

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:
Проректор по учебной работе,
качеству образования – первый
проректор
Хагуров Т.А.
«30» мая 2025 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.35 «Технологии программирования»

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Математическое моделирование в естествознании
и технологиях

Форма обучения очная

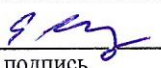
Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Технологии программирования» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная математика и информатика

Программу составил:

Евдокимов А.А., канд. физ.-мат. наук, доцент кафедры математического моделирования КубГУ


подпись

Рабочая программа дисциплины «Технологии программирования» утверждена на заседании кафедры математического моделирования протокол №11 от «22» мая 2025 г.

Заведующий кафедрой акад. РАН, д-р физ.-мат. наук, проф. Бабешко В.А.


подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №4 от «23» мая 2025 г.

Председатель УМК факультета
д-р. техн. наук, доцент Коваленко А.В.


подпись

Рецензенты:

Марков В.Н. д-р техн. наук, профессор кафедры Кафедра информационных систем и программирования ФГБОУ ВО «КубГТУ»

Синица С.Г., канд физ.-мат. наук, доцент кафедры информационных технологий ФГБОУ ВО «КубГУ»

1 Цели и задачи изучения дисциплины

1.1 Цель освоения дисциплины

Дисциплина «Технологии программирования» ставит своей целью изучение теоретических основ современных технологий программирования и получение практических навыков их реализации.

Цели дисциплины соответствуют формируемой компетенции УК-2 и ОПК-5.

1.2 Задачи дисциплины

Основные задачи дисциплины:

- формирование систематизированного представления о концепциях и принципах организации, положенных в основу современных технологий программирования;
- получение практической подготовки в области применения технологии программирования.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Технологии программирования» относится к дисциплинам по выбору вариативной части Блока 1 "Дисциплины (модули)" учебного плана подготовки бакалавра. В профессиональной подготовке выпускника курс завершает линию обучения программиста.

Данный курс наиболее тесно связан со следующими дисциплинами:

Методы программирования, Объектно-ориентированное программирование, Базы данных, Администрирование БД, Анализ, проектирование и разработка БД, Технологии XML.

Для полноценного освоения курса студент должен в достаточно большом объеме владеть Web-программированием и объектным программированием, в первую очередь, программированием и технологиями Java. Необходимо хорошее знание основ баз данных

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Программа определяет общий объем знаний, позволяющий сформировать у студента целостное представление о современных программных технологиях и соответствующие умения и навыки. Вместе с тем, из-за обширности изучаемого предмета в изложении многих разделов курса приходится ограничиваться подробным изложением только части возможных подходов, а описание остальных неизбежно имеет, в основном, информационный характер. В процессе освоения дисциплины у студента формируется следующие компетенции

Изучение данной учебной дисциплины направлено на овладение обучающимся следующими компетенциями:

УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
Знать	ИУК-2.4 (06.001 D/03.06 Зн.1) Виды архитектуры программного обеспечения и принципы построения архитектуры программного обеспечения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений ИУК-2.5 (06.001 D/03.06 Зн.2) Оптимальные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, исходя из имеющихся ресурсов и ограничений ИУК-2.6 (06.001 D/03.06 Зн.3) Способы определения круга задач, методы и средства проектирования программного обеспечения, выбирать оп-

тимальные способы решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

- Уметь** ИУК-2.14 (06.001 D/03.06 У.1) Выбирать и использовать оптимальные существующие типовые решения и шаблоны проектирования программного обеспечения
ИУК-2.15 (06.001 D/03.06 У.2) Определять круг задач, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов, исходя из имеющихся ресурсов и ограничений
ИУК-2.16 (06.016 А/06.6 У.1) Способен определять круг задач и выбирать оптимальную структуру документации ПО
- Владеть** ИУК-2.23 (06.001 D/03.06 Тд.2) Определение и проектирование структур данных в заданной предмет-ной области
ИУК-2.26 (06.016 А/30.6 Тд.1) Качественный анализ рис-ков в проектах в области ИТ, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
- ОПК-5** **Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения**
- Уметь** ИОПК-5.3 (У.2) Применять методы и средства проектирования системного и прикладного программного обеспечения, структур данных, баз данных, программных интерфейсов
- Владеть** ИОПК-5.4 (Тд.1) Разработка математически сложных алгоритмов, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения
ИОПК-5.5 (Тд.2) Проектирование структур данных при разработке системного и прикладного программного обеспечения

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Код и наименование индикатора	Результаты обучения по дисциплине
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	
ИУК-2.4 (06.001 D/03.06 Зн.1) Виды архитектуры программного обеспечения и принципы построения архитектуры программного обеспечения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	Знает: способы использования современных методов для решения научных и практических задач; принципы выбора методов и средств изучения математической модели; подходы к верификации моделей и оценке затрат на их реализацию; пакеты прикладных программ, используемые при построении моделей
ИУК-2.5 (06.001 D/03.06 Зн.2) Оптимальные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, исходя из имеющихся ресурсов и ограничений	Умеет: строить математические модели и оценивать их адекватность; содержательно интерпретировать результаты
ИУК-2.6 (06.001 D/03.06 Зн.3) Способы определения круга задач, методы и средства проектирования программного обеспечения, выбирать оптимальные способы решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	Владеет: навыками анализа, сопоставления и обобщения результатов теоретических и практических исследований в предметной области; основными методами исследования и математических моделей; навыками
ИУК-2.14 (06.001 D/03.06 У.1) Выбирать и	

