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

_____ / С.Н. Щербич/

«30» августа 2019 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**

Курган 2019

Рабочая программа дисциплины «Конструирование программ» составлена в соответствии с учебным планом программы бакалавриата: «Программная инженерия» (Программное обеспечение автоматизированных систем), утвержденным для очной формы обучения 29 августа 2019 г.

Рабочая программа дисциплины одобрена на заседании кафедры Программного обеспечения автоматизированных систем 30 августа 2019 года, протокол № 1.

Рабочую программу разработал
доцент кафедры ПОАС _____ Д.И.Дик

Заведующий
кафедрой ПОАС _____ Т.Р.Змызгова

Согласовано:

Начальник
Управления
образовательной деятельности _____ С.Н. Синицын

Специалист
по учебно-методической работе
Учебно-методического отдела _____ Г.В. Казанкова

1 ОБЪЕМ ДИСЦИПЛИНЫ

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

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

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Конструирование программного обеспечения» 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», «Введение в программную инженерию».

Дисциплина обеспечивает изучение дисциплин: «Разработка и анализ требований», «Технологии проектирования программных систем», а также выполнение курсовых работ и проектов и выпускной квалификационной работы.

3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплины.

Целью изучения дисциплины является обучение студентов основам конструирования программного обеспечения.

Задачами дисциплины является:

– получение общего представления о конструирования 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).

уметь:

– устанавливать программное обеспечение для поддержки коллективной разработки программного обеспечения (для ОПК-5);

– применять основы информатики в 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).

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

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		2
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
		Всего:	**16**	**32**

4.2 Содержание лекционных занятий

1. Введение в конструирование программного обеспечения

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

6. Интеграция программного обеспечения

Понятие интеграции. Поэтапная и инкрементная интеграция. Нисходящая интеграция. Восходящая интеграция. Сендвич-интеграция. Риск-ориентированная интеграция. Функционально-ориентированная интеграция. Т-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. Стиль 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

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
	Итого		**32**

4.4 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 студент 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 студентом 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 форматирование 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 Арканоид
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 студенты 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.

Конспект каждой лекции завершается перечнем контрольных вопросов, ответы на которые должны быть получены студентом в процессе самостоятельной проработки материала лекции при подготовке к очередному лекционному занятию.

Лабораторные работы 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 технологий развивающейся 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, рубежным контролям подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	**Рекомендуемая трудоемкость, акад. час.**
Самостоятельное изучение тем дисциплины:	**15**
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
Подготовка к лабораторным работам (по 2ч. на каждую работу)	**14**
Подготовка к рубежным контролям (по 2ч. на каждый рубежный контроль)	**4**
Выполнение курсового проекта	**36**
Подготовка к экзамену	**27**
Всего:	**96**

6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1 Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ
2. Отчеты студентов по лабораторным работам.
3. Банк тестовых заданий к рубежным контролям № 1, № 2.
4. Курсовой проект.
5. Вопросы к экзамену.

6.2 Система балльно-рейтинговой оценки работы студентов по дисциплине

<table>
<tr><td>№</td><td>Наименование</td><td colspan="7">Содержание</td></tr>
<tr><td rowspan="5">1</td><td rowspan="5">Распределение баллов за семестр по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)</td><td colspan="7">Распределение баллов за 4 семестр</td></tr>
<tr><td>Вид учебной работы:</td><td>Посещение лекций</td><td>Выполнение и защита лабораторной работы</td><td>Рубежный контроль №1</td><td>Рубежный контроль №2</td><td colspan="2">экзамен</td></tr>
<tr><td>Балльная оценка:</td><td>$2_б$ x 8=$16_б$</td><td>$4_б$ x 2=$8_б$
2 лр.р. – 2 ч.
$5_б$ x 4=$20_б$
4 лр.р. – 4 ч.
$6_б$ x 1=$6_б$
1 лр.р. – 8 ч

34</td><td>10</td><td>10</td><td colspan="2">30</td></tr>
<tr><td colspan="7">Курсовой проект</td></tr>
<tr><td>Полнота и понятность описания логики работы</td><td>Качество форматирования кода, понятность исходного кода</td><td>Полнота и качество документированности исходного кода</td><td>Качество оформления пояснительной записки</td><td>Ритмичность выполнения</td><td>Качество защиты</td><td>Всего</td></tr>
<tr><td>до 15</td><td>до 30</td><td>до 15</td><td>до 10</td><td>до 10</td><td>до 20</td><td>100</td></tr>
</table>

