

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Курганский государственный университет»  
(КГУ)

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:  
Первый проректор

\_\_\_\_\_ / Т.Р. Змызгова/

« \_\_\_\_ » \_\_\_\_\_ 2024 г.

Рабочая программа учебной дисциплины  
**Конструирование программ**

образовательной программы высшего образования –  
программы бакалавриата

**09.03.04 – Программная инженерия**

Направленность:

**Программное обеспечение автоматизированных систем**

Формы обучения: **очная, заочная**

Рабочая программа дисциплины «Конструирование программ» составлена в соответствии с учебным планом программы бакалавриата: «Программная инженерия» (Программное обеспечение автоматизированных систем), ), утвержденным:

- для очной формы обучения «28» июня 2024 года;
- для заочной формы обучения «28» июня 2024 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «29» августа 2024 года, протокол № 1.

Рабочую программу составил  
доцент кафедры  
«Программное обеспечение  
автоматизированных систем» \_\_\_\_\_

Д.И. Дик

Согласовано:

Заведующий кафедрой  
«Программное обеспечение  
автоматизированных систем» \_\_\_\_\_

С.В. Косовских

Специалист  
по учебно-методической работе  
учебно-методического отдела \_\_\_\_\_

Г.В. Казанкова

Начальник управления  
образовательной деятельности \_\_\_\_\_

И.В. Григоренко

## **1 ОБЪЕМ ДИСЦИПЛИНЫ**

Общая трудоемкость дисциплины - 4 зачетных единицы (144 акад. часа)

### **Очная форма обучения**

Вид учебной работы	Распределение трудоемкости по семестрам и видам учебных занятий, акад. часов	
	Всего	Семестры
		4
<b>Аудиторные занятия</b> в том числе:	<b>48</b>	<b>48</b>
Лекции	16	16
Лабораторные работы	32	32
<b>Самостоятельная работа</b> в том числе:	<b>96</b>	<b>96</b>
Курсовой проект	36	36
Подготовка к экзамену	27	27
Другие виды самостоятельной работы	33	33
<b>Общая трудоемкость дисциплины</b>	<b>144</b>	<b>144</b>
<b>Виды промежуточной аттестации</b>	<b>Экзамен</b>	

### **Заочная форма обучения**

Вид учебной работы	Распределение трудоемкости по курсам и видам учебных занятий, акад. часов		
	Всего	Семестры	
		5	6
<b>Аудиторные занятия</b> в том числе:	<b>14</b>	<b>10</b>	<b>4</b>
Лекции	4	4	–
Практические занятия	4	–	4
Лабораторные работы	6	6	–
<b>Самостоятельная работа</b> в том числе:	<b>130</b>	<b>98</b>	<b>32</b>
Курсовой проект	30	–	30
Подготовка к экзамену	27	27	–
Другие виды самостоятельной работы	73	71	2
<b>Общая трудоемкость дисциплины</b>	<b>144</b>	<b>108</b>	<b>36</b>
<b>Виды промежуточной аттестации</b>	<b>Экзамен</b>	<b>Экзамен</b>	

## **2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

Дисциплина «Конструирование программ» является базовой дисциплиной блока 1 и относится к модулю «Информатика и программирование».

Для освоения дисциплины необходимы компетенции, сформированные в результате изучения дисциплин: «Основы программирования», «Информатика».

Результаты обучения по дисциплине необходимы для выполнения курсовых работ и проектов, а также выпускной квалификационной работы в части проектирования пользовательских интерфейсов программного обеспечения.

Требования к входным знаниям, умениям, навыкам и компетенциям:

- знание базовых принципов функционирования ЭВМ;
- владение основными методами и программными средствами профессиональной деятельности;
- умение разрабатывать прикладное программное обеспечение;
- освоение следующих компетенций на уровне не ниже порогового: ОПК-2 (способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности), ОПК-6 (способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов).

## **3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ**

**Основная цель** изучения дисциплины.

Целью изучения дисциплины является обучение основам конструирования программного обеспечения.

**Задачами дисциплины** является:

- получение общего представления о конструировании программного обеспечения;
- изучение методов совместной работы при разработке программного обеспечения;
- изучение способов интеграции программного обеспечения;
- изучение основных подходов к отладке программного обеспечения;
- изучение способов форматирования и документирования исходного кода;
- изучение методов рефакторинга программного обеспечения;
- изучение конструирования надежного программного обеспечения.

