрать дизайн и форматировать слайды (количество картинок и текста, их расположение, цвет и размер).
7. Проверить визуальное восприятие презентации.

К видам визуализации относятся иллюстрации, образы, диаграммы, таблицы. Иллюстрация - представление реально существующего зрительного ряда. Образы – в отличие от иллюстраций - метафора. Их назначение - вызвать эмоцию и создать отношение к ней, воздействовать на аудиторию. С помощью хорошо продуманных и представляемых образов, информация может надолго остаться в памяти человека. Диаграмма - визуализация количественных и качественных связей. Их используют для убедительной демонстрации данных, для пространственного мышления в дополнение к логическому. Таблица - конкретный, наглядный и точный показ данных. Ее основное назначение - структурировать информацию, что порой облегчает восприятие данных аудиторией.

Практические советы по подготовке презентации готовьте отдельно:

- печатный текст + слайды + раздаточный материал;
- слайды - визуальная подача информации, которая должна содержать минимум текста, максимум изображений, несущих смысловую нагрузку, выглядеть наглядно и просто;
- текстовое содержание презентации – устная речь или чтение, которая должна включать аргументы, факты, доказательства и эмоции;
- рекомендуемое число слайдов 17-22;
- обязательная информация для презентации: тема, фамилия и инициалы выступающего; план сообщения; краткие выводы из всего сказанного; список использованных источников;
- раздаточный материал – должен обеспечивать ту же глубину и охват, что и живое выступление: люди больше доверяют тому, что они могут унести с собой, чем исчезающим изображениям, слова и слайды забываются, а раздаточный материал остается постоянным осязаемым напоминанием; раздаточный материал важно раздавать в конце презентации; раздаточный материалы должны отличаться от слайдов, должны быть более информативными.

Тема доклада должна быть согласованна с преподавателем и соответствовать теме учебного занятия. Материалы при его подготовке, должны соответствовать научно-методическим требованиям вуза и быть указаны в докладе. Необходимо соблюдать регламент, оговоренный при получении задания. Иллюстрации должны быть

достаточными, но не чрезмерными.

Работа обучающегося над докладом-презентацией включает отработку умения самостоятельно обобщать материал и делать выводы в заключении, умения ориентироваться в материале и отвечать на дополнительные вопросы слушателей, отработку навыков ораторства, умения проводить диспут.

Докладчики должны знать и уметь: сообщать новую информацию; использовать технические средства; хорошо ориентироваться в теме всего семинарского занятия; дискутировать и быстро отвечать на заданные вопросы; четко выполнять установленный регламент (не более 10 минут); иметь представление о композиционной структуре доклада и др.

### **Структура выступления**

Вступление помогает обеспечить успех выступления по любой тематике. Вступление должно содержать: название, сообщение основной идеи, современную оценку предмета изложения, краткое перечисление рассматриваемых вопросов, живую интересную форму изложения, акцентирование внимания на важных моментах, оригинальность подхода.

Основная часть, в которой выступающий должен глубоко раскрыть суть затронутой темы, обычно строится по принципу отчета. Задача основной части – представить достаточно данных для того, чтобы слушатели заинтересовались темой и захотели ознакомиться с материалами. При этом логическая структура теоретического блока не должны даваться без наглядных пособий, аудио-визуальных и визуальных материалов.

Заключение – ясное, четкое обобщение и краткие выводы, которых всегда ждут слушатели.

## **5.5 Методические указания по самостоятельной работе обучающихся**

Самостоятельная работа обучающихся предполагает различные формы индивидуальной учебной деятельности: конспектирование научной литературы, сбор и анализ практического материала в СМИ, проектирование, выполнение тематических и творческих заданий и пр. Выбор форм и видов самостоятельной работы определяется индивидуально-личностным подходом к обучению совместно преподавателем и обучающимся. Формы текущего контроля успеваемости и промежуточной аттестации обучающихся.

Содержание внеаудиторной самостоятельной работы обучающихся по дисциплине «Проектирование программного обеспечения» включает в себя различные виды деятельности:

- чтение текста (учебника, первоисточника, дополнительной литературы);
- составление плана текста;
- конспектирование текста;
- работа со словарями и справочниками;
- ознакомление с нормативными документами;
- исследовательская работа;
- использование аудио- и видеозаписи;
- работа с электронными информационными ресурсами;
- выполнение тестовых заданий;
- ответы на контрольные вопросы;
- аннотирование, реферирование, рецензирование текста;
- составление глоссария или библиографии по конкретной теме;



- решение задач и упражнений.

### **Работа с литературными источниками и интернет ресурсами**

В процессе подготовки к практическим занятиям, обучающимся необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у обучающихся свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического занятия, что позволяет обучающимся проявить свою индивидуальность в рамках выступления на данных занятиях, выявить широкий спектр мнений по изучаемой проблеме.

## **6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

<b>№ п/п</b>	<b>№ семестра</b>	<b>Виды учебной работы</b>	<b>Образовательные технологии</b>	<b>Всего часов</b>
1	2	3	4	
1	6	<i>Лекция 1.1</i> Цели и задачи технологий разработки ПО.	Технологии развития критического мышления. Обзорная лекция.	<b>2</b>
2	6	<i>Лекция 1.2</i> Особенности современных крупных проектов ИС	Лекция–информация.	<b>2</b>
3	6	<i>Лекция 2.1</i> Программные средства. Программное обеспечение (ПО).	Лекция – презентация с использованием Power Point.	<b>4</b>
4	6	<i>Лекция 2.2</i> Жизненный цикл (ЖЦ) ПО.	Лекция – презентация с использованием Power Point.	<b>4</b>
5	6	<i>Лекция 2.3</i> Процессы ЖЦ ПО	Лекция – презентация с использованием Power Point.	<b>4</b>
7	6	<i>Лекция 2.4</i> Модели ЖЦ ПО.	Лекция – презентация с использованием Power Point.	<b>4</b>
8	6	<i>Лекция 3.1</i> Классификация CASE- средств	Лекция-информация.	<b>6</b>
11	6	<i>Лекция 3.2</i> Проектирование ПО в Case-средстве RATIONAL ROSE	Использование компьютерных технологий при выполнении индивидуальных практических заданий	<b>4</b>
12	6	<i>Лекция 4.1</i> Международные стандарты проектирования.	Лекция – презентация с использованием Power Point.	<b>4</b>
18	6	<i>Лекция 4.2</i> Разработка, оформление документации, пользовательского интерфейса ПО	Использование компьютерных технологий при выполнении индивидуальных практических заданий	<b>2</b>
19	6	<i>Лабораторная работа №1</i> Создание интерфейса программного приложения в среде визуального программирования	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>2</b>
20	6	<i>Лабораторная работа № 2</i>	Самостоятельная работа обучающихся по	<b>2</b>

