

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования «Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ  
Директор Таганрогского института  
имени А.П. Чехова (филиала)  
РГЭУ (РИНХ)  
\_\_\_\_\_ Голобородько А.Ю.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

**Рабочая программа дисциплины  
Современные методы построения программ**

направление 09.03.03 Прикладная информатика  
направленность (профиль) 09.03.03.01 Прикладная информатика в менеджменте

Для набора \_\_\_\_\_ года

Квалификация  
Бакалавр

**КАФЕДРА информатики****Распределение часов дисциплины по семестрам**

Семестр (<Курс>. <Семестр на курсе>)	7 (4.1)		Итого	
	17 4/6			
Неделя	17 4/6			
Вид занятий	УП	РП	УП	РП
Лекции	16	16	16	16
Лабораторные	34	34	34	34
Итого ауд.	50	50	50	50
Контактная работа	50	50	50	50
Сам. работа	58	58	58	58
Часы на контроль	36	36	36	36
Итого	144	144	144	144

**ОСНОВАНИЕ**

Учебный план утвержден учёным советом вуза от 29.08.2023 протокол № 1.

Программу составил(и): Препод., Горкунов М.С. \_\_\_\_\_

Зав. кафедрой: Тюшнякова И.А. \_\_\_\_\_

### 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	формирование у студентов теоретических знаний и практических навыков, необходимых при создании программных систем: анализ требований, детального проектирования архитектуры приложения, обеспечения качества
-----	--

### 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

<b>ПКО-1.1:</b> Умеет проводить формализацию и алгоритмизацию поставленных задач
<b>ПКО-1.2:</b> Пишет программный код с использованием языков программирования, определения и манипулирования данными в соответствии с установленными требованиями
<b>ПКО-1.3:</b> Проверяет и выполняет отладку программного кода
<b>ОПК-7.1:</b> Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.
<b>ОПК-7.2:</b> Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.
<b>ОПК-7.3:</b> Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.
<b>ОПК-1.1:</b> Знает основы математики, физики, вычислительной техники и программирования.
<b>ОПК-1.2:</b> Умеет решать стандартные профессиональные задачи с применением естественнонаучных и обще- инженерных знаний, методов математического анализа и моделирования.
<b>ОПК-1.3:</b> Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности.

### В результате освоения дисциплины обучающийся должен:

<b>Знать:</b>
основные принципы построения современного программного обеспечения, типичные формы применения шаблонов проектирования
<b>Уметь:</b>
строить модель программного обеспечения на основе UML-диаграмм, применять основные паттерны проектирования, создавать эффективные сетевые и многопоточные приложения; разрабатывать, внедрять и адаптировать прикладное программное обеспечение
<b>Владеть:</b>
навыками применения современных методов разработки программного обеспечения; современных методов оценки качества программного обеспечения

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература
	<b>Раздел 1. Современные методы построения программ</b>				
1.1	Проблемы разработки сложных программ. Жизненный цикл программного обеспечения (ПО) и процессов разработки ПО. Каскадные и итеративные модели жизненного цикла, и набор стандартов, регулирующих процессы разработки ПО. Мобильность и переносимость ПО. Преимущества и недостатки переносимости программ. /Лек/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.2	Постановка задачи создания ПС. Разработка технического задания /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3

1.3	Анализ технического задания. Уточнение требуемого поведения (функциональности) разрабатываемой ПС. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.4	Стандарты, регулирующие процессы разработки ПО. Методология быстрой разработки приложений (RAD), методологии унифицированного процесса разработки Rational (RUP) и экстремального программирования (XP). Влияние архитектуры ПО на его свойства. Структурный подход в проектировании ПО и классификация структурных методологий. Диаграммы «сущность-связь» (ERD), диаграммы потоков данных (DFD), SADT-модели (стандарт IDEF0). Унифицированный язык моделирования UML. Основные понятия унифицированного языка моделирования (UML). Виды диаграмм UML. /Лек/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.5	Каскадные и итеративные технологии. Критичность и масштабность программных проектов. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.6	Построение объектно-ориентированной архитектуры системы. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.7	Связь тестирования и качества разрабатываемого ПО. Классификация типов тестов. Документирование и анализ ошибок. Оценка покрытия тестами ПО и требований к нему. Тестирование производительности и стабильности ПО. Оптимизация ПО с учетом производительности. Нагрузочное тестирование и стресс тестирование. /Лек/	7	2	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.8	Сбор метрик покрытия исходного кода тестами. Анализ кода на «мертвый код», маскирование требований. /Лаб/	7	6	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3

