

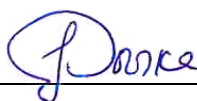
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«КАМЧАТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КамчатГТУ»)

Факультет информационных технологий

Кафедра «Информационные системы»

УТВЕРЖДАЮ

Декан факультета информационных  
технологий

 И.А. Рычка

«17» марта 2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Конструирование программного обеспечения»**

направление подготовки  
09.03.04 Программная инженерия  
(уровень бакалавриата)


направленность (профиль):  
«Разработка программно-информационных систем»

Петропавловск-Камчатский,  
2020

Рабочая программа дисциплины составлена на основании ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия, профиль «Разработка программно-информационных систем», учебного плана ФГБОУ ВО «КамчатГТУ».

Составитель рабочей программы

Доцент кафедры  
«Информационные системы»

  
\_\_\_\_\_  
(подпись)

С.В. Чебанюк  
(Ф.И.О.)

Доцент кафедры  
«Информационные системы»

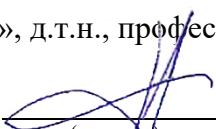
  
\_\_\_\_\_  
(подпись)

И.А. Рычка  
(Ф.И.О.)

Рабочая программа рассмотрена на заседании кафедры «Информационные системы». «05» марта 2021 г., протокол № 7.

Заведующий кафедрой «Информационные системы», д.т.н., профессор

«05» марта 2021 г.

  
\_\_\_\_\_  
(подпись)

И.Г. Проценко  
(Ф.И.О.)

## 1. Цели и задачи учебной дисциплины

Целью освоения дисциплины «Конструирование программного обеспечения» является формирование у студентов знаний по методам, инструментам и процессам разработки надежного, эффективного и безопасного программного обеспечения.

Задачами изучения дисциплины «Конструирование программного обеспечения» являются:

- систематизация, обобщение и углубление уже имеющиеся у студентов знаний в области конструирования программного обеспечения;
- формирование базового представления, первичных знаний, умений и навыков студентов в области конструирования программного обеспечения как этапа ЖЦ ПО;
- дать представление об эффективных способах и методах конструирования программного обеспечения.

## 2. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения (ПК-1).

Таблица – Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе бакалавриата индикаторами достижения компетенций

Код компетенции	Наименование компетенции	Код и наименование индикатора достижения	Планируемый результат обучения по дисциплине	Код показателя освоения
ПК-1	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	ИД-2 <sub>пкс-1</sub> умеет использовать методы моделирования, анализа и формальные методы конструирования программного обеспечения	Знать: – современные направления, стандарты, инструменты конструирования, методики организации и управления конструированием, сопутствующие работы: верификация, проектирование, тестирование, отладка в контексте конструирования.	3(ПК-1)1
			Уметь: – осуществлять организацию процесса конструирования и управление им, осуществлять создание кода в соответствии с требованиями качества и рекомендациями по конструированию.	У(ПК-1)1
			Владеть: – навыками планирования и управления конструированием, создания качественного кода, использования инструментов, поддерживающих работы по конструированию.	В(ПК-1)1

## 3. Место дисциплины в структуре образовательной программы

В структуре образовательной программы «Конструирование программного обеспечения» является дисциплиной части, формируемая участниками образовательных отношений.

Курс дисциплины «Конструирование программного обеспечения» продолжает цикл дисциплин разработки программных средств, опирается на знания в области информационных технологий и программирования, полученные на дисциплинах «Информатика и программирование», «Операционные системы», «Объектно-ориентированное программирование», «Машинно-зависимые языки программирования» и во время прохождения технологической (проектно-технологической) практики.

Знания, полученные обучающимися в процессе изучения дисциплины «Разработка программных приложений», позволят им выполнять на высоком уровне лабораторные и практические работы, курсовое и дипломное проектирование. Дисциплина «Разработка программных приложений» является базовой дисциплиной для дисциплины «Интернет-программирование», прохождения производственной (преддипломной) практики и выполнения дипломного проектирования.