Код и наименование индикатора	Результаты обучения по дисциплине
использовать оптимальные существующие типовые решения и шаблоны проектирования программного обеспечения ИУК-2.15 (06.001 D/03.06 У.2) Определять круг задач, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов, исходя из имеющихся ресурсов и ограничений ИУК-2.16 (06.016 А/06.6 У.1) Способен определять круг задач и выбирать оптимальную структуру документации ПО ИУК-2.23 (06.001 D/03.06 Тд.2) Определение и проектирование структур данных в заданной предметной области ИУК-2.26 (06.016 А/30.6 Тд.1) Качественный анализ рисков в проектах в области ИТ, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	содержательной интерпретации результатов; навыками использования пакетов прикладных программ для обеспечения процесса моделирования
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	
ИОПК-5.3 (У.2) Применять методы и средства проектирования системного и прикладного программного обеспечения, структур данных, баз данных, программных интерфейсов ИОПК-5.4 (Тд.1) Разработка математически сложных алгоритмов, изменение и согласование архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения ИОПК-5.5 (Тд.2) Проектирование структур данных при разработке системного и прикладного программного обеспечения	.Умеет: применять методы и средства проектирования системного и прикладного программного обеспечения, структур данных, баз данных, программных интерфейсов Владеет: навыками разработки математически сложных алгоритмов, проектирования структур, изменения и согласования архитектуры программного обеспечения с системным аналитиком и архитектором программного обеспечения

Процесс освоения дисциплины «Технологии программирования» направлен на получения необходимого объема знаний, отвечающих требованиям ФГОС ВО и обеспечивающих успешное ведение бакалавром производственной и научно-исследовательской деятельности, владение методикой формулирования и решения прикладных задач, а также на выработку умений применять на практике методы прикладной математики и информатики.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины.

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоемкость дисциплины составляет 2 зачетных единицы, 72 академических часа в восьмом семестре. Курс «Технологии программирования» состоит из лекционных и лабораторных занятий, сопровождаемых регулярной индивидуальной работой преподавателя со студентами в процессе самостоятельной работы. В конце

семестра проводится зачет. Программой дисциплины предусмотрены 14 часов лекционных занятий, 42 часов лабораторных занятий, 13,8 часов самостоятельной работы и 2 часа контролируемой самостоятельной работы.

Вид учебной работы	Всего часов	Семестр (часы)
		8
Контактная работа (всего)	58,2	58,2
В том числе:		
Занятия лекционного типа	14	14
Занятия семинарского типа (семинары, практические занятия)	–	–
Лабораторные занятия	42	42
Иная контактная работа:		
Контроль самостоятельной работы (КСР)	2	2
Промежуточная аттестация (ИКР)	0,2	0,2
Самостоятельная работа (всего)	13,8	13,8
В том числе:		
Курсовая работа	–	–
Проработка учебного (теоретического) материала	8,8	8,8
Подготовка к текущему контролю	5	5
Контроль: экзамен		
Общая трудоемкость	час.	72
	в том числе контактная работа	58,2
	зач. ед	2

2.2 Содержание дисциплины:

№	Наименование разделов	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛЗ	СРС
1	Программная инженерия.	1	1	–	–
2	Модели жизненного цикла ПО	3	1	2	–
3	Составление требований к программному обеспечению.	5	1	4	–
4	Проектирование и программирование	6,8	1	4	1,8
5	Шаблоны проектирования	8	2	4	2
6	Управление версиями. Модель.	6	1	4	1
7	Управление версиями. Subversion. Tortoise.	7	1	4	2
8	Документирование.	6	1	4	1
9	Тестирование.	7	1	4	2
10	Утилита JUnit. Рефакторинг.	6	1	4	1
11	Качество программного обеспечения.	7	1	4	2
12	Программная инженерия.	7	2	4	1
Промежуточная аттестация (ИКР)		0,2			
Контроль самостоятельной работы (КСР)		2			
Итого:		72	14	42	13,8

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы дисциплины, изучаемые в 8 семестре.

Примечание: ЛР – лабораторные занятия, СРС – самостоятельная работа студента, КСР – контролируемая самостоятельная работа.