1.9	Современные тенденции в архитектуре компьютеров – многоядерные процессоры – и архитектура ПО. Проблематика современных вычислений и предпосылки к параллельным вычислениям. Математические основы параллельных вычислений. Высокопроизводительные и параллельные вычисления в научных исследованиях. Характер увеличения скорости программы в результате распараллеливания. Реализация параллельных вычислений. Интерфейсы прикладного программирования для разработки параллельных программ. Проблемы распределения ресурсов и исключения тупиков при разработке сложных программ. /Лек/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.10	Виртуализация как новый подход к построению IT- инфраструктуры. Две фундаментально различающиеся категории виртуализации: виртуализация платформ и виртуализация ресурсов. Виртуализация уровня операционной системы. Виртуализация уровня приложений. Применение виртуализации при разработке и тестировании приложений. /Лек/	7	2	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.11	Программная реализация и тестирование ПС. Комплексная проверка. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.12	Программная реализация и тестирование ПС. Построение тестовых драйверов с использованием тестовых фреймворков. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.13	Планирование процесса внедрения программного продукта. Процесс устранения ошибок на этапе внедрения. Техническая поддержка пользователей на этапе сопровождения. /Лаб/	7	4	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.14	Подготовка к лабораторным занятиям. /Ср/	7	10	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.15	Язык UML. Диаграммы UML, их назначение и правила составления. Применение UML для выполнения этапов анализа и проектирования ПС. /Ср/	7	48	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3

1.16	/Экзамен/	7	36	ОПК-1.1 ОПК-1.2 ОПК-1.3 ОПК-7.1 ОПК-7.2 ОПК-7.3 ПКО-1.1 ПКО-1.2 ПКО-1.3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
------	-----------	---	----	---	----------------------------

#### 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущей и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

#### 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

##### 5.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л1.1	Мейер Б.	Объектно-ориентированное программирование и программная инженерия	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=429034">http://biblioclub.ru/index.php?page=book&amp;id=429034</a> неограниченный доступ для зарегистрированных пользователей
Л1.2	Суханов, М. Б.	Программная инженерия: учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2018	<a href="http://www.iprbookshop.ru/102465.html">http://www.iprbookshop.ru/102465.html</a> неограниченный доступ для зарегистрированных пользователей

##### 5.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л2.1		Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика: журнал	Астрахань: Астраханский Государственный Технический Университет, 2020	<a href="http://biblioclub.ru/index.php?page=book&amp;id=596084">http://biblioclub.ru/index.php?page=book&amp;id=596084</a> неограниченный доступ для зарегистрированных пользователей
Л2.2		Прикладная информатика: журнал	Москва: Университет Синергия, 2020	<a href="http://biblioclub.ru/index.php?page=book&amp;id=600623">http://biblioclub.ru/index.php?page=book&amp;id=600623</a> неограниченный доступ для зарегистрированных пользователей
Л2.3	Прохоренок Н.А., Дронов В.А.	Python 3 и PyQt 5. Разработка приложений	Санкт-Петербург: БХВ-Петербург, 2016	<a href="https://ibooks.ru/reading.php?short=1&amp;productid=353570">https://ibooks.ru/reading.php?short=1&amp;productid=353570</a> неограниченный доступ для зарегистрированных пользователей

##### 5.3 Профессиональные базы данных и информационные справочные системы

Научная электронная библиотека <https://www.elibrary.ru/defaultx.asp>

Александр Леоненков. Самоучитель UML <http://www.e-reading.club/book.php?book=33640>

Введение в UML <https://intuit.ru/studies/courses/1007/229/info>

##### 5.4. Перечень программного обеспечения

Python
Microsoft Office
MS Visual Studio 2008 Express Edition

**5.5. Учебно-методические материалы для студентов с ограниченными возможностями здоровья**

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

**6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Помещения для лекционных занятий и лабораторных работ укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения. Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными программными средствами и выходом в Интернет.