#### 4. Содержание дисциплины

##### 4.1 Тематический план дисциплины

Наименование разделов и тем	Всего часов	Аудиторные занятия	Контактная работа по видам учебных занятий			Самостоятельная работа	Формы текущего контроля <sup>1</sup>	Итоговый контроль знаний по дисциплине
			Лекции	практические занятия	Лабораторные работы			
1	2	3	4	5	6	7	8	9
Очная форма обучения								
<b>Раздел 1. Конструирование программного обеспечения</b>	22	10	2	2	6	12	Опрос	
Основные понятия и определения технологии конструирования ПО	6	2			2	4	Опрос, РЗ	
Процессы стадии конструирования ЖЦ ПО	11	6	1	1	4	5	Опрос, РЗ	
Технологии программирования	12	7	2	2	3	5	Опрос, РЗ	
Инструментальные средства	10	5	1	1	3	5	Опрос, РЗ	
<b>Раздел 2. Объектно-ориентированные методологии разработки ПО</b>	44	14	4	4	6	30	Опрос	
Системное проектирование программного продукта	13	8	2	2	4	5	Опрос, РЗ	
Объектно-ориентированные модели архитектуры	15	10	3	3	4	5	Опрос, РЗ	
Унифицированный процесс разработки программных средств	11	6	2	2	2	5	Опрос, РЗ	
<b>КП</b>	<b>36</b>					<b>36</b>	Доклад	
<b>Экзамен</b>	<b>36</b>							<b>36</b>
<b>Всего</b>	<b>216</b>	<b>68</b>	<b>17</b>	<b>17</b>	<b>34</b>	<b>112</b>		<b>36</b>
Заочная форма обучения								
<b>Раздел 1. Конструирование программного обеспечения</b>	47	7	2	2	3	40	Опрос	
Основные понятия и определения технологии конструирования ПО	24	2			2	20	Опрос, РЗ	
Процессы стадии конструирования ЖЦ ПО	16	2			2	14	Опрос, РЗ	
Технологии программирования	16	2			2	14	Опрос, РЗ	
Инструментальные средства	16	2			2	14	Опрос, РЗ	
<b>Раздел 2. Объектно-ориентированные</b>	37	7	2	2	3	30	Опрос	

<sup>1</sup>Т – тестирование, КС – кейс-стади, РЗ – решение задач, ПЗ – практические задания, Д - доклады

методологии разработки ПО							
Системное проектирование программного продукта	16	2			2	14	Опрос, РЗ
Объектно-ориентированные модели архитектуры	16	2			2	14	Опрос, РЗ
Унифицированный процесс разработки программных средств	16	2			2	14	Опрос, РЗ
<b>КРП</b>	<b>9</b>					<b>9</b>	Доклад
<b>Экзамен</b>	<b>9</b>						<b>9</b>
<b>Всего</b>	<b>216</b>	<b>24</b>	<b>4</b>	<b>4</b>	<b>16</b>	<b>183</b>	<b>9</b>

#### 4.2 Содержание дисциплины

Жизненный цикл программного обеспечения. Инженерия ПО, проблемы разработки ПО. Технология разработки ПО и качество ПО. Стандарты в конструировании. Системный подход к разработке ПО. Общие концепции конструирования. Методологии программирования.

Проектирование снизу-вверх и проектирование сверху-вниз. Программные заглушки и их использование. Основные понятия структурного и объектно-ориентированного подхода к конструированию ПО.

Конструирование ПО, устойчивого к ошибкам. Способы обработки ошибок. Исключения. Изоляция повреждений, вызванных ошибками.

Повторное использование кода, командная разработка и поддержка программных продуктов. Рефакторинг программного обеспечения. Способы выявления проблемных участков кода.

Единицы программного кода. Основы объектно-ориентированного представления программных систем. Язык UML, структурные и поведенческие диаграммы. Паттерны проектирования, задачи, решаемые при их помощи, история создания.