**Компетенции**, формируемые в результате освоения дисциплины:

- способность использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности (ОПК-2);
- способность участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью (ОПК-4);
- способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем (ОПК-5);
- способность разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-6);
- способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-8);

В результате изучения дисциплины обучающийся должен:

*знать:*

- основные методы конструирования программного обеспечения (для ОПК-2);
- основные методы и процессы обеспечения надежности программного обеспечения в процессе его конструирования (для ОПК-2);
- способы документирования исходного кода (ОПК-4);
- принципы рефакторинга программного обеспечения (ОПК-6);

*уметь:*

- устанавливать программное обеспечение для поддержки коллективной разработки программного обеспечения (для ОПК-5);
- применять основы информатики к конструированию программного обеспечения (для ОПК-6);
- использовать инструментальные средства для интеграции программного обеспечения (для ОПК-2);
- использовать инструментальные средства для совместной разработки программного обеспечения (для ОПК-2);
- оформлять документацию к исходному коду (ОПК-4);

*владеть:*

- средствами автоматизированного рефакторинга программного обеспечения (для ОПК-2).
- навыками разработки программного обеспечения на основе тестирования (для ОПК-2);
- навыками рефакторинга программного обеспечения (ОПК-6);
- методами конструирования программного обеспечения (для ОПК-6);
- навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации (для ОПК-8);

– навыками оформления документации к исходному коду (ОПК-4).

Индикаторы и дескрипторы части соответствующей компетенции, формируемой в процессе изучения дисциплины «Конструирование программ», оцениваются при помощи оценочных средств.

Планируемые результаты обучения по дисциплине «Конструирование программ», индикаторы достижения компетенций ОПК-2, ОПК-4, ОПК-5, ОПК-6, ОПК-8 перечень оценочных средств

№ п/п	Код индикатора достижения компетенции	Наименование индикатора достижения компетенции	Код планируемого результата обучения	Планируемые результаты обучения	Наименование оценочных средств
1.	ИД-1 <sub>ОПК-2</sub>	Знать: основные методы конструирования программного обеспечения	З (ИД-1 <sub>ОПК-2</sub> )	Знает: основные методы конструирования программного обеспечения	Тестовые вопросы Отчеты по лабораторным работам Отчеты по практическим занятиям Курсовой проект Вопросы для сдачи экзамена
2.	ИД-2 <sub>ОПК-2</sub>	Знать: основные методы и процессы обеспечения надежности программного обеспечения в процессе его конструирования	З (ИД-2 <sub>ОПК-2</sub> )	Знает: основные методы и процессы обеспечения надежности программного обеспечения в процессе его конструирования	Тестовые вопросы Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи экзамена
3.	ИД-3 <sub>ОПК-2</sub>	Уметь: использовать инструментальные средства для интеграции программного обеспечения	У (ИД-3 <sub>ОПК-2</sub> )	Умеет: использовать инструментальные средства для интеграции программного обеспечения	Тестовые вопросы Отчеты по лабораторным работам Вопросы для сдачи экзамена
4.	ИД-4 <sub>ОПК-2</sub>	Уметь: использовать инструментальные средства для совместной разработки программного обеспечения	У (ИД-4 <sub>ОПК-2</sub> )	Умеет: использовать инструментальные средства для совместной разработки программного обеспечения	Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи экзамена
5.	ИД-5 <sub>ОПК-2</sub>	Владеть: средствами автоматизированного рефакторинга программного обеспечения	В (ИД-5 <sub>ОПК-2</sub> )	Владеет: средствами автоматизированного рефакторинга программного обеспечения	Тестовые вопросы Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи экзамена