**7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

1.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ПКО-1 -Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение			
З: основные принципы построения современного программного обеспечения	Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы; Использование современных информационных технологий	соответствие проблеме исследования; полнота и содержательность ответа; умение приводить примеры; умение отстаивать свою позицию; умение пользоваться дополнительной литературой при подготовке к занятиям; соответствие представленной информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет;	<i>ВЭ, ЛЗ, ИДЗ,Д</i>
У: строить модель программного обеспечения на основе UML-диаграмм			
В: навыками применения современных методов разработки программного обеспечения; современных методов оценки качества программного обеспечения			
ОПК-7 -Способен разрабатывать алгоритмы и программы, пригодные для практического применения			
З: основные принципы построения современного программного обеспечения	Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы; Использование современных информационных технологий	соответствие проблеме исследования; полнота и содержательность ответа; умение приводить примеры; умение отстаивать свою позицию; умение пользоваться дополнительной литературой при подготовке к занятиям; соответствие представленной информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет;	<i>ВЭ, ЛЗ, ИДЗ,Д</i>
У: разрабатывать, внедрять и адаптировать прикладное программное обеспечение			
В: навыками применения современных методов разработки программного обеспечения; современных методов оценки качества программного обеспечения			
ОПК-1 -Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности			



З: типичные формы применения шаблонов проектирования	Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы; Использование современных информационных технологий	соответствие проблеме исследования; полнота и содержательность ответа; умение приводить примеры; умение отстаивать свою позицию; умение пользоваться дополнительной литературой при подготовке к занятиям; соответствие представленной информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет;	<i>ВЭ, ЛЗ, ИДЗ,Д</i>
У: применять основные паттерны проектирования, создавать эффективные сетевые и многопоточные приложения			
В: навыками применения современных методов разработки программного обеспечения; современных методов оценки качества программного обеспечения			

### 1.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

84–100	5 (отлично)
67–83	4 (хорошо)
50–66	3 (удовлетворительно)
0–49	2 (неудовлетворительно)

## **2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

### **Вопросы к экзамену**

по дисциплине *Современные методы построения программ*

1. Проблемы разработки сложных программ.
2. Жизненный цикл программного обеспечения (ПО) и процессов разработки ПО.
3. Каскадные и итеративные модели жизненного цикла, и набор стандартов, регулирующих процессы разработки ПО.
4. Мобильность и переносимость ПО.
5. Преимущества и недостатки переносимости программ.
6. Стандарты, регулирующие процессы разработки ПО.
7. Методология быстрой разработки приложений (RAD), методологии унифицированного процесса разработки Rational (RUP) и экстремального программирования (XP).
8. Влияние архитектуры ПО на его свойства.

10. Структурный подход в проектировании ПО и классификация структурных методологий.
11. Диаграммы «сущность-связь» (ERD), диаграммы потоков данных (DFD), SADT-модели (стандарт IDEF0).
12. Унифицированный язык моделирования UML. Основные понятия унифицированного языка моделирования (UML).
13. Виды диаграмм UML.
14. Связь тестирования и качества разрабатываемого ПО.
15. Классификация типов тестов.
16. Документирование и анализ ошибок. Оценка покрытия тестами ПО и требований к нему.
17. Тестирование производительности и стабильности ПО. Оптимизация ПО с учетом производительности.
18. Нагрузочное тестирование и стресс тестирование.
19. Современные тенденции в архитектуре компьютеров – многопоточные приложения и архитектура ПО.
20. Проблематика современных вычислений и предпосылки к параллельным вычислениям.
21. Математические основы параллельных вычислений.
22. Характер увеличения скорости программы в результате распараллеливания.
23. Реализация параллельных вычислений.
24. Проблемы распределения ресурсов при асинхронном доступе к ресурсам при разработке сложных программ.
25. Виртуализация как новый подход к построению IT-инфраструктуры.
26. Две фундаментально различающиеся категории виртуализации: виртуализация платформ и виртуализация ресурсов.
27. Виртуализация уровня операционной системы.
28. Виртуализация уровня приложений. Применение виртуализации при разработке и тестировании приложений.

Образец билета:

Билет №1.