		Создание базы данных для своего приложения.	индивидуальным заданиям лабораторного практикума с ПЭВМ.	
21	6	<i>Лабораторная работа №3,4</i> Описание процессов с помощью UML диаграмм	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>10</b>
22	6	<i>Лабораторная работа №5</i> Подключение поисковой системы к ПО	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>4</b>
23	6	<i>Практическое занятие 2.1 .</i> различные интерфейсы прикладного программного обеспечения.	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>2</b>
24	6	<i>Практическое занятие 2.2</i> Различные способы создания баз данных.	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>4</b>
25	6	<i>Практическое занятие 3.1</i> Изучение практических возможностей визуальной среды Visual Paradigm UML	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>10</b>
26	6	<i>Практическое занятие 4.1</i> Связывание всех частей программного обеспечения	Самостоятельная работа обучающихся по индивидуальным заданиям лабораторного практикума с ПЭВМ.	<b>4</b>

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 7.1. Перечень основной и дополнительной учебной литературы

<b>Список основной литературы</b>	
1.	Влацкая И.В. Проектирование и реализация прикладного программного обеспечения : учебное пособие / Влацкая И.В., Заельская Н.А., Надточий Н.С.. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2015. — 119 с. — ISBN 978-5-7410-1238-3. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/54145.html">https://www.iprbookshop.ru/54145.html</a> — Режим доступа: для авторизир. пользователей
2.	Смирнов А.А. Разработка прикладного программного обеспечения : учебное пособие / Смирнов А.А.. — Москва : Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2003. — 101 с. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/10808.html">https://www.iprbookshop.ru/10808.html</a> — Режим доступа: для авторизир. пользователей
3.	Смирнов А.А. Прикладное программное обеспечение : учебное пособие / Смирнов А.А.. — Москва : Евразийский открытый институт, 2011. — 384 с. — ISBN 978-5-374-00340-6. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/11079.html">https://www.iprbookshop.ru/11079.html</a> — Режим доступа: для авторизир. пользователей
4.	Иванова Н.Ю. Системное и прикладное программное обеспечение : учебное пособие / Иванова Н.Ю., Маняхина В.Г.. — Москва : Прометей, 2011. — 202 с. — ISBN 978-5-4263-0078-1. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/58201.html">https://www.iprbookshop.ru/58201.html</a> — Режим доступа: для авторизир. пользователей
<b>Список дополнительной литературы</b>	
5.	Гуныко А.В. Системное программное обеспечение : конспект лекций / Гуныко А.В.. — Новосибирск : Новосибирский государственный технический университет, 2011. — 138 с. — ISBN 978-5-7782-1670-9. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/45020.html">https://www.iprbookshop.ru/45020.html</a> — Режим доступа: для авторизир. пользователей

6.	Журавлёва И.А. Системное и прикладное программное обеспечение : лабораторный практикум / Журавлёва И.А., Корнеев П.К.. — Ставрополь : Северо-Кавказский федеральный университет, 2017. — 132 с. — Текст : электронный // IPR SMART : [сайт]. — URL: <a href="https://www.iprbookshop.ru/69432.html">https://www.iprbookshop.ru/69432.html</a> — Режим доступа: для авторизир. пользователей
----	---

## 7.2 Интернет-ресурсы, справочные системы

1. Электронно-библиотечная система IPRbooks URL: [http:// www.iprbooks.ru/](http://www.iprbooks.ru/) ООО «Ай Пи Эр Медиа»

## 7.3 Информационные технологии

Лицензионное программное обеспечение	Реквизиты лицензий/ договоров
Microsoft Azure Dev Tools for Teaching 1. Windows 7, 8, 8.1, 10 2. Visual Studio 2008, 2010, 2013, 2019 5. Visio 2007, 2010, 2013 6. Project 2008, 2010, 2013 7. Access 2007, 2010, 2013 и т. д.	Идентификатор подписчика: 1203743421 Срок действия: 30.06.2022  (продление подписки)
MS Office 2003, 2007, 2010, 2013	Сведения об Open Office: 63143487, 63321452, 64026734, 6416302, 64344172, 64394739, 64468661, 64489816, 64537893, 64563149, 64990070, 65615073 Лицензия бессрочная
Антивирус Dr.Web Desktop Security Suite	Лицензионный сертификат Серийный № 8DVG-V96F-H8S7-NRBC Срок действия: с 20.10.2022 до 22.10.2023
Консультант Плюс	Договор № 272-186/С-23-01 от 20.12.2022 г.
ЭБС IPR SMART	Лицензионный договор № 9368/22П от 01.07.2022 г. Срок действия: с 01.07.2022 до 01.07.2023

Бесплатное ПО: Lazarus, Firebird, IBE Expert, Pascal ABC, Python, VBA, Virtual box, Sumatra PDF, 7-Zip

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### 8.1. Требования к аудиториям (помещениям, местам) для проведения занятий

#### 1. Учебная аудитория для проведения занятий лекционного типа:

Специализированная мебель:

Стол преподавательский - 1 шт., стул мягкий - 1 шт., доска меловая - 1 шт., парты - 10 шт., компьютерные столы - 11 шт., стулья - 21 шт.,

Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:

Персональный компьютер – 1 шт.

Экран рулонный настенный – 1 шт.

Проектор – 1 шт.

#### 2. Учебная аудитория для проведения занятий семинарского типа, курсового проектирования (выполнение курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Специализированная мебель:

Стол преподавательский - 1 шт., стул мягкий - 1 шт., доска меловая - 1 шт., парты - 10 шт., компьютерные столы - 11 шт., стулья - 21 шт.,  
Лабораторное оборудование, технические средства обучения, служащие для предоставления учебной информации большой аудитории:  
Персональный компьютер – 1 шт.  
Экран рулонный настенный – 1 шт.  
Проектор – 1 шт.

### **3. Помещение для самостоятельной работы**

Отдел обслуживания печатными изданиями

Специализированная мебель: Рабочие столы на 1 место – 21 шт. Стулья – 55 шт. Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации: экран настенный – 1 шт.

Проектор – 1 шт. Ноутбук – 1 шт.

Информационно-библиографический отдел.

Специализированная мебель:

Рабочие столы на 1 место - 6 шт. Стулья - 6 шт.

Компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ФГБОУ ВО «СевКавГА»:

Персональный компьютер – 1 шт. Сканер – 1 шт. МФУ – 1 шт. Отдел обслуживания электронными изданиями Специализированная мебель:

Рабочие столы на 1 место – 24 шт. Стулья – 24 шт.

Набор демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации:

Интерактивная система - 1 шт. Монитор – 21 шт. Сетевой терминал - 18 шт. Персональный компьютер - 3 шт. МФУ – 2 шт. Принтер – 1 шт.

### **4. Помещение для хранения и профилактического обслуживания учебного оборудования**

Специализированная мебель: Шкаф – 1 шт., стул - 2 шт., кресло компьютерное – 2 шт., стол угловой компьютерный – 2 шт., тумбочки с ключом – 2 шт. Учебное пособие (персональный компьютер в комплекте) – 2 шт.