2.3 Содержание разделов дисциплины:

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Программная инженерия.	Проект SWEBOOK. Десять областей знания в программной инженерии.	Контрольное задание
2	Модели жизненного цикла ПО	Последовательная, инкрементная, спиральная модели. Эволюционные модели. Прототипирование.	Контрольное задание
3	Составление требований к ПО.	Анализ. Виды и источники требований. Спецификации требований к системе и ПО.	Контрольное задание
4	Проектирование и программирование	Модульность. Структуры данных. Структурное программирование. Нисходящее и восходящее проектирование. Исключительные ситуации, их распространение и обработка.	Контрольное задание
5	Шаблоны проектирования	Особенности разработки объектных программ. Шаблоны проектирования.	Контрольное задание
6	Управление версиями. Модель.	Управление файлами и каталогами во времени. Модели “блокирование – изменение – разблокирование” и “копирование – изменение – слияние”. Рабочие копии. Коллективная работа	Контрольное задание
7	Управление версиями. Subversion. Tortoise.	Простой рабочий цикл. Фиксация изменений. Анализ истории. Ветвление и слияние. Ветви. Копирование изменений между ветвями.	Контрольное задание
8	Документирование.	Работа с Docbook.	Контрольное задание
9	Тестирование.	Тестирование программного обеспечения по спецификации и по коду.	Контрольное задание
10	Утилита JUnit. Рефакторинг.	Утилита JUnit. Понятие рефакторинга.	Контрольное задание
11	Качество ПО	Соглашения о кодировании. Факторы качества. Метрики кода, их виды и реализация в Subversion.	Контрольное задание

2.3.1 Занятия лекционного типа

№ раздела	Тема лекций
1	Программная инженерия. (Проект SWEBOOK. Десять областей знания в программной инженерии)
2	Модели жизненного цикла ПО (Последовательная, инкрементная, спиральная модели. Эволюционные модели. Прототипирование.)
3	Составление требований к программному обеспечению. (Анализ. Виды и источники требований. Спецификации требований к системе и ПО.)
4	Проектирование и программирование (Модульность. Структуры данных. Структурное программирование. Нисходящее и восходящее проектирование. Исключительные ситуации, их распространение и обработка.)
5	Шаблоны проектирования (Особенности разработки объектных программ. Шаблоны проектирования.)
6	Управление версиями. Модель. (Управление файлами и каталогами во времени. Модели “блокирование – изменение – разблокирование” и “копирование – изменение – слияние”. Рабочие копии. Коллективная работа)

7	Управление версиями. Subversion. Tortoise. (Простой рабочий цикл. Фиксация изменений. Анализ истории. Ветвление и слияние. Ветви. Копирование изменений между ветвями.)
8	Документирование. (Работа с Docbook.)
9	Тестирование. (Тестирование программного обеспечения по спецификации и по коду.)
10	Утилита JUnit. Рефакторинг. (Утилита JUnit. Понятие рефакторинга.)
11	Качество программного обеспечения. (Соглашения о кодировании. Факторы качества. Метрики кода, их виды и реализация в Subversion)

2.3.2 Занятия семинарского типа

Учебный план не предусматривает занятий семинарского типа по дисциплине «Технологии программирования».

2.3.3 Лабораторные занятия

№ работы	№ раздела дисциплины	Наименование лабораторных работ
1	1	Программная инженерия.
2	2	Модели жизненного цикла ПО.
3	3	Составление требований к программному обеспечению.
4	4	Проектирование и программирование.
5	5	Шаблоны проектирования.
6	6	Управление версиями. Модель.
7	7	Управление версиями. Subversion. Tortoise.
8	8	Документирование.
9	9	Тестирование.
10	10	Утилита JUnit. Рефакторинг.
11	11	Качество программного обеспечения.

2.3.4 Примерная тематика курсовых работ (проектов)

Учебный план не предусматривает курсовых работ по дисциплине «Технологии программирования».

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	Подготовка к текущему контролю, подготовка индивидуальных заданий	Смирнов А.А. Технологии программирования. М.: Евразийский открытый институт, 2011. 192 с. [Электронный ресурс]. - Режим доступа: http://biblioclub.ru/index.php?page=book&id=90777 . 1. Антамошкин О.А. Программная инженерия. Теория и практика. Красноярск: Сибирский федеральный университет, 2012. 247 с. [Электронный ресурс]. - Режим доступа: http://biblioclub.ru/index.php?page=book&id=363975

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

2.5 Самостоятельное изучение разделов дисциплины

Целью самостоятельной работы является углубление знаний, полученных в результате аудиторных занятий, выработка навыков индивидуальной работы, закрепление навыков, сформированных во время лабораторных занятий.

Содержание приведенной основной и дополнительной литературы позволяет охватить все вопросы, рассматриваемые в настоящем курсе.

Раздел 1. Программная инженерия. Проект SWEBOK. Десять областей знания в программной инженерии.

Раздел 2. Модели жизненного цикла ПО. Последовательная, инкрементная, спиральная модели. Эволюционные модели. Прототипирование.

Раздел 3. Составление требований к программному обеспечению. Анализ. Виды и источники требований. Спецификации требований к системе и ПО.

Раздел 4. Проектирование и программирование. Модульность. Структуры данных. Структурное программирование. Нисходящее и восходящее проектирование. Исключительные ситуации, их распространение и обработка.

Раздел 5. Шаблоны проектирования. Особенности разработки объектных программ.