1. Рассказать о жизненном цикле дефекта.
2. Описать достоинства и недостатки «спирального» жизненного цикла ПО.
3. Выявить потенциальные дефекты у следующего требования:  
«Для обеспечения работы возложенных на него функций приложение должно обладать интуитивно понятным интерфейсом.»

### Критерии оценки:

Оценка	Критерии
Отлично (84–100)	ответы на вопросы четкие, обоснованные и полные, проявлена готовность к дискуссии, студент демонстрирует высокий уровень владения знаниями, умениями и навыками соответствующих компетенций, что позволяет ему решать широкий круг типовых и нетиповых задач.
Хорошо (67–83)	ответы на вопросы преимущественно правильные, но недостаточно четкие, студент способен самостоятельно воспроизводить и применять соответствующие знания, умения и навыки для решения типовых задач дисциплины, может выполнять поиск и использование новой информации для выполнения новых профессиональных действий на основе полностью освоенных знаний, умений и навыков соответствующих компетенций

Удовлетворительно (50-66)	ответы на вопросы не полные, на некоторые ответ не получен, знания, умения, навыки сформированы на базовом уровне, студенты частично, с помощью извне (например, с использованием наводящих вопросов, ассоциативного ряда понятий и т.д.) могут воспроизводить и применять соответствующие знания, умения, навыки
Неудовлетворительно (0-49)	на большую часть вопросов ответы не были получены, либо они показали полную некомпетентность студента в материале дисциплины, студент не способен самостоятельно, без помощи извне, воспроизводить и применять соответствующие знания, умения, навыки или знания, умения и навыки у студента не выявлены

## Лабораторные задания

по дисциплине Современные методы построения программ

### 1. Тематика лабораторных работ по разделам и темам

Постановка задачи создания ПС. Разработка технического задания
Анализ технического задания. Уточнение требуемого поведения (функциональности) разрабатываемой ПС.
Каскадные и итеративные технологии. Критичность и масштабность программных проектов.
Построение объектно-ориентированной архитектуры системы.
Сбор метрик покрытия исходного кода тестами. Анализ кода на «мертвый код», маскирование требований.
Программная реализация и тестирование ПС. Комплексная проверка.
Программная реализация и тестирование ПС. Построение тестовых драйверов с использованием тестовых фреймворков.
Планирование процесса внедрения программного продукта. Процесс устранения ошибок на этапе внедрения. Техническая поддержка пользователей на этапе сопровождения.

### Критерии оценки:

За выполнение всех лабораторных работ курса запланирован максимум в 40 баллов, если студент в ходе защиты показал наличие твердых знаний по материалу лабораторной работы, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике. В случае частичного выполнения работ, баллы уменьшаются пропорционально количеству защищенных работ.

## Демонстрационный вариант индивидуального домашнего задания

по дисциплине Современные методы построения программ

1. Для выбранного варианта задания разработать функциональные требования и диаграмму классов по стандарту UML.
2. Выполнить генерацию кода по созданной диаграмме классов и перенести его в среду программирования.
3. Разработать проект, в среде программирования, в виде библиотеки классов. Созданная библиотека классов, должна реализовывать запроюктированные, варианты использования.
4. Создать клиентское приложение для тестирования основных возможностей, созданной библиотеки классов.
5. 5. Предусмотреть обработку исключительных ситуаций

### **Критерии оценки:**

За выполнение ИДЗ запланирован максимум в 40 баллов. В случае частичного решения работы, баллы уменьшаются пропорционально количеству верно выполненных заданий.

## **Темы докладов**

по дисциплине Современные методы построения программ

1. Основные принципы структурного и модульного программирования.
2. Иерархическое упорядочивание.
3. Принципы тестирования.
4. Подходы к тестированию реализации.
5. Тестирование интеграции.
6. Объектно-ориентированное программирование (ООП). Краткий обзор современных ООП языков программирования.
7. Инструментальные среды разработки.
8. Тестирование объектно-ориентированных систем. Объектно-ориентированный анализ.
9. Язык UML.
10. Agile и XP процесс разработки программного обеспечения.
11. Особенности разработки web приложений.
12. Особенности разработки мобильных систем.
13. Анализ алгоритмов. Скорости роста, классификация скоростей роста.
14. Рекурсивные алгоритмы; анализ сложности. Построение дерева рекурсивных вызовов.
15. Линейные структуры данных: стек, очередь, дек.
16. Задачи поиска в структурах данных: линейный поиск.
17. Задачи поиска в структурах данных: алгоритмы поиска на строках.
18. Задачи поиска в структурах данных: хэширование.
19. Задачи поиска в структурах данных: trie-дерево.
20. Определение и использование шаблона класса. Параметры шаблонов.
21. Использование классов функциональных объектов для настройки шаблонных классов. Обработка исключительных ситуаций.
22. Основные концепции STL: контейнеры, итераторы, алгоритмы.