#### **8.2. Требования к оборудованию рабочих мест преподавателя и обучающихся**

Рабочее место преподавателя, оснащенное компьютером с доступом в Интернет.

Рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде, и т.п.

#### **8.3. Требования к специализированному оборудованию нет**

## **9. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Для обеспечения образования инвалидов и обучающихся с ограниченными возможностями здоровья разрабатывается (в случае необходимости) адаптированная образовательная программа, индивидуальный учебный план с учетом особенностей их психофизического развития и состояния здоровья, в частности применяется

индивидуальный подход к освоению дисциплины, индивидуальные задания: рефераты, письменные работы и, наоборот, только устные ответы и диалоги, индивидуальные консультации, использование диктофона и других записывающих средств для воспроизведения лекционного и семинарского материала.

В целях обеспечения обучающихся инвалидов и лиц с ограниченными возможностями здоровья комплектуется фонд основной учебной литературой, адаптированной к ограничению электронных образовательных ресурсов, доступ к которым организован в БИЦ Академии. В библиотеке проводятся индивидуальные консультации для данной категории пользователей, оказывается помощь в регистрации и использовании сетевых и локальных электронных образовательных ресурсов, предоставляются места в читальном зале.

**ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ**  
**«Проектирование программного обеспечения»**

# ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

## «Проектирование программного обеспечения»

### 1. Компетенции, формируемые в процессе изучения дисциплины

Индекс	Формулировка компетенции
ПК-3	Способен ориентироваться в современных алгоритмах компьютерной математики; обладать способностями к эффективному применению и реализации математически сложных алгоритмов в современных программных комплексах

### 2. Этапы формирования компетенции в процессе освоения дисциплины

Основными этапами формирования указанных компетенций при изучении обучающимися дисциплины являются последовательное изучение содержательно связанных между собой разделов (тем) учебных занятий. Изучение каждого раздела (темы) предполагает овладение обучающимися необходимыми компетенциями. Результат аттестации обучающихся на различных этапах формирования компетенций показывает уровень освоения компетенций обучающимися.

Этапность формирования компетенций прямо связана с местом дисциплины в образовательной программе.

Разделы (темы ) дисциплины	ПК -3
	+
Раздел 1. Основные сведения о программном обеспечении	+
Тема 1.1 Цели и задачи технологий разработки ПО.	+
Тема 1.2 Особенности современных крупных проектов ИС	+
Раздел 2. Жизненный цикл программного обеспечения	+
Тема 2.1 Программные средства. Программное обеспечение (ПО).	
Тема 2.2 Жизненный цикл (ЖЦ) ПО.	+
Тема 2.3 Процессы ЖЦ ПО	+
Тема 2.4 Модели ЖЦ ПО	
Раздел 3. CASE-технологии при создании программного обеспечения.	+
Тема 3.1 Классификация CASE- средств	+

Тема 3.2 Проектирование ПО в Case-средстве RATIONAL ROSE	+
Раздел 4. Международные стандарты проектирования прикладного обеспечения.	
Тема 4.1 Международные стандарты проектирования.	+
Тема 4.2 Разработка, оформление документации, пользовательского интерфейса ПО	+



### 3. Показатели, критерии и средства оценивания компетенций, формируемых в процессе изучения дисциплины

**ПК-3** Способность ориентироваться в современных алгоритмах компьютерной математики, обладать способностями к эффективному применению и реализации

математически сложных алгоритмов в современных программных комплексах

Индикаторы достижений	Критерии оценивания результатов обучения				Средства оценивания результатов обучения	
	неудовлетв	удовлетв	хорошо	отлично	Текущий контроль	Промежуточная аттестация
ПК-3.1 Осуществляет выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов	Не может осуществить выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов	Частично осуществляет выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов	Может осуществить выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов	Отлично осуществляет выбор современных алгоритмов компьютерной математики с целью дальнейшей реализации таких алгоритмов в виде прикладных программ или прикладных комплексов	Коллоквиум, контрольные вопросы, тестирование. Реферат. индивидуальные задания к лабораторным работам	экзамен.
ПК-3.2 Способен практически реализовать алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи	Не может реализовать алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи	Частично реализовывает алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи	Может реализовать с небольшими ошибками алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи	Может реализовать алгоритм компьютерной математики для дальнейшей автоматизации решения прикладной задачи	Коллоквиум, контрольные вопросы, тестирование. Реферат. индивидуальные задания к лабораторным работам	экзамен.
ПК-3.3 Способен осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Не может осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Частично осуществляет оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Может осуществлять оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Отлично может осуществить оптимизацию алгоритмов при создании прикладных программ или прикладных комплексов	Коллоквиум, контрольные вопросы, тестирование, индивидуальные задания к лабораторным работам.	экзамен.

## 4. Комплект контрольно-оценочных средств по дисциплине

### Вопросы к экзамену

по дисциплине «Проектирование программного обеспечения»

1. Исследование CASE – технологии в разработке программного обеспечения.
2. Преимущества CASE – технологий по сравнению с другими способами моделирования.
3. Унифицированный язык моделирования UML.
4. Основные принципы языка UML.
5. CASE – средства RATIONAL ROSE (2003).
6. Функциональные возможности RATIONAL ROSE (2003).
7. Функциональные возможности системы ABC Pascal.
8. Модули ABC Pascal.NET.
9. Что такое Microsoft.Net.
10. Объектно-ориентированное программирование. Основные понятия.
11. Реализация объектно-ориентированного подхода. Инкапсуляция. Свойства.
12. Реализация объектно-ориентированного подхода. Наследование. Свойства.
13. Реализация объектно-ориентированного подхода. Униформизм. Свойства.
14. Проектирование программного обеспечения. Основные направления. Классификация типов.
15. Жизненный цикл программного обеспечения. Процессы жизненного цикла.
16. Модели жизненного программного обеспечения.
17. Процессы командной разработки программного обеспечения MSF. Принципы MSF.
18. Процессы командной разработки программного обеспечения MSF. Модель команд.
19. Процессы командной разработки программного обеспечения MSF. Масштабирование команд.
20. Гибкие технологии разработки программного обеспечения. Принципы и значение гибкой разработки.
21. Гибкие технологии разработки программного обеспечения. Методологии разработки программного обеспечения.
22. Операции работы с файлами.
23. Объектно-ориентированная разработка программ. Объектно-ориентирование языка программирования.
24. Объектно ориентированные методологии. Разработки программных систем CASE-средства. Разработки программного обеспечения.
25. Рациональный унифицированный процесс. Динамические аспекты процесса.
26. Методические основы технологий создания программного обеспечения. Визуальное моделирование.
27. Методы структурного анализа и проектирование программного обеспечения.
28. Сопоставление и взаимосвязь структурного и объектно – ориентированного подходов .
29. Методы моделирования бизнес – процессов и спецификация требований
30. Методы анализа и проектирования программного обеспечения.