Раздел 6. Управление версиями. Модель. Управление файлами и каталогами во времени. Модели “блокирование-изменение-разблокирование” и “копирование-изменение-слияние”. Рабочие копии. Коллективная работа

Раздел 7. Управление версиями. Subversion. Tortoise. Простой рабочий цикл. Фиксация изменений. Анализ истории. Ветвление и слияние. Ветви. Копирование изменений между ветвями.

Раздел 8. Документирование. Работа с Docbook.

Раздел 9. Тестирование. Тестирование программного обеспечения по спецификации и по коду.

Раздел 10. Утилита JUnit. Рефакторинг. Утилита JUnit. Понятие рефакторинга.

Раздел 11. Качество программного обеспечения. Соглашения о кодировании. Факторы качества. Метрики кода, их виды и реализация в Subversion.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач.

Программа по дисциплине предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; работа над индивидуальными заданиями с использованием пакетов прикладных программ, разбор конкретных ситуаций на практических занятиях.

Компьютерные технологии предоставляют средства разнопланового отображения алгоритмов и демонстрационного материала.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и бакалаврами во время лекций и анализа результатов самостоятельной работы. Это обусловлено тем, что в процессе моделирования часто встречаются задачи, для которых единых подходов не существует. При исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) – совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- анализ конкретных ситуаций – анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	Общее количество часов
8	ЛР	Компьютерные занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент» (разделы 4–5)	10

С точки зрения применяемых методов используются как традиционные занятия, так и интерактивная подача материала с мультимедийной системой. Компьютерные технологии в данном случае обеспечивают возможность разнопланового отображения алгоритмов и демонстрационного материала. Такое сочетание позволяет оптимально использовать отведенное время и раскрывать логику и содержание дисциплины.

Цель *лабораторного занятия* – научить применять теоретические знания при решении конкретных задач. Лабораторные занятия проводятся в компьютерных классах.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лабораторных занятий.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Учебная деятельность проходит в соответствии с графиком учебного процесса. Процесс самостоятельной работы контролируется во время аудиторных занятий и индивидуальных консультаций. Самостоятельная работа студентов проводится в форме изучения отдельных теоретических вопросов по предлагаемой литературе.

Фонд оценочных средств дисциплины состоит из средств текущего контроля (см. список лабораторных работ, задач и вопросов) и итоговой аттестации (зачёта).

В качестве оценочных средств, используемых для текущего контроля успеваемости, предлагается перечень вопросов, которые прорабатываются в процессе освоения курса. Данный перечень охватывает все основные разделы курса, включая знания, получаемые во время самостоятельной работы. Кроме того, важным элементом технологии является самостоятельное решение студентами и сдача заданий. Это полностью индивидуальная форма обучения. Студент рассказывает свое решение преподавателю, отвечает на дополнительные вопросы.

Оценка успеваемости осуществляется по результатам: самостоятельного выполнения лабораторных работ, устного опроса при сдаче выполненных самостоятельных заданий, индивидуальных лабораторных заданий и защиты групповых заданий.

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

Примерные задания на лабораторные работы

SVN

1. Изучение базовых принципов работы с SVN
1. Зарегистрироваться на assembla.com
2. Получить доступ к проекту у преподавателя.
3. Создать каталог проекта.
4. Создать базовую структуру репозитория.
5. Начать разработку кода.
6. Сделать не менее пяти фиксаций, касающихся добавления файлов и редактирования кода.
2. Работа с ветками в SVN
1. На основе trunk создать ветку в каталоге /branches.
2. Переключить рабочую копию на созданную ветку.
3. Сделать не менее пяти фиксаций, касающихся добавления файлов и редактирования кода.
4. Переключить рабочую копию на trunk.
5. Влить все изменения из ветки с помощью команды merge.
6. Удалить ветку.

ДОКУМЕНТИРОВАНИЕ

1. Оформить краткое описание разработанного продукта (не менее двух абзацев).
2. Рассмотреть с примерами кода не менее трех сценариев использования вашего приложения.
3. Проверить корректность созданного DocBook файла с помощью средств редактора или одного из онлайн валидаторов, например <http://www.mashupsoft.com/docbooks/validator/>.

ТЕСТИРОВАНИЕ И JUNIT

1. Одиночные тесты
1. Спроектируйте класс суммирования элементов массива ArraySum, содержащий статический метод Sum(), принимающий массив в качестве параметра.
2. Разработайте класс ArraySumTest для тестирования ArraySum.
3. Реализуйте класс ArraySum. Добейтесь корректного выполнения тестов.
2. JUnit: тесты над общими наборами объектов (fixtures)

1. Расширьте класс `ArraySum`, запланировав в нем конструктор и метод экземпляра `Sum()`, суммирующий массив, заданный в конструкторе.

2. Расширьте класс `ArraySumTest`, превратив его в `fixture` (тест, использующий общий набор объектов). В качестве общего объекта должен выступать массив для суммирования.