Назначение и разработка предметно-ориентированных языков программирования. Тенденции развития технологий разработки ПО.

### Раздел 1. Конструирование программного обеспечения

#### Тема 1.1 Основные понятия и определения технологии конструирования ПО

Проблема сложности больших информационных систем. Программная инженерия. Конструирование программного обеспечения. Программное обеспечение как продукт. Программа как форма описания процесса. Понятие о программном средстве. Неконструктивность понятия правильной программы. Надежность программного средства. Технология программирования как технология разработки надежных программных средств.

#### Тема 1.2 Процессы стадии конструирования ЖЦ ПО

Эволюция программного средства. Жизненный цикл и процессы этапа разработки программного обеспечения. Оценка качества процессов создания программного обеспечения. Стандарты в конструировании.

Основные фазы разработки программного обеспечения: формулирование требований, формулирование целей проекта, анализ прикладных программ, создание функциональной спецификации. Специфика разработки программных средств. Жизненный цикл программного средства.

Понятие качества программного средства. Обеспечение надежности - основной мотив разработки программного средства. Методы борьбы со сложностью. Документация на программное обеспечение. Назначение и составление документации на программные изделия. Документы разработки сопровождения. Эксплуатационная документация на программные изделия.

Кодирование. Планирование разработки ПО. Технологические разработки программного обеспечения: инструментальная среда разработки, средства поддержки

проекта, отладчики. Методы тестирования. Сопровождение. Планирование разработки ПО.

### **Тема 1.3** Технологии программирования

Процедурное, структурное, объектно-ориентированное программирование. Объектно-ориентированные языки программирования.

Структурирование системы. Декомпозиция системы. Модульность программного средства. Связность модуля. Типы связности. Сцепление модуля. Типы сцепления

Реализация методом сверху-вниз и снизу-вверх. Комбинированные подходы. Системы управления исходными текстами программного управления конфигурациями ПО. Интеграция системы. Среда функционирования ПО. Доступные вычислительные ресурсы, платформа разработки. Уровень доступности ресурсов. Типы обнаружения проектных ошибок. Применение формальных методов построения программ.

### **Тема 1.4** Инструментальные средства

Технологические средства разработки программного обеспечения.

Компоненты интегрированных CASE-средств. Средства проектирования баз данных. Средства разработки приложений: 4 GL и генератора

Состав CASE-средств разработки ПО. История развития и Базовые принципы построения CASE-средств. Классификация сред разработки ПО. Сроки реализации отдельных подсистем. Управление конфигурацией проекта. Ведение версий проекта. Автоматизация выпуска проектной документации и синхронизация версий с версиями проекта. Независимость выполняемых проектных решений от средств реализации. Спиральная модель ЖЦ. Методология быстрой разработки приложений RAD (Rapid Application Development). Разработка приложений. Использование прототипирования для выяснения и удовлетворение потребностей конечного пользователя. Моделирование потоков. Информационные потоки. Процессы. Построение иерархии диаграмм потоков данных систем, диаграммы «сущность-связь» (ERD). Верификация системы. Понятие об архитектуре программной системы. Системы структур, которых отображает структуру входных данных. Системы построения на основе иерархии подзадач. Системы, управляемые событиями.

Унифицированный язык моделирования UML. Варианты использования. Диаграммы классов. Диаграммы взаимодействия. Диаграммы состояний. Модели реализации объектно-ориентированных программных систем.

### **Практическое занятие 1.1**

Класс интеллектуального ПО

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### **Практическое занятие 1.2**

Оконные приложения

*Рассматриваемые вопросы:*

- Системная функция. Системные величины.
- Элементы управления, события.

### **Практическое занятие 1.3**

Состав CASE-средств разработки

*Рассматриваемые вопросы:*

- Поддержка мультиязычности.
- Редактор. Настройка среды. Отладчик.
- Поддержка платформ.

### **Практическое занятие 1.4**

Нотация UML и диаграммы

*Рассматриваемые вопросы:*

- Диаграммы взаимодействия. Диаграммы состояний.
- Редактор UML-диаграмм, генерация кода.