### Практические задания к экзамену

по дисциплине «Проектирование программного обеспечения»

**Задача №1.** В отделе работает восемь человек, вывести на экран следующие данные: имя, отчество, фамилию, стаж работы работников, чей возраст не превышает 30 лет. Исходные данные ввести с клавиатуры.

**Задача №2.** Сотрудник налоговой инспекции оштрафовал за день шесть человек. Вывести на экран фамилии, номера машин, сумму штрафа для водителей, оштрафованных более чем на 10т. и общую сумму штрафов. Исходные данные ввести с клавиатуры.

**Задача №3.** Расчетная ведомость содержит фамилии, табельные номера и сумму заработка за январь, и Февраль семи работников. Вывести все о работниках, чей январский заработок превышает февральский. Исходные данные ввести с клавиатуры.

**Задача №4.** Акт инвентаризации материальных ценностей содержит шесть наименований предметов (шкаф, стол и т.д.), их количество и стоимость. Вывести на экран данные акта и суммарную стоимость по наименованиям. Исходные данные ввести с клавиатуры

**Задача №5.** Реестр поступлений за коммунальные услуги за год жильцов восьмиквартального дома содержит лицевые счета и суммы поступлений за отопление, газ и Эл. Вывести на экран данные реестра и общую сумму оплаты по каждому счету. Исходные данные ввести с клавиатуры.

**Задача №6.** Расчет заработной платы содержит фамилии десяти работников, их табельные номера, сумму зарплаты, сумму премии, составляющей 20% от зарплаты и сумму к выдаче. Подсчитать сумму премии и сумму к выдаче, остальные данные ввести с клавиатуры. Вывести на экран все данные.

**Задача №7.** Журнал отгрузки готовой продукции содержит наименование шести видов продукции, количество, цену за единицу продукции, суммарную стоимость о наименованиям. Подсчитать суммарную стоимость, остальные данные ввести с клавиатуры. Вывести на экран все данные журнала.

**Задача №8.** Реестр актов на покупку продуктов содержит наименование десяти продуктов. Стоимость за 1 кг., количество продукта в кг и общую стоимость по наименованиям. Вычислить общую стоимость, остальные данные ввести с клавиатуры. Вывести на экран все данные реестра

**Задача №9.** Реестр актов на покупку продуктов содержит наименование десяти продуктов. Стоимость за 1кг, количество продукта в кг и общую стоимость по наименованиям. Вычислить общую стоимость, остальные данные ввести с клавиатуры. Вывести на экран все данные реестра.

**Задача №10.** Ведомость затрат по шести заказам содержит номера заказа, фамилию ведущего, сумму затрат по трем статьям расходов и общую сумму затрат по каждому заказу. Вычислить общую сумму затрат, остальные данные ввести с клавиатуры. Вывести на экран все данные ведомости.

**Задача №11.** Ведомость затрат по шести заказам содержит номер заказа. Фамилию ведущего, сумму затрат по трем статьям расходов и общую сумму затрат по каждому заказу. Вычислить общую сумму затрат, остальные данные ввести с клавиатуры. Вывести на экран все данные ведомости.

**Задача №12.** Ведомость выработки по пяти бригадам за три месяца содержит наименование месяца, сумму выработки по бригадам за месяц. Общую сумму выработки за месяц. Вычислить общую сумму выработки, остальные данные ввести с клавиатуры. Вывести на экран все данные ведомости.

# СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

Кафедра Математика

20\_ - 20\_ учебный год

Экзаменационный билет № 1

по дисциплине «Проектирование программного обеспечения»  
для обучающихся 3 курса направления подготовки 01.03.04 Прикладная математика

1. Жизненный цикл программного обеспечения. Процессы жизненного цикла.
2. Объектно- ориентированные методологии. Разработки программных систем CASE-средства. Разработки программного обеспечения.
3. Создать прикладное приложение, интерфейс которого будет состоять из трех окон, в первом окне информация об авторе, второе окно – расчет простых и сложных процентов, третье окно – краткая теоретическая справка.

Зав. кафедрой «Математика»

Кочкаров А.М.

## Контрольные вопросы

по дисциплине «Проектирование программного обеспечения»

### Вопросы к разделу 1.

1. Особенности современных крупных проектов ИС (ЖЦ, сложность, качество).
2. Разработка требований. Спецификации.
3. Проектирование архитектуры и структуры ПО (включая испытание, внедрение, орган. разработки).
4. Планирование проектирования ПО. САПР
5. Технологии программирования. Групповая разработка.

### Вопросы к разделу 2.

1. Архитектурное и детальное проектирование.
2. Реализация и кодирование.
3. Тестирование и верификация. Процесс контроля качества. Методы контроля качества.
4. Цели тестирования. Верификация, валидация и системное тестирование.
5. Характеристики качества и критерии качества ПО, (надежность; эффективность; практичность; универсальность; сопровождаемость; корректность; обеспечение завершенности ПС).

6. Жизненный цикл программы. Циклический характер разработки.
7. Процессы и модели. Какой международный стандарт определяет перечень и содержание процессов ЖЦ ПО?
8. Основные группы процессов жизненного цикла и процессы каждой из групп.
9. Стадии жизненного цикла ПС, особенности разработки ПС, основные международные стандарты, пять подходов к разработке.
10. Модели процесса разработки. Водопадный подход (каскадная модель), выводы о применимости классической каскадной модели.
11. Модели процесса разработки. Итерационный подход (спиральная и инкрементальная модели). Гибкие модели процесса разработки.

### **Вопросы к разделу 3.**

1. Особенности современных крупных проектов ИС (ЖЦ, сложность, качество).
2. Разработка требований. Спецификации.
3. Проектирование архитектуры и структуры ПО (включая испытание, внедрение, орган. разработки).
4. Планирование проектирования ПО. САПР
5. Технологии программирования. Групповая разработка.

### **Вопросы к разделу 4.**

1. Международные стандарты проектирования, разработки, оформления документации, пользовательского интерфейса ПИ.
2. Измерения, меры и метрики. Размерно-ориентированные метрики. Функционально-ориентированные метрики.
3. Выполнение оценки проекта на основе LOC- и FP-метрик.
4. Коллективный характер разработки. Состав и структура коллектива разработчиков, их функции. Конструирование модели команды.
5. Конструирование модели процесса. Выявление требований к процессу

## **Вопросы для коллоквиума**

по дисциплине «Проектирование программного обеспечения»

### **Вопросы к разделу 1.**

6. Особенности современных крупных проектов ИС (ЖЦ, сложность, качество).
7. Разработка требований. Спецификации.
8. Проектирование архитектуры и структуры ПО (включая испытание, внедрение, орган. разработки).
9. Планирование проектирования ПО. САПР
10. Технологии программирования. Групповая разработка.