3. Реализуйте дополнительную функциональность в классе `ArraySum`. Добейтесь корректного выполнения тестов.

3. JUnit: тестирование исключений

- Добавьте в класс `ArraySumTest` метод, тестирующий поведение класса `ArraySum` при передаче в его статический метод `Sum()` значения `null`. Оцените результат выполнения теста и добейтесь его корректного выполнения.

- Откорректируйте тестирующий метод так, чтобы он ожидал генерации исключения и выдавал ошибку в случае, если исключение не возникает. Добейтесь корректного выполнения теста.

4. JUnit: группы тестов (test suites)

- Добавьте в проект новый класс `ArrayProd`, реализующий произведение элементов массива и соответствующий тестовый класс `ArrayProdTest`.

- Объедините `ArraySumTest` и `ArrayProdTest` в рамках группы тестов `AllTests`.

- Выполните группу тестов за один раз.

Разработайте на основе тестов один из следующих классов.

1. Разработать класс, представляющий студента. Студент характеризуется именем, фамилией, группой и набором экзаменов, которые он сдавал. Экзамен характеризуется названием предмета, оценкой студента по нему и датой сдачи (год, семестр). Группа характеризуется курсом и факультетом.

Необходимые операции таковы:

- узнать наивысшую оценку среди всех экзаменов по данному предмету
- добавить ему оценку по экзамену
- удалить для него оценку по экзамену;
- если он такой экзамен не сдавал - сгенерировать исключение
- узнать число экзаменов, которые он сдал с указанной оценкой
- узнать его средний балл за указанный семестр;

2. Разработать класс, представляющий общежитие. Общежитие характеризуется улицей, номером дома, факультетом и набором комнат. Комната характеризуется номером, вместимостью и числом занятых мест. Факультет характеризуется институтом, названием и числом студентов.

Необходимые операции таковы:

- открыть комнату для заселения
- занять комнату
- в случае если она уже занята, сгенерировать исключение
- освободить комнату
- вернуть число свободных комнат (полностью/частично)
- узнать, какой процент студентов института живет в общежитии

Перечень вопросов, выносимых на зачет

1. Программа, программная система, программный продукт, программное обеспечение.

2. Программная инженерия. Проект SWEBOK. Десять областей знания в программной инженерии.

3. Причины сложности разработки программного обеспечения.

4. Модели жизненного цикла ПО. Последовательная, инкрементная, спиральная модели. Эволюционные модели. (Стандарт ISO/IEC 12207 (ГОСТ Р ИСО/МЭК 12207)).
5. Прототипирование.
6. Составление требований к программному обеспечению. Анализ. Виды и источники требований. Спецификации требований к системе и ПО.
7. Проектирование и программирование. Модульность. Структуры данных. Структурное программирование.
8. Нисходящее и восходящее проектирование.
9. Исключительные ситуации, их распространение и обработка.
10. Шаблоны проектирования. Особенности разработки объектных программ.
11. Понятие модуля. Критерии качества проектирования модулей и классов.
12. Конфигурации и управление ими.
13. Варианты использования
14. Модель зрелости возможностей СММ.
15. Особенности проектирования интерфейса пользователя.
16. Управление версиями. Модель. Управление файлами и каталогами во времени. Модели “блокирование-изменение-разблокирование” и “копирование-изменение-слияние”. Рабочие копии.
17. Управление версиями. Subversion. Tortoise. Простой рабочий цикл. Фиксация изменений. Анализ истории. Ветвление и слияние. Ветви. Копирование изменений между ветвями.
18. Документирование. Работа с Docbook.
19. Тестирование. Тестирование программного обеспечения по спецификации и по коду.
20. Критерии качества тестирования.
21. Тесты и тестовые процедуры (определения, принципы создания).
22. Метод ручной инспекции кода; метод эквивалентов и граничных условий.
23. Тесты и тестовые процедуры (определения, принципы создания).
24. Критерии качества тестирования.
25. Утилита JUnit. Рефакторинг.
26. Понятие рефакторинга.
27. Качество программного обеспечения. Соглашения о кодировании. Факторы качества (по ГОСТ Р ИСО/МЭК 912693).
28. Метрики кода, их виды и реализация в Subversion.

Структура фонда оценочных средств для текущей и промежуточной аттестации

№	Контролируемые разделы (темы) дисциплины	компетенции	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Проект SWEBOK. Десять областей знания в программной инженерии.	ИУК-2.4 (06.001 D/03.06 Зн.1) ИУК-2.5 (06.001 D/03.06 Зн.20), ИУК-2.6 (06.001 D/03.06 Зн.3)	УО	ЭВ(1-3)
2	Последовательная, инкрементная, спиральная модели.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1)	УО	ЭВ(4)
3	Эволюционные модели. Прототипирование.	ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЭВ(4-5)
4	Анализ. Виды и источники требований.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1)	УО	ЭВ(6)