### **Лабораторная 1.1** Изучение RAD-среды.

Цель: настройка RAD-среды.

*Рассматриваемые вопросы:*

1. Настройка параметров среды разработки.
2. Технологии конструирования консольных и оконных приложений.
3. Состав программного проекта.

*Задания:*

- Настроить среду для создания консольного приложения, составить техническое задание на разработку приложения, реализовать, описать продукт.
- Настроить среду для создания оконного приложения, составить техническое задание на разработку приложения, реализовать, описать продукт.

### **Лабораторная 1.2** Технология создания многооконного приложения

Цель: оценка качественных показателей ПП.

*Рассматриваемые вопросы:*

1. Архитектура многооконного приложения.
2. Проектирование поведения ПС.
3. Объектно-ориентированное проектирование ПС. Прототипы

*Задания:*

- Выполнить проектирование архитектуры многооконного приложения.
- Составить спецификации модулей программного средства при объектном подходе.
- Выполнить проектирование экранных форм.

### **Лабораторная 1.3** Событийно-ориентированное программирование

Цель: изучение RAD-среды при конструировании.

*Рассматриваемые вопросы:*

1. Описание поведения.
2. Системные события и операции.

*Задания:*

- Изучить стандартные системные события и определить поведение элементов управления.
- Получить техно-рабочий проект многооконного приложения.
- Описать возможности RAD-среды на стадии кодирования.
- Учет версий ПС.

**Самостоятельная работа по разделу.** Работа с конспектом лекций и рекомендованной литературой. Защита лабораторных работ.

## **Раздел 2. Объектно-ориентированные методологии разработки ПО**

### **Тема 2.1** Системное проектирование программного продукта

Основы проектирования. Ключевые вопросы проектирования. Процесс объектно-ориентированного проектирования, преимущества ОО методологии.

Архитектура программного обеспечения, задачи, цели, классификация. Архитектурное представление (логическое представление, представление процесса, представление развертывания и представление реализации). Документ архитектуры программного обеспечения, разработчик архитектуры.

Архитектурные стили проектирования. Графическое представление архитектуры

### **Тема 2.2** Объектно-ориентированные модели архитектуры

Объектно-ориентированные модели системной архитектуры. Модели подсистем,

Модели последовательности, Модели конечного автомата.

Модели реализации объектно-ориентированных программных систем. Компонентные диаграммы. Диаграммы размещения.

Объявление классов. Функции. Модификаторы. Передача примитивных типов в функции. Локальные и глобальные переменные. Модификаторы доступа и правила видимости. Ссылка this. Передача ссылочных типов в функции. Проблема изменения ссылки внутри подпрограммы.

**Тема 2.3** Методологии разработки программных средств

Agile-методы. Унифицированный процесс разработки программных средств Rational Unified Process (RUP). Экстремальное программирование. Scrum и Канбан (Kanban).

### ***Практическое занятие 2.1***

Архитектура ПО консольных приложений

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### ***Практическое занятие 2.2***

Архитектура ПО windows-приложений

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### ***Практическое занятие 2.3***

Архитектура ПО web-приложений

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### ***Практическое занятие 2.4***

Архитектура ПО мобильных-приложений

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### ***Практическое занятие 2.5***

Средства коллективной разработки

*Рассматриваемые вопросы:*

- Классификация ПО.
- Рынок ПО.
- Функциональность ПО.

### ***Лабораторная 2.1***

Цели: описание объектного типа данных.

*Рассматриваемые вопросы:*

1. Объектный анализ предметной области.
2. Диаграммы классов.

*Задания:*

- выполнить объектно-ориентированный анализ проблемной области и проектирование системы для задач по вариантам

### ***Лабораторная 2.2***



Цели: разработка приложения.

*Рассматриваемые вопросы:*

1. Интерфейсы. Композиция.
2. Компонентное программирование.