6.	ИД-6 <sub>ОПК-2</sub>	Владеть: навыками разработки программного обеспечения на основе тестирования	В (ИД-6 <sub>ОПК-2</sub> )	Владеет: навыками разработки программного обеспечения на основе тестирования	Тестовые вопросы Отчеты по лабораторным работам Отчеты по практическим занятиям Вопросы для сдачи экзамена
7.	ИД-1 <sub>ОПК-4</sub>	Знать: способы документирования исходного кода	З (ИД-1 <sub>ОПК-4</sub> )	Знает: способы документирования исходного кода	Тестовые вопросы Отчеты по лабораторным работам Отчеты по практическим занятиям Курсовой проект Вопросы для сдачи экзамена
8.	ИД-2 <sub>ОПК-4</sub>	Уметь: оформлять документацию к исходному коду	У (ИД-2 <sub>ОПК-4</sub> )	Умеет: оформлять документацию к исходному коду	Тестовые вопросы Отчеты по лабораторным работам Отчеты по практическим занятиям Курсовой проект Вопросы для сдачи экзамена
9.	ИД-3 <sub>ОПК-4</sub>	Владеть: навыками оформления документации к исходному коду	В (ИД-3 <sub>ОПК-4</sub> )	Владеет: навыками оформления документации к исходному коду	Тестовые вопросы Отчеты по лабораторным работам Отчеты по практическим занятиям Курсовой проект Вопросы для сдачи экзамена
10.	ИД-1 <sub>ОПК-5</sub>	Уметь: устанавливать программное обеспечение для поддержки коллективной разработки программного обеспечения	У (ИД-1 <sub>ОПК-5</sub> )	Умеет: устанавливать программное обеспечение для поддержки коллективной разработки программного обеспечения	Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи зачета
11.	ИД-1 <sub>ОПК-6</sub>	Знать: принципы рефакторинга программного обеспечения	З (ИД-1 <sub>ОПК-6</sub> )	Знает: принципы рефакторинга программного обеспечения	Тестовые вопросы Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи экзамена
12.	ИД-2 <sub>ОПК-6</sub>	Уметь: применять основы информатики к конструированию программного обеспечения	У (ИД-2 <sub>ОПК-6</sub> )	Умеет: применять основы информатики к конструированию программного обеспечения	Отчеты по лабораторным работам Курсовой проект Вопросы для сдачи зачета

13.	ИД-3 <sub>ОПК-6</sub>	Владеть: методами конструирования программного обеспечения	В (ИД-3 <sub>ОПК-6</sub> )	Владеет: методами конструирования программного обеспечения	Тестовые вопросы Курсовой проект Отчеты по лабораторным работам Отчеты по практическим занятиям Вопросы для сдачи экзамена
14.	ИД-4 <sub>ОПК-6</sub>	Владеть: навыками рефакторинга программного обеспечения	В (ИД-4 <sub>ОПК-6</sub> )	Владеет: навыками рефакторинга программного обеспечения	Тестовые вопросы Курсовой проект Отчеты по лабораторным работам Вопросы для сдачи экзамена
15.	ИД-1 <sub>ОПК-8</sub>	Владеть: навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	В (ИД-1 <sub>ОПК-8</sub> )	Владеет: навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации	Тестовые вопросы Отчеты по лабораторным работам Вопросы для сдачи экзамена

## 4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Учебно-тематический план

#### Очная форма обучения

Рубеж	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем	
			Лекции	Лабораторные работы
Рубеж 1	1	Введение в конструирование программного обеспечения	1	
	2	Совместное конструирование	4	8
	3	Процесс программирования с псевдокодом	1	
	4	Отладка программного кода	1	
	5	Рефакторинг программного обеспечения	1	8
			Рубежный контроль №1	
Рубеж 2	6	Интеграция программного обеспечения	2	4
	7	Форматирование кода и самодокументирующийся код	4	6
	8	Проектирование по контракту	1	
	9	Разработка через тестирование	1	2
		Рубежный контроль №2		2
<b>Всего:</b>			<b>16</b>	<b>32</b>

## Заочная форма обучения

Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем		
		Лекции	Практич. занятия	Лабораторные работы
1	Введение в конструирование программного обеспечения	0,5		
2	Совместное конструирование	0,5		4
3	Процесс программирования с псевдокодом	0,5		
4	Рефакторинг программного обеспечения	0,5		
5	Интеграция программного обеспечения	0,5		
6	Форматирование кода и самодокументирующийся код	0,5	2	2
7	Проектирование по контракту	0,5		
8	Разработка через тестирование	0,5	2	
	<b>Всего:</b>	<b>4</b>	<b>4</b>	<b>6</b>

### 4.2 Содержание лекционных занятий

#### 1. Введение в конструирование программного обеспечения

Процессы разработки программного обеспечения, относящиеся к конструированию. Обоснование важности конструирования программного обеспечения. Характеристики качества ПО. Методики повышения качества ПО. Относительная эффективность методик контроля качества ПО. Когда выполнять контроль качества ПО. Стоимость контроля качества ПО.