### **Вопросы к разделу 2.**

12. Архитектурное и детальное проектирование.
13. Реализация и кодирование.
14. Тестирование и верификация. Процесс контроля качества. Методы контроля качества.
15. Цели тестирования. Верификация, валидация и системное тестирование.
16. Характеристики качества и критерии качества ПО, (надежность; эффективность; практичность; универсальность; сопровождаемость; корректность; обеспечение завершенности ПС).
17. Жизненный цикл программы. Циклический характер разработки.
18. Процессы и модели. Какой международный стандарт определяет перечень и содержание процессов ЖЦ ПО?
19. Основные группы процессов жизненного цикла и процессы каждой из групп.
20. Стадии жизненного цикла ПС, особенности разработки ПС, основные международные стандарты, пять подходов к разработке.
21. Модели процесса разработки. Водопадный подход (каскадная модель), выводы о применимости классической каскадной модели.
22. Модели процесса разработки. Итерационный подход (спиральная и инкрементальная модели). Гибкие модели процесса разработки.

### **Вопросы к разделу 3.**

6. Особенности современных крупных проектов ИС (ЖЦ, сложность, качество).
7. Разработка требований. Спецификации.
8. Проектирование архитектуры и структуры ПО (включая испытание, внедрение, орган. разработки).
9. Планирование проектирования ПО. САПР
10. Технологии программирования. Групповая разработка.

### **Вопросы к разделу 4.**

6. Международные стандарты проектирования, разработки, оформления документации, пользовательского интерфейса ПИ.
7. Измерения, меры и метрики. Размерно-ориентированные метрики. Функционально-ориентированные метрики.
8. Выполнение оценки проекта на основе LOC- и FP-метрик.
9. Коллективный характер разработки. Состав и структура коллектива разработчиков, их функции. Конструирование модели команды.
10. Конструирование модели процесса. Выявление требований к процессу

## Индивидуальные задания лабораторных работ

по дисциплине «Проектирование программного обеспечения»

### Лабораторная работа №3,4.

#### Unified Modeling Language как унифицированный язык моделирования

##### Задание

С помощью UML диаграмм описать бизнес-процессы «Химчистки», в визуальной среде Visual Paradigm UML 2.0.

##### Диаграмма классов

**Диаграмма классов, Class diagram** — статическая структурная диаграмма, описывающая структуру системы, она демонстрирует классы системы, их атрибуты, методы и зависимости между классами.

Существуют разные точки зрения на построение диаграмм классов в зависимости от целей их применения:

- концептуальная точка зрения — диаграмма классов описывает модель предметной области, в ней присутствуют только классы прикладных объектов;
- точка зрения спецификации — диаграмма классов применяется при проектировании информационных систем;
- точка зрения реализации — диаграмма классов содержит классы, используемые непосредственно в программном коде (при использовании объектно-ориентированных языков программирования).

Каждый класс имеет свое название и свои атрибуты и операции, которые впоследствии будут реализованы в программном коде. Если операция имеет какой-либо тип, например, *+Занесать информацию в БД(Num: int; Zakaz: string; Zakazchik: string; Status: boolean): boolean*, то в таком случае в программном коде данная операция будет представлена в виде функции указанного типа с параметрами, которые находятся в круглых скобках. Класс «Номер заказа» имеет стереотип «entity» (сущность).

На рисунке видны связи между классами. Данные связи были выявлены после просмотра и анализа диаграмм последовательности, а также на данной диаграмме добавлены множественности. На основе диаграммы классов, будет построен фрагмент кода программы, будут проанализированы и выявлены алгоритмы наследия классов. Эти классы непосредственно будут фигурировать в программном коде.

Для более наглядного и понятного взаимодействия классов между собой, нужно объединить классы в пакеты. Таким образом, мы сгруппируем все наши классы в модули. И при кодогенерации каждый пакет выведется в отдельный модуль, согласно той группировке, которую мы произвели. Диаграммы пакетов служат, в первую очередь, для организации элементов в группы по какому-либо признаку с целью упрощения структуры и организации работы с моделью нашей системы.

На данном этапе мы разделил классы на три пакета: «Сущность», «Управление», «Реализация». Каждый пакет непосредственно взаимодействует с другим, так пакет «Сущность» порождает связь с пакетом «Управление», а пакет «Управление в свою очередь» порождает связь с пакетом «Реализация». Классы, находящиеся в пакетах, могут так же между собой взаимодействовать, это наглядно показано на рисунке. Деление на пакеты было произведено по функциональному принципу. В пакет «Сущность» мы поместили 2 класса: «Заявка», «Номер заявки» — так как именно эти классы порождают наш бизнес-процесс и отражают его сущность. В пакет «Управление» мы поместили 3 класса: «Управляющий заказами», «Финансовый директор», «Бухгалтерия», так как эти классы осуществляют управление бизнес-процессом. И, наконец, пакет «Реализация», в него вошли 3 класса: «БД», «Механик», «Поставщик», так как именно в этих классах происходит реализация бизнес-

процесса.

### **Deployment Diagram (Диаграмма развертывания)**

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения (runtime). При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме развертывания они не указываются.

Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развертывания является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы и ее разработка, как правило, является последним этапом спецификации модели.

Узел (node) представляет собой некоторый физически существующий элемент системы, обладающий некоторым вычислительным ресурсом. В качестве вычислительного ресурса узла может рассматриваться наличие по меньшей мере некоторого объема электронной или магнитооптической памяти и/или процессора.

Кроме собственно изображений узлов на диаграмме развертывания указываются отношения между ними. В качестве отношений выступают физические соединения между узлами и зависимости между узлами и компонентами, изображения которых тоже могут присутствовать на диаграммах развертывания.

Соединения являются разновидностью ассоциации и изображаются отрезками линий без стрелок. Наличие такой линии указывает на необходимость организации физического канала для обмена информацией между соответствующими узлами.

На нашей диаграмме развертывания указана зависимость компонента реализации сервера приложения от рабочих станций.

unified language диаграмма кодогенерация

### **Кодогенерация на Delphi**

```
Unit СТО_Бизнес;
```

```
Interface
```

```
Type
```

```
Заявка = Class;
```

```
Class = Class;
```

```
Бухгалтерия = Class;
```

```
БД = Class;
```

```
Заявка = Class(TObject)
```

```
Public Procedure Составление_текста_заявки();
```

```
End;
```

```
Class = Class(TObject)
```

```
Public Procedure Произвести_осмотр_согласно_заявке();
```

```
End;
```

```
Бухгалтерия = Class(TObject)
```

```
Public Procedure Произведит_расчет();
```

```
Public Procedure Получение_квитанции_об_оказании_услуг();
```

```
End;
```

```
БД = Class(TObject)
```

```
Public Procedure Ввод_данных_в_БД();
```

```
Public Procedure Сохранение_данных();
```

```
Public Procedure Создать_новую_запись_в_БД();
```

```
End;
```



```

Implementation
Procedure Заявка.Составление_текста_заявки();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure Class.Произвести_осмотр_согласно_заявке();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure Бухгалтерия.Произвестит_расчет();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure Бухгалтерия.Получение_квитанции_об_оказании_услуг();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure БД.Ввод_данных_в_БД();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure БД.Сохранение_данных();
Begin
Raise Exception.Create('Not yet implemented');
End;
Procedure БД.Создать_новую_запись_в_БД();
Begin
Raise Exception.Create('Not yet implemented');
End;
End.