№	Контролируемые разделы (темы) дисциплины	компетенции	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
5	Спецификации требований к системе и ПО.	ИПК-1.6 (06.016 А/30.6 У.1) ИПК-1.7 (40.001 А/02.5 Тд.2) ИПК-2.1 (06.016 А/30.6 Зн.3)	УО	ЗВ(6)
6	Модульность. Структуры данных. Структурное программирование.	ИПК-1.6 (06.016 А/30.6 У.1) ИПК-1.7 (40.001 А/02.5 Тд.2) ИПК-2.1 (06.016 А/30.6 Зн.3)	ОУ	ЗВ(7)
7	Нисходящее и восходящее проектирование.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	ПДР	ЗВ(8)
8	Исключительные ситуации, их распространение и обработка.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2))	ПДР	ЗВ(9)
9	Особенности разработки объектных программ.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(10)
10	Шаблоны проектирования.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(10-11)
11	Управление файлами и каталогами во времени.	ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2)	УО	ЗВ(12-15)
12	Модели “блокирование – изменение – разблокирование” и ”копирование – изменение – слияние”. Рабочие копии. Коллективная работа.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(16)
13	Простой рабочий цикл. Фиксация изменений.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИОПК-5.3 (У.2)	УО	ЗВ(17)
14	Анализ истории. Ветвление и слияние.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(17)
15	Ветви. Копирование изменений между ветвями.	ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2)	ПДР	ЗВ(17)
16	Работа с Docbook.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2)	ПДР	ЗВ(18)
17	Преобразование DocBook-документа в форматы, доступные для печати. Понятие о DITA.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(18)
18	Тестирование программного обеспечения по спецификации и по коду.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	УО	ЗВ(19-24)
19	Утилита JUnit.	ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1) ИОПК-5.3 (У.2,3)	УО	ЗВ(25)
20	Понятие рефакторинга.	ИОПК-5.3 (У.2) ИОПК-5.4 (Тд.1) ИОПК-5.5 (Тд.2)	ПДР	ЗВ(26)
21	Соглашения о кодировании. Факторы качества.	ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1) ИУК-2.23 (06.001 D/03.06 Тд.2) ИУК-2.26 (06.016 А/30.6 Тд.1)	ПДР	ЗВ(27)
22	Качество программного обеспечения.	ИУК-2.14 (06.001 D/03.06 У.1) ИУК-2.15 (06.001 D/03.06 У.2) ИУК-2.16 (06.016 А/06.6 У.1)	ПДР	ЗВ(28)

Сокращения: УО – устный опрос, ПДР – проверка самостоятельной неаудиторной работы, ИД - индивидуальное задание, ЗВ – зачтённые вопросы.

Показатели, критерии и шкала оценки сформированных компетенций

Код и наименование компетенции	Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания		
	пороговый	базовый	продвинутый
	Оценка		
	зачтено	зачтено	зачтено
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений			
ИУК-2.4 (06.001 D/03.06 Зн.1) Виды архитектуры программного обеспечения и принципы построения архитектуры программного обеспечения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений ИУК-2.5 (06.001 D/03.06 Зн.2) Оптимальные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, исходя из имеющихся ресурсов и ограничений ИУК-2.6 (06.001 D/03.06 Зн.3) Способы определения круга задач, методы и средства проектирования программного обеспечения, выбирать оптимальные способы решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений ИУК-2.14 (06.001 D/03.06 У.1) Выбирать и использовать оптимальные существующие типовые решения и	Знать: состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач; методики проектирования ПО, некоторые типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ПО; Стандарты программной инженерии; модели жизненного цикла ПО; методики проектирования ПО; языки программирования; сетевые технологии; методы тестирования. Уметь: использовать некоторые существующие типовые решения и шаблоны проектирования системного и прикладного ПО Владеть: некоторыми навыками проектирования структур данных	Знать: состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач; технологии распределённой коллективной работы; языки программирования; языки баз данных, основные типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ПО; Стандарты программной инженерии; модели жизненного цикла ПО; методики проектирования ПО; языки программирования; сетевые технологии; методы тестирования. Уметь: составлять документацию ПО и использовать основные существующие типовые решения и шаблоны проектирования системного и прикладного ПО Владеть: навыками проектирования структур данных и программных интерфейсов	Знать: состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач; технологии распределённой коллективной работы; языки программирования; языки баз данных; сетевые технологии; методы тестирования; определение качества ПО типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке ПО; Стандарты программной инженерии; модели жизненного цикла ПО; методики проектирования ПО; языки программирования; сетевые технологии; методы тестирования. Уметь: составлять документацию ПО, использовать существующие типовые решения и шаблоны проектирования системного и прикладного ПО Владеть: устойчивыми