#### 2. Совместное конструирование

Обзор методик совместной разработки ПО. Парное программирование. Формальные инспекции. Анализ кода. Чтение кода. Сравнение методик совместного конструирования.

#### 3. Процесс программирования с псевдокодом

Этапы создания классов и методов. Конструирование методов с использованием программирования с псевдокодом

#### 4. Отладка программного кода

Общие вопросы отладки. Поиск дефекта. Устранение дефекта. Психологические аспекты отладки. Инструменты отладки.

#### 5. Рефакторинг программного обеспечения

Понятие рефакторинга и основания для его проведения (“запахи” плохого кода). Отдельные виды рефакторинга. Безопасный рефакторинг. Стратегии рефакторинга.

#### 6. Интеграция программного обеспечения

Понятие интеграции. Поэтапная и инкрементная интеграция. Нисходящая интеграция. Восходящая интеграция. Сэндвич-интеграция. Риск-ориентированная интеграция. Функционально-ориентированная интеграция.

T-образная интеграция. Ежедневная сборка и дымовые тесты. Непрерывная интеграция. Непрерывное развертывание программного обеспечения. Конвейер развертывания.

### **7. Форматирование кода и самодокументирующийся код**

Основные принципы форматирования. Способы форматирования. Стили форматирования.

Форматирование управляющих структур и классов. Форматирование отдельных операторов. Внешняя и внутренняя документация. Стил программирования как вид документации. Требования к именованию объектов для обеспечения понятности кода. Требования к методам (функциям) для обеспечения понятности кода. Классификация комментариев. Недостатки и преимущества комментариев в коде. Требования к стилю комментирования кода.

### **8. Проектирование по контракту**

Корректность ПО. Формула корректности. Слабые и сильные условия. Предусловия и постусловия. Контракты и надежность ПО. Инварианты класса. Утверждения. Инварианты и варианты цикла

### **9. Разработка через тестирование**

Законы разработки через тестирование. Цикл разработки через тестирование. Разработка, основанная на описании поведения.

## **4.3 Лабораторные работы**

### **Очная форма обучения**

Номер темы	Наименование раздела, темы	Наименование тем лабораторных работ	Норматив времени, час.
2	Совместное конструирование	Система контроля версий исходного кода Subversion	4
		Система контроля версий исходного кода Git	4
5	Рефакторинг программного обеспечения	Рефакторинг программного обеспечения	8
	Рубежный контроль №1		2
6	Интеграция программного обеспечения	Настройка среды непрерывного развертывания с помощью Jenkins	4
7	Форматирование кода и самодокументирующийся код	Система документирования исходных кодов Doxygen	2
7	Форматирование кода и самодокументирующийся код	Стили форматирования исходных кодов	4
9	Разработка через тестирование	Разработка через тестирование	2
	Рубежный контроль №2		2
	<b>Итого</b>		<b>32</b>

## Заочная форма обучения

Номер темы	Наименование раздела, темы	Наименование тем лабораторных работ	Норматив времени, час.
2	Совместное конструирование	Система контроля версий исходного кода Git	4
6	Форматирование кода и самодокументирующийся код	Система документирования исходных кодов Doxygen	2
	<b>Итого</b>		<b>6</b>

## 4.4 Практические занятия

### Заочная форма обучения

Номер темы	Наименование раздела, Темы	Наименование тем практических занятий	Норматив времени, час.
6	Форматирование кода и самодокументирующийся код	Стили форматирования исходных кодов	2
9	Разработка через тестирование	Разработка через тестирование	2
	<b>Итого</b>		<b>4</b>

## 4.5 Курсовой проект

Целью курсового проектирования является формирование навыков в конструировании программного обеспечения.

В рамках курсового проекта выполняется разработка и документирование программного продукта.