```

### **Индивидуальные задания :**

Проектирование и разработка программного обеспечения с использованием принципов и технологий объектно-ориентированного анализа и программирования. При реализации проекта для хранения информации нужно использовать файловое хранилище собственного или открытого (XML) формата.

#### **1.Перенос слов**

Как показывают многочисленные эксперименты, разбиение русского слова на части для переноса с одной строки на другую с большой вероятностью выполняются правильно, если пользоваться следующими простыми приемами:

- 1) Две идущие подряд гласные можно разделить, если первой из них предшествует согласная, а за второй идет хотя бы одна буква (буква й при этом рассматривается вместе с предшествующей гласной как единое целое).
- 2) Две идущие подряд согласные можно разделить, если первой из них предшествует гласная, а в той части слова, которая идет за второй согласной, имеется хотя бы одна гласная (буквы ь, ъ вместе с предшествующей согласной рассматриваются как единое целое).

3) Если не удастся применить пункты 1), 2), то следует попытаться разбить слово так, чтобы первая часть содержала более чем одну букву и оканчивалась на гласную, а вторая содержала хотя бы одну гласную. Вероятность правильного разбиения увеличивается, если предварительно воспользоваться хотя бы неполным списком приставок, содержащих гласные, и попытаться прежде всего выделить из слова такую приставку. Дан текст на русском языке. Выполнить форматирование его строк по длине с помощью на переноса слов.

## 2. Морской бой

На поле 10 на 10 позиций стоят невидимые вражеские корабли: 4 корабля по одной клетке, три корабля по 2 клетки, 2 корабля по 3 клетки, 1 корабль в 4 клетки. Позиции указываются русскими буквами от А до К (по строкам) и цифрами от 1 до 10 (по столбцам). Конфигурация и положение кораблей на поле выбираются с помощью датчика случайных чисел. Если клетка корабля угадана играющим верно, она отмечается крестиком; в противном случае точкой.

Написать программу для игры против компьютера в односторонний морской бой

## 3. Одномерные клеточные автоматы

Рассмотрим клетки, расположенные вдоль прямой. Каждая из клеток может иметь состояние 0 или 1. На каждом шаге по времени новое состояние клетки вычисляется из старого состояния клетки и состояний её соседей. На экране каждый горизонтальный ряд показывает состояния клеток с помощью соответствующего цвета в очередной момент времени. Временная ось идет сверху вниз и каждый ряд вычисляется на основе предыдущего ряда. Различные классы линейных клеточных автоматов могут быть определены в зависимости от того, как много различных значений (состояний) клетка может иметь ( $k$ ), и как много соседей на каждой стороне клетки используются при вычислении нового состояния  $\mathbb{R}$ . Правила для определения следующего состояния клетки используют сумму состояний самой клетки и клеток-соседей. Сумма состояний отображается правилом на новое состояние клетки. Если  $k=2$ ,  $r=1$ , то только ближайшие соседи клетки используются. Если каждая из клеток имеет состояние 1, то максимальная сумма  $1+1+1$  равна 3 и сумма может меняться от 0 до 3, т. е. имеем четыре величины. Поэтому правило преобразования можно задать с помощью строки из четырех двузначных цифр. Например, правило 1010, начиная с правой цифры, задает отображение суммы  $0 \rightarrow 0$ ,  $1 \rightarrow 1$ ,  $2 \rightarrow 0$ ,  $3 \rightarrow 1$  (таким образом, если сумма равна 2, то новое состояние равно 0). Для  $k=2$  и  $r=2$  каждая клетка (на экране она изображается пикселем) новое значение получает в зависимости от значений двух соседних клеток с каждой стороны. Соответствующие правила должны иметь 6 двузначных цифр. Если  $k=3$  и  $r=1$ , то каждая клетка может иметь значение 0, 1 или 2. Учитывается только единственный сосед с каждой стороны, поэтому результат преобразования задается правилом из 7 цифр и т. д.

Напишите программу для построения линейных клеточных автоматов следующих классов  $kr = 21, 31, 41, 22, 32, 42$ .

$k$

$r$

количество

пример правила

цифр в правиле

2

1

4

1010

3

1

7

1211001

4

1

10

3311100320

2

2

6

0110110

3

2

11

21212002010

4

2

16

2300331230331001

Начальная строка задается либо случайным образом, либо пользователем.

Как ведут себя клеточные линейные автоматы? Происходит ли переход к однородному состоянию независимо от начальных данных? Существуют ли классы автоматов с локализованными стационарными или периодическими конфигурациями? Есть ли автоматы с хаотическим временным поведением? Есть ли автоматы, которые ведут себя по разному при различных начальных данных? Необходимо ответить на эти вопросы. Смотрите программу-аналог Fractint.

#### **4. Часы с кукушкой**

Составить программу, моделирующую работу стрелочных часов с кукушкой. У часов должен быть циферблат с тремя стрелками, маятник и окошко. Каждый час в окошке должна появляться кукушка. Она должна появиться столько раз, сколько часов показывают часы

#### **5. Карточная игра**

Составить программу, которая раздает игральные карты заданному количеству игроков (одним из игроков является человек, за остальных играет компьютер) и моделирует игру в «дурака». Компьютерная программа играет случайным образом, без анализа уже вышедших карт.

#### **6. Крестики-нолики**

Составить программу, позволяющую играть на бесконечном поле в «крестики-нолики»:

а) игроку с компьютером;

б) двум игрокам.

Если в качестве игрока выступает компьютер, программа делает первый ход. Делая очередной ход, программа анализирует ситуацию, рассчитывая возможные ходы противника вперед на 1 — 2 хода, и в результате проведенного анализа поступает оптимальным образом.

## 7. Быки и коровы

Составить программу, позволяющую играть в «Быки и коровы»:

- а) игроку с компьютером;
- б) двум игрокам.

Суть игры в следующем. Каждый из противников задумывает четырехзначное число, все цифры которого различны (первая цифра числа отлична от нуля). Необходимо разгадать задуманное число. Выигрывает тот, кто отгадает первый.

Противники по очереди называют друг другу числа и сообщают о количестве «быков» и «коров» в названном числе («бык» — цифра есть в записи задуманного числа и стоит в той же позиции, что и в задуманном числе; «корова» — цифра есть в записи задуманного числа, но не стоит в той же позиции, что и в задуманном числе).