Код и наименование компетенции	Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания		
	пороговый	базовый	продвинутый
	Оценка		
	зачтено	зачтено	зачтено
шаблоны проектирования программного обеспечения ИУК-2.15 (06.001 D/03.06 У.2) Определять круг задач, применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов, исходя из имеющихся ресурсов и ограничений ИУК-2.16 (06.016 A/06.6 У.1) Способен определять круг задач и выбирать оптимальную структуру документации ПО ИУК-2.23 (06.001 D/03.06 Тд.2) Определение и проектирование структур данных в заданной предметной области ИУК-2.26 (06.016 A/30.6 Тд.1) Качественный анализ рисков в проектах в области ИТ, исходя из действующих правовых норм, имеющихся ресурсов и ограничений			навыками проектирования структур данных и программных интерфейсов
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения			
ИОПК-5.3 (У.2) Применять методы и средства проектирования системного и прикладного программного обеспечения, структур данных, баз данных, программных интерфейсов ИОПК-5.4 (Тд.1) Разработка математически сложных алгоритмов, изменение и согласование архитектуры программного	<i>Уметь:</i> анализировать предметную область, выделять основные бизнес-процессы; составлять техническое задание на разработку ПО; планировать ЖЦ ПО; собирать, обрабатывать и интерпретировать данные; работать с электронными библиотеками и пакетами программ для версионинга,	<i>Уметь:</i> анализировать предметную область, выделять основные бизнес-процессы; составлять техническое задание на разработку ПО; планировать ЖЦ ПО; собирать, обрабатывать и интерпретировать данные; работать с электронными библиотеками и пакетами программ для версионинга,	<i>Уметь:</i> анализировать предметную область, выделять основные бизнес-процессы; составлять техническое задание на разработку ПО; планировать ЖЦ ПО; собирать, обрабатывать и интерпретировать данные; работать с электронными библиотеками и пакетами программ для версионинга,

Код и наименование компетенции	Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания		
	пороговый	базовый	продвинутый
	Оценка		
	зачтено	зачтено	зачтено
обеспечения системным аналитиком архитектором программного обеспечения ИОПК-5.5 (Тд.2) Проектирование структур данных при разработке системного прикладного программного обеспечения	с тестирования и групповой работы <i>Владеть:</i> навыками составления требований к ПО; устойчивыми навыками тестирования; навыками работы с версиями документов <i>Обучающийся показывает</i> не достаточный уровень знаний учебного и лекционного материала, не в полном объеме владеет практическими навыками. В ответе не всегда присутствует логика, аргументы привлекаются недостаточно веские. На поставленные вопросы затрудняется с ответами, показывает недостаточно глубокие знания.	и тестирования и групповой работы <i>Владеть:</i> навыками составления требований к ПО; технологиями распределённой коллективной работы; устойчивыми навыками тестирования; навыками работы с версиями документов; <i>Обучающийся показывает</i> достаточный уровень профессиональных знаний, но допускает некоторые неточности и погрешности. Ответ построен достаточно логично, грамотно используются физические термины, но в ответе присутствуют незначительные ошибки. Вопросы, задаваемые преподавателем, не вызывают существенных затруднений.	и тестирования и групповой работы <i>Владеть:</i> навыками составления требований к ПО; технологиями распределённой коллективной работы; устойчивыми навыками тестирования; навыками работы с версиями документов; <i>Обучающийся показывает</i> не только высокий уровень теоретических знаний по дисциплине, свободно оперирует понятиями, умеет анализировать сложные процессы, увязывать знания, полученные при изучении различных дисциплин. Ответ, построен логично, материал излагается четко, ясно, аргументировано. На вопросы отвечает уверенно, по существу.

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации к сдаче зачета

Зачет является заключительным этапом процесса формирования компетенции студента при изучении дисциплины или ее части и имеет целью проверку и оценку знаний студентов по теории и применению полученных знаний, умений и навыков при решении практических задач. Зачеты проводятся по расписанию, в сроки, предусмотренные календарным графиком учебного процесса..

Зачет проводится только при предъявлении студентом зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой по изучаемой дисциплине. Студент должен ответить на вопросы экзаменационного билета и продемонстрировать выполненное задание. Результаты экзамена оцениваются по системе: «зачтено» или «незачтено».

Критерии выставления оценок

Оценка «зачтено»:

- систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы;
- точное использование научной терминологии систематически грамотное и логически правильное изложение ответа на вопросы;
- безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке и решении научных и практических задач;
- выраженная способность самостоятельно и творчески решать сложные проблемы и нестандартные ситуации;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой по дисциплине;
- умение ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин;
- творческая самостоятельная работа на практических занятиях, активное участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;
- высокий уровень сформированности заявленных в рабочей программе компетенций.

Оценка «зчтено»:

- достаточно полные и систематизированные знания по дисциплине;
- умение ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им критическую оценку;
- использование научной терминологии, лингвистически и логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- владение инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач;
- усвоение основной литературы, рекомендованной учебной программой по дисциплине;
- самостоятельная работа на практических занятиях, участие в групповых обсуждениях, средний уровень культуры исполнения заданий;
- средний уровень сформированности заявленных в рабочей программе компетенций.

Оценка «зачтено»:

- достаточный минимальный объем знаний по дисциплине;
- усвоение основной литературы, рекомендованной учебной программой;
- умение ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку;
- использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок;
- владение инструментарием учебной дисциплины, умение его использовать в решении типовых задач;
- умение под руководством преподавателя решать стандартные задачи;
- работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий;
- достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций.