2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена	60 и менее баллов – неудовлетворительно; 61…73 – удовлетворительно; 74… 90 – хорошо; 91…100 – отлично
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического экзамена по дисциплине, возможность получения бонусных баллов	Для допуска к промежуточной аттестации (экзамену) студент должен набрать по итогам текущего и рубежного контроля не менее 50 баллов и должен выполнить все лабораторные работы и курсовой проект. Для получения экзаменационной оценки «автоматически» студенту необходимо набрать следующее минимальное количество баллов: - 68 баллов для получения «автоматически» оценки удовлетворительно». По согласованию с преподавателем студенту, набравшему 68 баллов, могут быть добавлены дополнительные (бонусные) баллы за активность на лабораторных работах, активное участие в научной и методической работе, оригинальность принятых решений в ходе выполнения лабораторных работ, за участие в значимых учебных и внеучебных мероприятиях кафедры и выставлена за экзамен «автоматически» оценка «хорошо» или «отлично».
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) студентов для получения недостающих баллов в конце семестра	В случае если к промежуточной аттестации (экзамену) набрана сумма менее 50 баллов, студенту необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. При этом необходимо проработать материал всех пропущенных лабораторных работ. Формы дополнительных заданий (назначаются преподавателем): - выполнение и защита пропущенной лабораторной работы – до 6 баллов. 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.

6.3 Процедура оценивания результатов освоения дисциплины

Рубежный контроль осуществляется в форме фронтального тестирования по разделам дисциплины. На каждое тестирование при рубежном контроле студенту отводится 2 академических часа. Тест для каждого рубежного контроля содержит 10 вопросов. Баллы студенту 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: студент выполняет задания билета, включающего два теоретических вопроса, и отвечает экзаменатору. Оцениваются полнота и правильность ответов студента на теоретические вопросы билета, его эрудиция в смежных вопросах.

Вопросы к экзамену доводятся до студентов на последней лекции в семестре. Время, отводимое студенту 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 выставляют-

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?

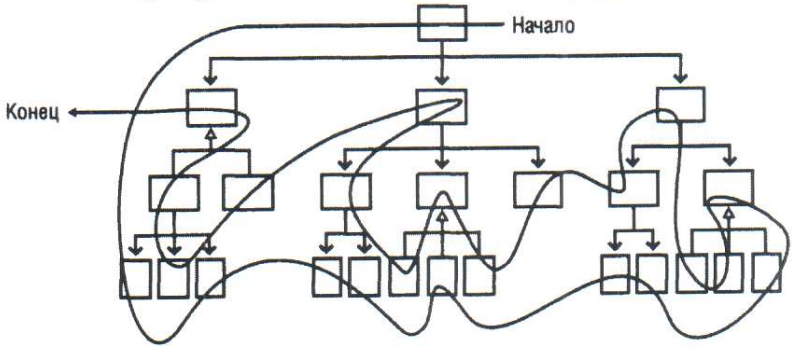
1. Переработка внутренней структуры программного обеспечения для оптимизации его производительности
2. Переработка внутренней структуры программного обеспечения для реализации новых требований к системе
3. Изменение внутренней структуры ПО без изменения его наблюдаемого поведения, призванное облегчить 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 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/

3. Информационный сайт, содержащий справочные материалы по информатике, которые включают в себя курс лекций, схемы, презентации, рефераты и др. informatikaplus.narod.ru

4. Сайт о высоких технологиях, новости индустрии из мира компьютерного «железа», тестовые испытания и обзоры оборудования IXBT.com.

5. Портал «Информационно-коммуникационные технологии в образовании» http://www.ict.edu.ru.

10 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

При чтении лекций используются слайдовые презентации.

Минимальные требования к операционной системе и программному обеспечению компьютера, используемого при показе слайдовых презентаций: Windows XP.

11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя учебные лаборатории и классы, оснащенные современными компьютерами (все – в стандартной комплектации для лабораторных занятий и самостоятельной работы), объединенными локальными вычислительными сетями с выходом в Интернет, мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, мультимедийный экран). Дисциплина должна быть поддержана соответствующими лицензионными программными продуктами.

Программные средства обеспечения учебного процесса должны включать: базовые (операционные системы (Windows); инструментальные средства программирования) и вспомогательные (программы презентационной графики; текстовые редакторы; графические редакторы).

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.

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
_____ С.Н. Щербич/
«30» августа 2019 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: **заочная**

Курган 2019

Рабочая программа дисциплины «Конструирование программ» составлена в соответствии с учебным планом программы бакалавриата: «Программная инженерия» (Программное обеспечение автоматизированных систем), утвержденным для заочной формы обучения 29 августа 2019 г.

Рабочая программа дисциплины одобрена на заседании кафедры Программного обеспечения автоматизированных систем 30 августа 2019 года, протокол № 1.

Рабочую программу разработал
доцент кафедры ПОАС _____ Д.И.Дик

Заведующий
кафедрой ПОАС _____ Т.Р.Змызгова

Согласовано:

Начальник
Управления
образовательной деятельности _____ С.Н. Синицын

Специалист
по учебно-методической работе
Учебно-методического отдела _____ Г.В. Казанкова

1 ОБЪЕМ ДИСЦИПЛИНЫ

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

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

2 МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Конструирование программного обеспечения» 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», «Введение в программную инженерию».