### **Критерии оценки:**

- 10-20 баллов - выставляется студенту, если: тема соответствует содержанию доклада; основные понятия проблемы изложены верно; сделаны обобщения и сопоставления различных точек зрения по рассматриваемому вопросу; сделаны и аргументированы основные выводы, доклад сопровождается разработанной мультимедийной презентацией;

- 0-9 баллов - выставляется студенту, если: содержание не соответствует теме; нет ссылок на использованные источники; тема не полностью раскрыта; нет выводов.

### **3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

**Промежуточная аттестация** проводится в форме экзамена, это аттестация в период сессии, которая проводится в соответствии с действующим в РГЭУ (РИНХ) Положением о курсовых, экзаменах и зачётах.

Экзамен проводится по расписанию экзаменационной сессии в компьютерном классе. Количество вопросов в экзаменационном задании – 3. Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины адресованы студентам всех форм обучения.

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные работы.

Важным условием успешного освоения дисциплины «Современные методы построения программ» является создание системы правильной организации труда, позволяющей распределить учебную нагрузку равномерно в соответствии с графиком образовательного процесса. Большую помощь в этом может оказать составление плана работы на семестр, месяц, неделю, день. Его наличие позволит подчинить свободное время целям учебы, трудиться более успешно и эффективно. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подвести итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине они произошли. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана. Все задания к лабораторным занятиям, а также задания, вынесенные на самостоятельную работу, рекомендуется выполнять непосредственно после соответствующей темы лекционного курса, что способствует лучшему усвоению материала, позволяет своевременно выявить и устранить «пробелы» в знаниях, систематизировать ранее пройденный материал, на его основе приступить к овладению новыми знаниями и навыками.

Знакомство с дисциплиной происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая студенту понять глубинные процессы развития изучаемого предмета как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном

материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

В процессе подготовки к лабораторным занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Изучение дисциплины проходит с акцентом на лабораторные работы. Лабораторные занятия проводятся в компьютерных классах с применением специально разработанных учебно-методических материалов, в которых изложены подробные методические рекомендации по изучению каждой темы и выполнению заданий. Наличие таких учебно-методических и дидактических материалов позволяет каждому студенту работать в своем индивидуальном темпе, а также дополнительно прорабатывать изучаемый материал во время самостоятельных занятий.

Перед выполнением лабораторной работы требуется получить вариант задания.

Далее необходимо ознакомиться с заданием. Выполнение лабораторной работы следует начать с изучения теоретических сведений, которые приводятся в соответствующих методических указаниях. Результаты работы необходимо оформить в виде отчета.

Лабораторная работа считается выполненной, если

- предоставлен отчет о результатах выполнения задания;
- проведена защита проделанной работы.

Защита проводится в несколько этапов:

- 1) Демонстрируются результаты выполнения задания.
- 2) В случае лабораторной работы, предусматривающей разработку программного приложения при помощи тестового примера доказывається, что результат, получаемый при выполнении программы правильный.
- 3) Далее требуется ответить на ряд вопросов из перечня контрольных вопросов, который приводится в задании на лабораторную работу.

Вариант задания выдается преподавателем, проводящим лабораторные занятия.

Для успешного овладения предлагаемым курсом студент должен обладать определённой информационной культурой: навыками работы с литературой, умением определять и находить информационные ресурсы, соответствующие целям и задачам образовательного процесса, получать к ним доступ и использовать в целях повышения эффективности своей профессиональной деятельности. При изучении данного курса необходимо максимально использовать компьютер, изучать дополнительные информационные ресурсы.

### **Подготовка к промежуточной аттестации.**

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).