*Задания:*

- Выполнить проектирования и реализацию проектных решения.
- Выполнить тестирование проектных решения.

**Самостоятельная работа по разделу.** Работа с конспектом лекций и рекомендованной литературой. Защита лабораторных работ.

## **5 Учебно-методическое обеспечение для самостоятельной работы обучающихся**

### ***5.1. Внеаудиторная самостоятельная работа студентов***

В целом внеаудиторная самостоятельная работа студента при изучении курса включает в себя следующие виды работ:

- проработка (изучение) материалов лекций;
- чтение и проработка рекомендованной основной и дополнительной литературы;
- подготовка к практическим занятиям;
- поиск и проработка материалов из Интернет-ресурсов, научных публикаций;
- выполнение домашних заданий в форме практических заданий, докладов и рефератов;
- выполнение курсового проектирования;
- подготовка презентаций для иллюстрации результатов курсового проектирования, докладов;
- подготовка к текущему(индивидуальные опросы) и итоговому (промежуточная аттестация) контролю знаний по дисциплине (экзамен).

Основная доля самостоятельной работы студентов приходится на подготовку к лабораторным занятиям, тематика которых полностью охватывает содержание курса. Самостоятельная работа по подготовке к лекционным занятиям предполагает умение работать с первичной информацией.

*Самостоятельная работа по разделу 1:*

Работа с конспектом лекций и рекомендованной литературой (1-3 и дополнительная).

Подготовка материалов к контрольному опросу по изученным темам, практических занятиях, диалогах с преподавателем и участниками проверки знаний дисциплинарного модуля.

*Самостоятельная работа по разделу 2:*

Работа с конспектом лекций и рекомендованной литературой (1-3 и дополнительная).

Подготовка материалов к контрольному опросу по изученным темам, практических занятиях, диалогах с преподавателем и участниками проверки знаний дисциплинарного модуля.

## **6 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Фонд оценочных средств для проведения промежуточной аттестации обучающихся

по дисциплине «Разработка программных приложений» представлен в приложении к рабочей программе дисциплины и включает в себя:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- типовые контрольные задания или материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

### **Вопросы для проведения промежуточной аттестации по дисциплине (экзамен)**

1. Порядок разработки приложения. Через какие основные шаги разработки программы проходит приложение? Что такое «алгоритмизация»? Какая модель жизненного цикла программного обеспечения поддерживается RAD-технологией? Какие преимущества дает использование RAD-среды? Дать определение проекту. Особенности программного проекта. Примеры проектов.

2. Какие модели жизненного цикла программного обеспечения известны? Укажите достоинства и недостатки каждой модели.

3. Основные принципы ООА. Различия и общее в структурном и объектно-ориентированном подходах.

4. Уровни требований к ПО. Опишите вероятные риски появления «плохих» требований. Перечислите основные стандарты, регламентирующие состав документа «Спецификация требований»

5. Управление проектами. История управления проектами. Категории управления проектами. Треугольник ограничений проекта.

6. Объясните назначение диаграммы классов, диаграммы объектов.

7. Опишите уровни видимости элементов класса.

8. Связи между классами и экземплярами. Типы соединения для показа отношений.

9. Раскройте назначение CRC-карт.

10. Объектно-ориентированные модели системной архитектуры.

11. Модели подсистем, Модели последовательности, Модели конечного автомата.

12. Модели реализации объектно-ориентированных программных систем.

13. Agile-методы.

14. Унифицированный процесс разработки программных средств Rational Unified Process (RUP).

15. Экстремальное программирование.

16. Scram и Канбан (Kanban).

## **7 Рекомендуемая литература**

### **7.1 Основная литература**

1. Иванова Г.С., Технология программирования: Учебник для вузов. – М.: Изд-во МГТУ им. Н.Э Баумана, 2002. – 320 с.

2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник - М. Финансы и статистика, 2000 (2002). - 352 с

3. Вендров А.М. Практикум по проектированию программного обеспечения

экономических информационных систем: учеб. пособие. - М. Финансы и статистика, 2004 (2006). - 192 с.