Курсовой проект выполняется в соответствии с индивидуальным заданием. Объем курсового проекта 20-25 страниц.

В рамках курсового проекта обучающийся должен решить следующие задачи:

1) разработать программное приложение согласно индивидуальному заданию;

2) документировать программное приложение с использованием внутренней (внедренной в программный код) документации, с использованием разметки системы извлечения документации;

3) сопроводить код приложения модульными тестами.

Код приложения должен разрабатываться с учетом требования его понятности, удобства чтения и самодокументирования.

К защите обучающимся предоставляется пояснительная записка к курсовому проекту, которая должна включать в себя:

– задание;

- описание программного приложения, логики его работы и используемых алгоритмов;
- диаграммы UML, необходимые для описания логики работы приложения (диаграммы классов, последовательности, состояний и т.д.);
- описание стиля форматирования кода;
- документацию, извлеченную из кода системой документирования исходных кодов;
- текст программы и тестов на бумаге или приложенном оптическом диске.

Примерные темы курсового проекта:

- 1) Разработка игры Тетрис
- 2) Разработка игры Диггер
- 3) Разработка игры Пакман (Pac-Man)
- 4) Разработка игры Змейка
- 5) Разработка игры Бомбермен
- 6) Разработка игры Арканойд
- 7) Разработка игры Галакси
- 8) Разработка игры Пинбол
- 9) Разработка игры Фроджер
- 10) Разработка игры Луксор
- 11) Разработка игры Сокобан
- 12) Разработка игры Зума
- 13) Разработка игры Пузыри (bubble shooter) или Пушистики (Wooblies)
- 14) Разработка игры Астероид

## ***5 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ***

Лекционный курс базируется на пассивном методе обучения, реализующем традиционную объяснительно-иллюстративную образовательную технологию, в рамках которой обучающиеся выступают в роли слушателей, воспринимающих учебный материал и участвующих в дискуссиях и экспресс-опросах.

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующего лабораторной работы.

Лабораторные работы и практические занятия (для заочной формы обучения) проводятся на основе интерактивных методов в виде творческих заданий экспериментального характера, направленных не столько на закрепление уже изученного материала, сколько на изучение нового). Задания не имеют однозначного решения и соответствуют целям обучения.

Преподавателем запланировано применение на лабораторных работах и практических занятиях (для заочной формы обучения) технологий развива-

ющейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, выполнение курсового проекта, подготовку к лабораторным занятиям и практическим занятиям (для заочной формы обучения), рубежным контролям (для очной формы обучения), подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

### Рекомендуемый режим самостоятельной работы

#### Очная форма обучения

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
<b>Самостоятельное изучение тем дисциплины:</b>	<b>15</b>
1. Введение в конструирование программного обеспечения	1
2. Совместное конструирование	2
3. Процесс программирования с псевдокодом	1
4. Отладка программного кода	2
5. Рефакторинг программного обеспечения	2
6. Интеграция программного обеспечения	2
7. Форматирование кода и самодокументирующийся код	2
8. Проектирование по контракту	2
9. Разработка через тестирование	1
<b>Подготовка к лабораторным работам (по 2ч. на каждую работу)</b>	<b>14</b>
<b>Подготовка к рубежным контролям (по 2ч. на каждый рубежный контроль)</b>	<b>4</b>
<b>Выполнение курсового проекта</b>	<b>36</b>
<b>Подготовка к экзамену</b>	<b>27</b>
<b>Всего:</b>	<b>96</b>

#### Заочная форма обучения

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
<b>Самостоятельное изучение тем дисциплины:</b>	<b>67</b>
1. Введение в конструирование программного обеспечения	4
2. Совместное конструирование	8

3. Процесс программирования с псевдокодом	5
4. Рефакторинг программного обеспечения	16
5. Интеграция программного обеспечения	10
6. Форматирование кода и самодокументирующийся код	8
7. Проектирование по контракту	8
8. Разработка через тестирование	8
<b>Подготовка к практическим занятиям</b> (по 1ч. на каждое занятие)	<b>2</b>
<b>Подготовка к лабораторным работам</b> (по 2ч. на каждую работу)	<b>4</b>
<b>Выполнение курсового проекта</b>	<b>30</b>
<b>Подготовка к экзамену</b>	<b>27</b>
<b>Всего:</b>	<b>130</b>

## **6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ**

### **6.1 Перечень оценочных средств**

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся (для очной формы обучения)
2. Отчеты обучающихся по лабораторным работам.
3. Отчеты обучающихся по практическим занятиям (для заочной формы обучения).
4. Банк тестовых заданий к рубежным контролям № 1, № 2 (для очной формы обучения).
5. Курсовой проект.
6. Вопросы к экзамену.