Например, если задумано число 3275 и названо число 1234, получаем в названном числе одного «быка» и одну «корову». Очевидно, что число отгадано в том случае, если имеем 4 «быка».

## 8. Две лисы и 20 кур

На поле, указанной на рисунке формы находятся две лисы и 20 кур. Куры могут перемещаться на один шаг вверх, влево или вправо, но не назад и не по диагонали. Лисы также могут перемещаться только на один шаг (вверх, вниз, влево и вправо).

Лиса может съесть курицу, как в игре в шашки: если в горизонтальном или вертикальном направлении за курицей на один шаг следует свободное поле, то лиса перепрыгивает через курицу и берет ее.

Лисы всегда обязаны есть, и, когда у них есть выбор, они обязаны осуществлять «наиболее длинное поедание». Если два приема пищи имеют одинаковую длину, осуществляется один из них — по выбору лисы.

Составить программу, которая играет за лис (лисы перемещаются вверх, вниз и в стороны, но не по диагонали). Игрок перемещает кур (куры могут двигаться вверх и в стороны, но не назад). Партнеры играют по очереди, причем куры начинают. Они

л л к

выигрывают партию, если девяти из них удастся занять 9 полей, образующих верхний квадрат поля.

Начальное положение кур и лис изображено на рисунке.

Лисы выигрывают, если им удастся съесть 12 кур, так как тогда оставшихся кур недостаточно, чтобы занять 9 верхних полей.

## 9. Игра в слова

Составить программу, позволяющую компьютеру и человеку играть в слова. Предварительно программа объясняет правила игры и позволяет уточнить их в любой

момент. Тематикой игры могут быть по выбору города, животные, растения и т. д. Тема выбирается из предложенных компьютером (не менее 3).

## 10. Морской бой

Составить программу для игры в морской бой игрока с компьютером. Программа должна позволять расставлять корабли на поле  $10 \times 10$ , контролировать правильность их расстановки, давать противнику возможность поочередно делать ходы и выдавать соответствующие информационные сообщения. Программа должна анализировать предыдущие ходы и следующий делать на основе проведенного анализа.

## 11. Графики

Составить программу, которая предлагает пользователю некоторый список функций для построения графиков (например,  $y = ax^2 + b$ ,  $y = ax + b$  и т.д. — до 10 наименований). После выбора соответствующей функции, задания коэффициентов и отрезка, на котором выполняется построение, программа строит указанный график. Затем значение коэффициентов и положение графика можно менять (например, с помощью клавиш управления курсором), после чего график перестраивается и записывается обновленное уравнение соответствующей кривой.

## 12. Задача об инфекции стригущего лишая

Промоделировать процесс распространения инфекции — стригущего лишая по участку кожи размером  $N \times N$  клеток ( $N$  — нечетное). Предполагается, что исходной зараженной клеткой кожи является центральная. В каждый интервал времени пораженная инфекцией клетка может с вероятностью 0,5 заразить любую из соседних здоровых клеток. По прошествии шести единиц времени зараженная клетка становится невосприимчивой к инфекции. Возникший иммунитет действует в течение последующих четырех единиц времени, а затем клетка оказывается здоровой. В ходе моделирования описанного процесса выдавать текущее состояние моделируемого участка кожи в каждом интервале времени отмечая зараженные, невосприимчивые к инфекции и здоровые клетки.

## 13. Создать информационно-поисковую систему

Она должна обеспечивать возможность ввода, редактирования, удаления, поиска, фильтрации данных, а так же вывода отчетов по заданным критериям (не менее 3х отчетов)

14. ИПС "Библиотека"
15. ИПС "Проф. колледж"
16. ИПС "Расписание занятий"
17. ИПС "Успеваемость"
18. ИПС "Склад оптовой базы"
19. ИПС "Поликлиника"
20. ИПС "Продажа видео и аудио продукции"
21. ИПС "ПО и состав ПК организации"
22. ИПС "Каталог музыкальных треков"
23. ИПС "Ветеринарная клиника"

## Комплект тестовых вопросов

по дисциплине «Проектирование программного обеспечения»

**№1 К какому типу проектов относятся проекты по разработке ПО:**

1. и к творческим, и к промышленным проектам
2. к промышленным проектам
3. к творческим проектам

**№2 Какие возвраты невозможны при разработке по водопадной модели:**

1. возврат от кодированию к тестированию
2. возврат от тестирования к анализу
3. возврат от тестирования к кодированию

**№3 Какие возвраты невозможны при разработке по водопадной модели:**

1. возврат от кодированию к тестированию
2. возврат от тестирования к кодированию
3. возврат от кодирования к разработке системных требований

**№4 Этап, занимающий наибольшее время, при разработке программы:**

---

**№5 Первый этап в жизненном цикле программы:**

1. анализ требований
2. формулирование требований
3. проектирование
4. автономное тестирование
5. комплексное тестирование.

**№6 Один из необязательных этапов жизненного цикла программы:**

1. тестирование
2. проектирование
3. тестирование
4. оптимизация
5. программирование
6. анализ требований

**№7 Самый большой этап в жизненном цикле программы:**

1. эксплуатация
2. изучение предметной области
3. программирование
4. тестирование
5. корректировка ошибок

**№8 Какой этап выполняется раньше:**

1. отладка

2. тестирование
3. компиляция

**№9 Способы оценки надежности:**

1. тестирование
2. сравнение с аналогами
3. трассировка
4. оптимизация

**№10 Необходимо определить максимальное количество узлов в двоичном дереве с высотой  $k$ , где корень — нулевая высота:**

---

**№11 Укажите обозначение следующей фразы: “алгоритм  $X$  асимптотически более эффективен, чем  $Y$ ”:**

1.  $X$  будет лучшим выбором для всех входов
2.  $X$  будет лучшим выбором для всех входов, кроме больших входов
3.  $X$  будет лучшим выбором для всех входов, за исключением, возможно, небольших входов

**№12 Что выполняет следующее выражение на  $C$   $x = x \& (x-1)$ :**

---

**№13 Team Foundation Server:**

1. включает в себя систему управления версиями
2. система, поддерживающая сбор данных для построения отчетов
3. это продукт корпорации Intel

**№14 Диаграмма состояний применяется для описания поведения таких компонентов системы как:**

---

**№ 15 Последовательность этапов программирования:**

1. компилирование, компоновка, отладка
2. компоновка, отладка, компилирование
3. отладка, компилирование, компоновка
4. компилирование, отладка, компоновка.

**№ 16 Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:**

---

**№ 17 Методы программирования (укажите НЕ верный ответ):**

1. логическое
2. структурное
3. модульное.