## 7.2 Дополнительная литература:

4. Соммервилл, Иан. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом Вильямс, 2002. – 624 с.

5. Стандартизация разработки программных средств : [Учеб. пособие для вузов по специальности 351400 "Прикладная информатика (в экономике)"] / В. А. Благодатских, В. А. Волнин, К. Ф. Посакалов ; Под ред. О. С. Разумова. - Москва : Финансы и статистика, 2003. - 284 с.

6. Секунов Н. Разработка приложений на C++ и C#. Библиотека программиста. – СПб.: Питер, 2003. — 608 с. с.

## 8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Научная электронная библиотека eLIBRARY.RU – крупнейший российский информационный портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты : [Электронный ресурс]. - Режим доступа URL:[www.elibrary.ru](http://www.elibrary.ru).

2. Российское образование. Федеральный портал: [Электронный ресурс]. - Режим доступа URL: <http://www.edu.ru>.

3. Software Engineering Body of Knowledge (SWEBOOK) [Электронный ресурс] // The IEEE Computer Society. – USA, Washington, (2001-) – Режим доступа URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering> (дата обращения: 15.01.2019).

4. Стандарты и регламенты [Электронный ресурс] // Федеральное агентство по техническому регулированию и метрологии РФ – М.: Режим доступа URL: <https://www.gost.ru/portal/gost/home/standarts> (дата обращения: 15.01.2019).

5. Электронный фонд правовой и нормативно-технической документации [Электронный ресурс] // М.: АО «Кодекс». – Режим доступа URL: <http://docs.cntd.ru/> (дата обращения: 15.01.2019).

6. Microsoft Docs - Документация Майкрософт для пользователей, разработчиков и ИТ-специалистов – [Электронный ресурс] // Microsoft, М.: (2006-). – Режим доступа URL: <https://docs.microsoft.com/ru-ru/> (дата обращения: 15.01.2019).

7. Документации, обзоры, публикации по продуктам – [Электронный ресурс] // Компания «Интерфейс», М.: (2006-). – Режим доступа URL: <http://www.interface.ru/iservices/catalog.asp?catId=160> (дата обращения: 15.01.2019).

8. Распределённая система управления версиями Git. – [Электронный ресурс] // Режим доступа URL: <https://git-scm.com/> (дата обращения: 15.01.2019).

9. Липаев В.В. Программная инженерия: Комплекс учебников и монографий. – [Электронный ресурс] // Виртуальный компьютерный музей, М.: (2010-). – Режим доступа URL: <http://www.computer-museum.ru/books/lipaev/> (дата обращения: 15.01.2019).

10. Курсы для приобретения знаний и навыков, необходимых для успешной современной карьеры. // Oracle – Режим доступа URL: <https://academy.oracle.com/ru/solutions-curriculum-full.html> (дата обращения: 15.01.2019).

11. Компонентный подход в программировании : курс [Электронный ресурс] // сост. Кулямин В. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/64/64/info> (дата обращения: 15.01.2019).

12. Программирование на Java: курс [Электронный ресурс] // сост. Вязовик Н. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/16/16/info> (дата обращения: 15.01.2019).