### **6.2 Система балльно-рейтинговой оценки работы обучающихся по дисциплине (для очной формы обучения)**

№	Наименование	Содержание					
		Распределение баллов за 4 семестр					
1	Распределение баллов за семестр по видам учебной работы, сроки сдачи учебной работы ( <i>доводятся до сведения обучающихся на первом учебном занятии</i> )	Вид учебной работы:	Посещение лекций	Выполнение и защита лабораторной работы	Рубежный контроль №1	Рубежный контроль №2	экзамен
		Балльная оценка:	до 16	до 34	до 10	до 10	до 30
		Примечания:	2 <sub>б</sub> x 8=16 <sub>б</sub>	4 <sub>б</sub> x 2=8 <sub>б</sub> 2 лр.р. – 2 ч. 5 <sub>б</sub> x 4=20 <sub>б</sub> 4 лр.р. – 4 ч. 6 <sub>б</sub> x 1=6 <sub>б</sub> 1 лр.р. – 8 ч			
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и эк-	60 и менее баллов – неудовлетворительно; 61...73 – удовлетворительно; 74... 90 – хорошо; 91...100 – отлично					

	замена	
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического экзамена по дисциплине, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается.</p> <p>Для получения экзамена или зачета без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.</p> <p>Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине (модулю, практике) не снижается.</p> <p>За академическую активность в ходе освоения дисциплины (модуля, практики), участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30.</p> <p>Основанием для получения дополнительных баллов являются:</p> <ul style="list-style-type: none"> <li>- выполнение дополнительных заданий по дисциплине (модулю, практике); дополнительные баллы начисляются преподавателем;</li> <li>- участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.</li> </ul>
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации (экзамену) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра.</p> <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>
5	Критерии оценки курсового проекта	<p>По курсовому проекту выставляется отдельная оценка. Максимальная сумма по курсовому проекту устанавливается в 100 баллов.</p> <p>При оценке качества выполнения работы и уровня защиты устанавливается следующее распределение баллов:</p> <ul style="list-style-type: none"> <li>а) качество пояснительной записки и графической части – до 40 баллов;</li> <li>б) качество доклада – до 20 баллов;</li> <li>в) качество защиты работы – до 40 баллов.</li> </ul> <p>При рассмотрении качества пояснительной записки принимается к сведению ритмичность выполнения работы, отсутствие ошибок, логичность и последовательность построения материала, правильность выполнения и полнота расчетов, соблюдение требований к оформлению и аккуратность исполнения работы.</p> <p>При оценке качества доклада учитывается уровень владения материалом, степень аргументированности, четкости, последовательности и правильности изложения материала, а также соблюдение регламентов.</p> <p>При оценке уровня качества ответов на вопросы принимается во внимание правильность, полнота и степень ориентированности в материале.</p>

### **6.3 Процедура оценивания результатов освоения дисциплины**

#### **Очная форма обучения**

Рубежный контроль осуществляется в форме фронтального тестирования по разделам дисциплины. На каждое тестирование при рубежном контроле обучающемуся отводится 2 академических часа. Тест для каждого рубежного контроля содержит 10 вопросов. Баллы обучающемуся выставляются в зависимости от числа правильно выбранных ответов. Итоговая оценка по тесту формируется путем суммирования набранных баллов и отнесения их к общему количеству вопросов в задании.

Экзамен проводится в традиционной (устной) форме: обучающийся выполняет задания билета, включающего два теоретических вопроса, и отвечает экзаменатору. Оцениваются полнота и правильность ответов обучающегося на теоретические вопросы билета, его эрудиция в смежных вопросах.