**№18 Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:**

1. прямым указанием
2. венгерской нотацией
3. структурным программированием
4. поляризацией

**№ 19 Наличие комментариев позволяет:**

---

**№ 20 Что определяет выбор языка программирования:**

1. область приложения
2. знание языка
3. наличие дополнительных библиотек

**№ 21 Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:**

1. для экономических задач
2. для системных задач
3. для инженерных задач

**№ 22 Набор рекомендаций по выполнению разных процессов жизненного цикла программ, оформленный в виде базы знаний**

1. Rational Process
2. Rational Unified Process
3. Rational Unified

**№ 23 Система Paradigm Plus, которая используется в качестве поддержки программного обеспечения ЕСМ представляет собой**

1. набор рекомендаций по разбиению жизненного цикла программ на отдельные этапы, рекомендации по организации этих этапов, объединенные с CASE-системой построения моделей для всех этапов
2. набор рекомендаций по разбиению программного кода на отдельные этапы
3. набор рекомендаций по разбиению модуля программного кода на отдельные этапы, рекомендации по организации этих этапов, объединенные с CASE-системой построения моделей для всех этапов

**№ 24 Вид защиты программного средства включающего в себя защиту от так называемых «компьютерных вирусов»**

---

**№ 25 Типы по целям библиотеки классов**

1. на библиотеки специального назначения и библиотеки, узконаправленного по областям применения
2. на библиотеки общего назначения и библиотеки, специализированные по областям применения
3. на библиотеки общего и специального назначения и библиотеки,

**№ 26 Укажите наиболее часто применяемый способ приспособления классов и объектов к конкретной задаче**



1. уточнение с помощью наследования, т.е. базовые классы, имеющиеся в библиотеке, реализуют основные алгоритмы обработки, а в программе из них выводятся конкретные классы, изменяющие и приспособливающие их к текущей задаче
2. уточнение с помощью иерархии, классов, имеющихся в библиотеке, реализуют основные алгоритмы обработки, а в программе из них выводятся конкретные классы, изменяющие и приспособливающие их к текущей задаче
3. уточнение с помощью компиляции, классов, имеющиеся в библиотеке, реализуют основные алгоритмы обработки, а в программе из них выводятся конкретные классы, изменяющие и приспособливающие их к текущей задаче

**№ 27 Укажите способ приспособления функции к конкретной программе**

1. задание различных элементов
2. задание параметров результатов
3. задание различных аргументов

**№ 28 Назначение Rational Unified Process**

1. для организации всего жизненного цикла программирования, начиная от анализа деятельности организации (бизнес-моделирования) и кончая тестированием и установкой системы
2. для организации отдельных этапов жизненного цикла программирования, начиная от анализа деятельности организации (бизнес-моделирования) и кончая тестированием и установкой системы
3. для организации модулей программирования, начиная от анализа деятельности организации и кончая тестированием и установкой системы

**№ 29 CASE-средство, выпускаемое компанией Rational, предназначено для автоматизации тестирования**

---

**№ 30 Методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы**

1. объектно-ориентированное проектирование
2. структурное программирование
3. кодовое программирование

**№31 CASE-средство, выпускаемое компанией Rational, предназначенное для построения моделей и их графического изображения с помощью языка UML**

1. Rational Rose®
2. Rational Rose
3. Rational Lit

**№ 32 Ошибки компоновки заключаются в том, что:**

1. указано внешнее имя, но не объявлено
2. неправильно использовано зарезервированное слово
3. составлено неверное выражение
4. указан неверный тип переменной

**№33 Защитное программирование это:**

---

## Темы рефератов

по дисциплине «Проектирование программного обеспечения»

1. Системный подход к проектированию программного обеспечения
2. Методы проектирования программного обеспечения (программных продуктов)
3. Принципы проектирования информационного обеспечения программного комплекса.
4. Показатели качества программного обеспечения
5. Методы создания надежного программного обеспечения
6. Ассемблеры. Определение. Области применения.
7. Управление ресурсами вычислительных систем
8. Классификация и организация прерываний в ОС
9. Общая характеристика (типы) операционных систем
10. Модели баз данных
11. Уровни отображения данных в СУБД
12. Языки общения с СУБД (SQL, QBE)
13. Отличие баз данных от файловых систем
14. Методика проектирования имитационных моделей
15. Обработка списковых структур на языке Турбо Пролог.
16. Организация рекурсивных структур на Прологе.
17. Системы ИИ. Фреймы, семантические сети и процедуры.
18. Экспертные системы. Цель, состав и принципы функционирования.
19. Техническое и программное обеспечение глобальных компьютерных сетей (INTERNET).
20. Типы, стандарты протоколов компьютерной сети.
21. Топология компьютерных сетей.
22. Технические средства передачи информации.
23. Многоуровневая модель взаимодействия компьютерных сетей

## **5. Методические материалы, определяющие процедуры оценивания компетенции**

### **5.1 Критерии оценивания качества выполнения лабораторного практикума**

Оценка «**зачтено**» выставляется обучающемуся, если лабораторная работа выполнена правильно и обучающийся ответил на все вопросы, поставленные преподавателем на защите. Оценка «**не зачтено**» выставляется обучающемуся, если лабораторная работа выполнена не правильно или обучающийся не проявил глубоких теоретических знаний при защите работы

### **5.2 Критерии оценивания качества устного ответа**

Оценка «**отлично**» выставляется за глубокое знание предусмотренного программой материала, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы.

Оценка «**хорошо**» – за твердое знание основного (программного) материала, за грамотные, без существенных неточностей ответы на поставленные вопросы.

Оценка «**удовлетворительно**» – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности

изложения материала.

Оценка **«неудовлетворительно»** – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в материале, за незнание основных понятий дисциплины.

### **5.3 Критерии оценивания тестирования**

При тестировании все верные ответы берутся за 100%.

90%-100% отлично

75%-90% хорошо

60%-75% удовлетворительно

менее 60% неудовлетворительно

### **5.5 Критерии оценивания результатов освоения дисциплины**

Оценка **«отлично»** выставляется за глубокое знание предусмотренного программой материала, содержащегося в основных и дополнительных рекомендованных литературных источниках, за умение четко, лаконично и логически последовательно отвечать на поставленные вопросы, за умение анализировать изучаемые явления в их взаимосвязи и диалектическом развитии, применять теоретические положения при решении практических задач.

Оценка **«хорошо»** – за твердое знание основного (программного) материала, включая расчеты (при необходимости), за грамотные, без существенных неточностей ответы на поставленные вопросы, за умение применять теоретические положения для решения практических задач.

Оценка **«удовлетворительно»** – за общее знание только основного материала, за ответы, содержащие неточности или слабо аргументированные, с нарушением последовательности изложения материала, за слабое применение теоретических положений при решении практических задач.

Оценка **«неудовлетворительно»** – за незнание значительной части программного материала, за существенные ошибки в ответах на вопросы, за неумение ориентироваться в расчетах, за незнание основных понятий дисциплин.