13. Углубленное программирование на Java : курс [Электронный ресурс] // сост. Чибриков В. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/3711/953/info> (дата обращения: 15.01.2019).
14. Программирование на языке высокого уровня С# : курс [Электронный ресурс] // сост. Павловская Т. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/629/485/info> (дата обращения: 15.01.2019).
15. Методы и средства инженерии программного обеспечения : курс [Электронный ресурс] // сост. Лаврищева Е., Петрухин В. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/2190/237/info> (дата обращения: 15.01.2019).
16. Язык программирования Java и среда NetBeans : курс [Электронный ресурс] // сост. Монахов В. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/569/425/info> (дата обращения: 15.01.2019).
17. Верификация программного обеспечения: курс [Электронный ресурс] // сост. Налютин Н.Ю., Сеницын С.В. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/1040/209/info> (дата обращения: 15.01.2019).
18. Объектно-ориентированное программирование и программная инженерия: курс [Электронный ресурс] // сост. Meyer, Bertrand. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/2309/609/info> (дата обращения: 15.01.2019).
19. Технологии командной разработки программного обеспечения информационных систем : курс [Электронный ресурс] // сост. Долженко А. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/4806/1054/info> (дата обращения: 15.01.2019).
20. Технологии разработки современных информационных систем на платформе Microsoft.NET: курс [Электронный ресурс] // сост. Павлова Е. – М.: НОУ ИНТУИТ, 2011. Режим доступа URL: <https://www.intuit.ru/studies/courses/1162/285/info> (дата обращения: 15.01.2019).

## **9 Методические указания для обучающихся по освоению дисциплины**

Методика преподавания данной дисциплины предполагает чтение лекций, проведение практических занятий, групповых и индивидуальных консультаций по отдельным вопросам дисциплины. Предусмотрена самостоятельная работа студентов, а также прохождение аттестационных испытаний промежуточной аттестации (экзамен).

Лекции посвящаются рассмотрению наиболее важных концептуальных вопросов: основным вопросам программной инженерии, инструментам и методам программной инженерии; навыками составления технической документации на различных этапах жизненного цикла информационной системы; разработки программного обеспечения.

Целью проведения практических занятий является закрепление знаний студентов, полученных ими в ходе изучения дисциплины на лекциях и самостоятельно. Во время практических занятий выполняются лабораторные работы; на них разбираются конкретные ситуации по изучаемой теме. Для подготовки к занятиям практического типа студенты выполняют проработку рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины, конспектирование источников и работу с конспектом лекций, просмотр рекомендуемой литературы.

При изучении дисциплины используются интерактивные методы обучения, такие как:

### **1. Лекция:**

– лекция-визуализация — подача материала осуществляется средствами технических средств обучения с кратким комментированием демонстрируемых визуальных материалов (презентаций).

### **2. Практические занятия:**

– лабораторные работы — это вид учебной работы, в рамках которого

осуществляется тот или иной эксперимент, направленный на получение результатов, имеющих значение с точки зрения успешного освоения студентами учебной программы.

Методические указания для курсового проектирования по дисциплине «Конструирование программного обеспечения» для студентов направления 09.03.04 «Программная инженерия» / сост. Рычка И.А., Чебанюк С.В. – Петропавловск-Камчатский: КамчатГТУ, 2019.

## **10 Курсовой проект (работа)**

Учебным планом предусмотрено выполнение курсовой работы по дисциплине. Курсовая работа имеет своей целью систематизацию теоретических знаний и практических навыков, полученных при изучении дисциплины, полученные студентами при изучении основного курса по дисциплине «Конструирование программного обеспечения», выполнение обучающимся научного исследования конкретной проблемы. Курсовая работа представляет собой важный этап в подготовке к написанию выпускной квалификационной работы.

Курсовая работа включает в себя следующие элементы:

- титульный лист;
- бланк рецензии (рецензия на курсовую работу в общую нумерацию страниц не включается);
- задание;
- содержание;
- введение;
- основная часть;
- заключение;
- список использованных источников;
- приложения.

«Задание» включает в себя формулировку задания на выполнение курсовой работы и может быть уточнено в процессе выполнения работы.

«Содержание» (оглавление) – представляет собой структуру курсовой работы в виде перечня пунктов (заголовков) с указанием для каждого пункта номера страницы, с которой он начинается.

«Введение» может состоять из краткого описания области исследования, поставленной задачи (предмета курсового исследования) по теме выполняемой работы, формулировки цели курсовой работы и задач, решение которых позволит достичь цель работы, выбранных способов решения этих задач – инструментария исследования (методики, технологии, класс программного обеспечения), основных результатов анализа информации, необходимой для реализации задач курсовой работы, и самих результатов курсового проектирования (программный продукт, пояснительная записка, презентация). Формулировка цели курсовой работы совпадает с формулировкой темы. Перечень задач приводится в последовательности их решения.