Вопросы к экзамену доводятся до обучающихся на последней лекции в семестре. Время, отводимое обучающемуся на подготовку вопросов, составляет 1 академический час. Каждый вопрос оценивается в 15 баллов.

Результаты текущего контроля успеваемости, экзамена и курсового проекта заносятся преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляются в зачетную книжку обучающегося.

#### **Заочная форма обучения**

Зачет – в форме устного ответа на 2 вопроса. Перечень вопросов преподаватель выдает заранее. Время, отводимое обучающемуся на подготовку вопросов, составляет 1 академический час.

Результаты текущего контроля успеваемости, зачета заносятся преподавателем в зачетную ведомость, которая сдается в организационный отдел института в день зачета, а также выставляются в зачетную книжку обучающегося.

### **6.4 Примеры оценочных средств для рубежных контролей и экзамена**

#### ***Примерные тестовые задания для рубежного контроля №1***

*1. Какая из методик относится совместному конструированию программного обеспечения?*

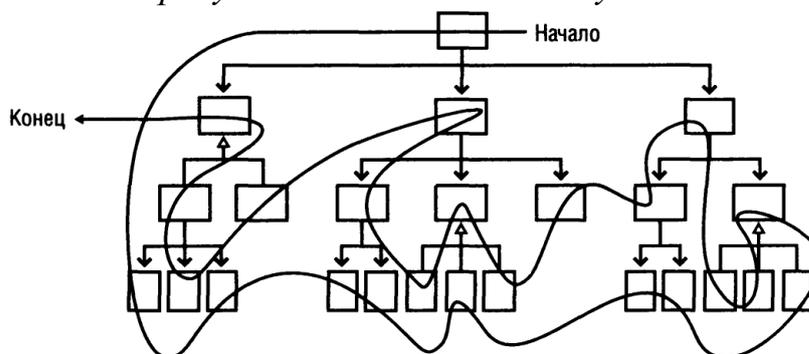
1. Разработка на основе тестирования
2. Парное программирование
3. Интеграция
4. Процесс программирования с псевдокодом

*2. В чем заключается преимущество совместного конструирования?*

1. Повышение качества кода
  2. Повышение эффективности отладки кода
  3. Повышение уровня покрытия кода тестами
3. *Рефакторинг это?*
1. Переработка внутренней структуры программного обеспечения для оптимизации его производительности
  2. Переработка внутренней структуры программного обеспечения для реализации новых требований к системе
  3. Изменение внутренней структуры ПО без изменения его наблюдаемого поведения, призванное облегчить его понимание и удешевить модификацию.

### ***Примерные тестовые задания для рубежного контроля №2***

1. *Инкрементная интеграция предполагает?*
1. Постепенное добавление к системе небольших протестированных компонентов с последующим запуском тестов системы
  2. Объединение протестированных компонентов системы в конце этапа разработки.
  3. Объединение разнородных программных продуктов в единый взаимодействующий комплекс
2. *Представленная на рисунке схема соответствует:*



1. Нисходящей интеграции
  2. Сэндвич-интеграции
  3. Риск-ориентированной интеграции
  4. Функционально-ориентированной интеграции
3. *Дымовые тесты процессе интеграции ПО предназначены?*
1. Для выявления основных проблем, возникающих при добавлении новых компонентов в систему
  2. Для всеобъемлющего ежедневного тестирования системы
  3. Содержат приемочные тесты, выполняемые при ежедневной сборке

### **Примерный перечень вопросов к экзамену**

- 1) Характеристики качества ПО и методики и его повышения

- 2) Методика парного программирования
- 3) Методика проведения формальных инспекций
- 4) Методика проведения анализа кода
- 5) Методика проведения чтения кода
- 6) Методика отладки программного кода
- 7) Понятие интеграции. Поэтапная и инкрементная интеграция.
- 8) Нисходящая интеграция
- 9) Восходящая интеграция
- 10) Сендвич-интеграция
- 11) Риск-ориентированная интеграция
- 12) Функционально-ориентированная интеграция
- 13) Т-образная интеграция
- 14) Ежедневная сборка и дымовые тесты. Непрерывная интеграция.
- 15) Цели форматирования кода. Требования к стилю форматирования кода.
- 16) Стили форматирования блоков кода
- 17) Форматирование управляющих структур и классов
- 18) Форматирование отдельных операторов
- 19) Требования к именованию объектов для обеспечения понятности кода
- 20) Требования к методам (функциям) для обеспечения понятности кода
- 21) Классификация комментариев. Недостатки и преимущества комментариев в коде.
- 22) Требования к стилю комментирования кода
- 23) Понятие рефакторинга и основания для его проведения (“запахи” плохого кода).
- 24) Программирование с псевдокодом
- 25) Проектирование по контракту
- 26) Разработка через тестирование

## **6.5 Фонд оценочных средств**

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов приведены в УМК дисциплины.