Дисциплина обеспечивает изучение дисциплин: «Разработка и анализ требований», «Технологии проектирования программных систем», а также выполнение курсовых работ и проектов и выпускной квалификационной работы.

3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Основная цель изучения дисциплины.

Целью изучения дисциплины является обучение студентов основам конструирования программного обеспечения.

Задачами дисциплины является:

– получение общего представления о конструирования 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).

уметь:

– устанавливать программное обеспечение для поддержки коллективной разработки программного обеспечения (для ОПК-5);

– применять основы информатики в 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).

4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Учебно-тематический план

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	
	Всего:	**4**	**4**	**6**

4.2 Содержание лекционных занятий

1. Введение в конструирование программного обеспечения

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

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

5. Интеграция программного обеспечения

Понятие интеграции. Поэтапная и инкрементная интеграция. Нисходящая интеграция. Восходящая интеграция. Сендвич-интеграция. Риск-ориентированная интеграция. Функционально-ориентированная интеграция. Т-образная интеграция. Ежедневная сборка и дымовые тесты. Непрерывная

интеграция. Непрерывное развертывание программного обеспечения. Конвейер развертывания.

6. Форматирование кода и самодокументирующийся код

Основные принципы форматирования. Способы форматирования. Стили форматирования.

Форматирование управляющих структур и классов. Форматирование отдельных операторов. Внешняя и внутренняя документация. Стиль 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.

7. 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

8. Разработка через тестирование

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

4.3 Лабораторные работы

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
	Итого		**6**

4.4 Практические занятия

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

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 студент 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 студентом 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 форматирование 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 Арканоид
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 студенты 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 и практического занятия.

Конспект каждой лекции завершается перечнем контрольных вопросов, ответы на которые должны быть получены студентом в процессе самостоятельной проработки материала лекции при подготовке к очередному лекционному занятию.

Лабораторные работы и практические занятия 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 технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, выполнение курсового проекта, подготовку к лабораторным и практическим занятиям, подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	**Рекомендуемая трудоемкость, акад. час.**
Самостоятельное изучение тем дисциплины:	**67**
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
Подготовка к практическим занятиям (по 1ч. на каждое занятие)	**2**

Подготовка к лабораторным работам (по 2ч. на каждую работу)	**4**
Выполнение курсового проекта	**30**
Подготовка к экзамену	**27**
Всего:	**130**

6 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1 Перечень оценочных средств

1. Отчеты студентов по лабораторным работам.
2. Отчеты студентов по практическим занятиям.
3. Курсовой проект.
4. Вопросы к экзамену.

6.2 Процедура оценивания результатов освоения дисциплины

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

Вопросы к экзамену доводятся до студентов на последней лекции в семестре. Время, отводимое студенту на подготовку вопросов, составляет 1 академический час.

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 студента.

6.3 Пример оценочных средств для экзамена

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) Блочные сборки 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

6.4 Фонд оценочных средств

Полный банк заданий для текущего 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 практических занятий по дисциплине «Конструирование программ» программы бакалавриата 09.03.04 – Программная инженерия [Электронный ресурс] / Д.И. Дик. – Электрон. текстовые дан. – Курган : КГУ, 2019. – 44 с.

2 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. – 70 с.

3 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/

3. Информационный сайт, содержащий справочные материалы по информатике, которые включают в себя курс лекций, схемы, презентации, рефераты и др. informatikaplus.narod.ru

4. Сайт о высоких технологиях, новости индустрии из мира компьютерного «железа», тестовые испытания и обзоры оборудования IXBT.com.

5. Портал «Информационно-коммуникационные технологии в образовании» http://www.ict.edu.ru.

10 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

При чтении лекций используются слайдовые презентации.

Минимальные требования к операционной системе и программному обеспечению компьютера, используемого при показе слайдовых презентаций: Windows XP.

11 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя учебные лаборатории и классы, оснащенные современными компьютерами (все – в стандартной комплектации для практических занятий и самостоятельной работы), объединенными локальными вычислительными сетями с выходом в Интернет, мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, мультимедийный экран). Дисциплина должна быть поддержана соответствующими лицензионными программными продуктами.

Программные средства обеспечения учебного процесса должны включать: базовые (операционные системы (Windows); инструментальные средства программирования) и вспомогательные (программы презентационной графики; текстовые редакторы; графические редакторы).

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: **заочная**

Трудоемкость дисциплины: 4 ЗЕ (144 академических часа)
Семестры: 5 и 6
Форма промежуточной аттестации: экзамен

Содержание дисциплины

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