Разделы основной части курсовой работы должны содержать описание результата исследования проблемной области и объекта автоматизации (аналитический раздел), результат проектирования архитектуры, пользовательского интерфейса и детального проектирования программного средства (проектный раздел), конструирования и испытания проектных решений (раздел реализации и тестирования).

В заключении курсовой работы суммируются все выводы, подученные ее автором. В относительно небольшом объеме (2-3 страницы) студент в сжатой, лаконичной форме излагает наиболее важные, с его точки зрения, положения рассмотренной задачи, особо подчеркивая самостоятельность сделанных выводов. Самостоятельность и обоснованность выводов наиболее всего ценны в курсовой работе и существенно влияют

на ее оценку.

В приложениях помимо обязательного составленного технического задания на разработку программного средства, могут быть включены руководства пользователя, программиста, коды программ с необходимыми комментариями, таблицы, рисунки и т.п., а также тексты, которые ввиду их громоздкости, большого количества или по другим причинам нецелесообразно размещать в других структурных элементах текстовой части курсовой работы.

### ***10.2 Примерная тематика курсовых работ***

1. Разработка программного обеспечения для автоматизированного учета движения основных средств.

2. Разработка программного обеспечения автоматизированной системы «Конкурс» для секретаря конкурсной комиссии организации.

3. Разработка программного обеспечения автоматизированной системы для калькуляции блюд в кафе.

4. Разработка программного обеспечения для автоматизированной системы учета приема заявок на объявление в ежедневной газете.

5. Разработка программного комплекса системы «Табель рабочего времени».

6. Разработка конфигурации 1С «Секретарь руководителя».

7. Разработка программного комплекса системы «Питомник».

8. Разработка программного комплекса автоматизированной системы «Спортивные соревнования».

9. Разработка программного комплекса автоматизированной системы «Учет материальных ценностей».

10. Разработка программного комплекса автоматизированной системы «Ревизия».

11. Разработка программного комплекса автоматизированной системы «Склад».

12. Разработка программного комплекса «Опрос потребителей».

13. Разработка программного комплекса «Ведение проекта».

Студент может предложить свою тему и после согласования с ведущим преподавателем выполнить курсовую работу на интересующую студента тему.

## **11 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационно-справочных систем**

### ***11.1 Перечень информационных технологий, используемых при осуществлении образовательного процесса***

– электронные образовательные ресурсы, представленные в п. 8 рабочей программы;

– использование слайд-презентаций;

– изучение межгосударственных стандартов ЕСПД на официальном сайте Росстандарта;

– интерактивное общение с обучающимися и консультирование посредством электронной почты.

### ***11.2 Перечень программного обеспечения, используемого при осуществлении образовательного процесса***

При освоении дисциплины используется лицензионное программное обеспечение:

– пакет Microsoft Office;

- Microsoft Visio;
- Microsoft Project;
- Microsoft Visual Studio;
- Java.

### ***11.3 Перечень информационно-справочных систем***

- справочная правовая система Консультант-плюс <http://www.consultant.ru/online>
- справочно-правовая система по законодательству Российской Федерации Гарант <http://www.garant.ru/online>

## **12 Материально-техническое обеспечение дисциплины**

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются учебные аудитории 7-206, 7-401, 7-402, 7-501 с комплектом учебной мебели.

Для самостоятельной работы обучающихся, в том числе для курсового проектирования, используются кабинеты 7-401, 7-402, 7-501, 7-520; каждый кабинет оборудован:

- комплектом учебной мебели,
- компьютерами с доступом в информационно-телекоммуникационную сеть «Интернет» и в электронную информационно-образовательную среду организации,
- техническими средствами обучения для представления учебной информации: аудиторная доска, мультимедийное оборудование (ноутбук, проектор), наглядными пособиями.