## **7 ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА**

### **7.1 Основная литература**

1 Мейер, Б. Основы объектно-ориентированного программирования : учебник / Бертран Мейер. – Москва : Интуит НОУ, 2016. – 970 с.– Режим доступа: <http://www.intuit.ru/studies/courses/71/71/info>, свободный. – Загл. с экрана.

2 Назаров, С.В. Архитектура и проектирование программных систем : Монография / С.В. Назаров. – М.: НИЦ ИНФРА-М, 2014. – 351 с. – Доступ из ЭБС «znanium.com».

3 Гэртнер, М. ATDD - разработка программного обеспечения через приемочные тесты / М. Гэртнер. – М.: ДМК Пресс, 2013. – 232 с. – Доступ из ЭБС «Консультант студента».

### **7.2 Дополнительная литература**

1 Белладжио, Д. Стратегия управления конфигурацией программного обеспечения с использованием IBM Rational ClearCase / Д. Белладжио, Т. Миллиган. – М.: ДМК Пресс. – 384 с. – Доступ из ЭБС «Консультант студента».

2 Роббинс Д. Отладка Windows-приложений / Д. Роббинс. – М.: ДМК Пресс. – 448 с. – Доступ из ЭБС «Консультант студента».

3 Грэхем, Л. Разработка через тестирование для iOS / Л. Грэхем. – М.: ДМК Пресс, 2013. – 272 с. – Доступ из ЭБС «Консультант студента».

## **8 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

1 Дик, Д.И. Методические указания по выполнению лабораторных работ по дисциплине «Конструирование программ» программы бакалавриата 09.03.04 – Программная инженерия [Электронный ресурс] / Д.И. Дик. – Электрон. текстовые дан. – Курган : КГУ, 2019. – 114 с.

2 Дик, Д.И. Методические указания по выполнению практических занятий по дисциплине «Конструирование программ» программы бакалавриата 09.03.04 – Программная инженерия [Электронный ресурс] / Д.И. Дик. – Электрон. текстовые дан. – Курган : КГУ, 2019. – 36 с.

2 Дик, Д.И. Методические указания по выполнению курсового проекта по дисциплине «Конструирование программ» программы бакалавриата 09.03.04 – Программная инженерия [Электронный ресурс] / Д.И. Дик. – Электрон. текстовые дан. – Курган : КГУ, 2019. – 40 с.

## ***9 РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ***

1. Сайт дистанционного обучения в НОУ (Национальный Открытый Университет) «ИНТУИТ» содержит бесплатные курсы, программы повышения квалификации и профессиональной переподготовки, интересные доклады и другую полезную информацию <http://www.intuit.ru>.
2. Федеральный портал «Российское образование» <http://www.edu.ru/>

## ***10 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ***

1. ЭБС «Лань»
2. ЭБС «Консультант студента»
3. ЭБС «Znanium.com»

## ***11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ***

Материально-техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе

Аннотация  
рабочей программы учебной дисциплины  
**«Конструирование программ»**  
образовательной программы высшего образования –  
программы бакалавриата  
**09.03.04 – Программная инженерия**

Направленность:  
**Программное обеспечение автоматизированных систем**  
Формы обучения: **очная**

Трудоемкость дисциплины: 4 ЗЕ (144 академических часа)

Семестры: 4-й

Форма промежуточной аттестации: экзамен

Содержание дисциплины

Совместное конструирование. Процесс программирования с псевдокодом. Отладка программного кода. Рефакторинг программного обеспечения. Интеграция программного обеспечения. Форматирование кода и самодокументирующийся код. Проектирование по контракту. Разработка через тестирование.