Оценка «незачтено»:

- фрагментарные знания по дисциплине;
- отказ от ответа;
- знание отдельных источников, рекомендованных учебной программой по дисциплине;
- неумение использовать научную терминологию;

- наличие грубых ошибок;
- низкий уровень культуры исполнения заданий;
- низкий уровень сформированности заявленных в рабочей программе компетенций.

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1 Учебная литература

Основная литература:

1. Антамошкин О.А. Программная инженерия. Теория и практика. Красноярск: Сибирский федеральный университет, 2012. 247 с. [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=363975>.
2. Глас Р. Программирование и конфликты 2.0: теория и практика программной инженерии. СПб.; М.: Символ-Плюс, 2010. 239 с.
3. Кручинин В.В. Технологии программирования. Томск: ТУСУР, 2013. 272 с. [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=480536>.
4. Смирнов А.А. Технологии программирования. М.: Евразийский открытый институт, 2011. 192 с. [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=90777>.

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах.

Дополнительная литература:

1. Иванова, Г.С. Технология программирования. М.: КНОРУС, 2011. 333 с.
2. Кулямин В.В. Технологии программирования. Компонентный подход. М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2010. 463 с.
3. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: Учебное пособие – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2008. 200с.
4. Терехов А.Н. Технология программирования. М: М.: Интернет-Университет Информационных Технологий, 2007. 149 с. [Электронный ресурс]. - Режим доступа: http://biblioclub.ru/index.php?page=book_red&id=233491&sr=1.
5. Технология программирования / Ю.Ю. Громов, О.Г. Иванова, М.П. Беляев, Ю.В. Минин. - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2013. 173 с. [Электронный ресурс]. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=277802>.

5.2. Периодическая литература

Не используются

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» www.biblioclub.ru
3. ЭБС «BOOK.ru» <https://www.book.ru>

4. ЭБС «ZNANIUM.COM» www.znanium.com

5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных:

1. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>

2. Springer Materials <http://materials.springer.com/>

3. zbMath <https://zbmath.org/>

Ресурсы свободного доступа:

1. Мир математических уравнений EqWorld. <http://eqworld.ipmnet.ru/ru/library.htm>

2. Физика, химия, математика. <http://www.ph4s.ru/index.html>

3. Journal of Mathematical Physics. Online ISSN 1089-7658. <http://jmp.aip.org>

4. Словари и энциклопедии <http://dic.academic.ru/>;

Собственные электронные образовательные и информационные ресурсы КубГУ:

1. Среда модульного динамического обучения <http://moodle.kubsu.ru>

2. База учебных планов, учебно-методических комплексов, публикаций и конференций
<http://mschool.kubsu.ru/>

Перечень необходимого программного обеспечения

1. XML редакторы.
2. Программное обеспечение для организации Wi-Fi сети
3. Инструментальное средство DocBook.
4. Инструментальное средство Subversion.
5. Инструментальное средство Tortoise.
6. Утилита JUnit.

6. Методические указания для обучающихся по освоению дисциплины

В связи с отсутствием лекций вводный инструктаж в начале каждой лабораторной работы проводится с использованием набора презентаций, которые позволяют быстро ввести студентов в проблематику изучаемого материала. С этой же целью используются задания на самостоятельную работу.

Вводный инструктаж подкрепляться примерами, решаемыми преподавателем при участии студентов с использованием доски и программного обеспечения. Это, с одной стороны, позволяет подключить к восприятию материала естественный для человека способ рассуждений на примерах, с другой подтверждает в глазах студентов квалификацию преподавателя как специалиста по изучаемой дисциплине.

Следует уделять достаточное внимание нормативным материалам, которые плохо усваиваются нашими студентами, и сформировать мотивацию на необходимость создания спецификаций, планов тестирования и т.п.

Лабораторные занятия проводятся в компьютерных классах, при этом практикуется разделение студентов на малые группы.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и лабораторных занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Перечень вопросов для самоподготовки

1. Что такое программа, программная система, программный продукт, программное обеспечение.
2. Причины сложности разработки программного обеспечения.
3. Что такое программная инженерия.

4. Процессы жизненного цикла программного продукта (в т.ч. по стандарту ISO/IEC 12207 (ГОСТ Р ИСО/МЭК 12207)).
5. Процессы разработки программного продукта.
6. Модели и методологии разработки ПО.
7. Конфигурации и управление ими.
8. Модель зрелости возможностей СММ.
9. Задачи анализа требований. Составление технического задания.
10. Виды прототипирования.
11. Варианты использования: определение, роль в жизненном цикле, UML-диаграмма, текстовые спецификации.
12. Архитектурное и детальное проектирование.
13. Декомпозиция системы. Основные структурные методы проектирования (по направлению декомпозиции) – метод последовательно детализации (сверху вниз) и сборочный метод (снизу вверх).
14. Понятие модуля. Критерии качества проектирования модулей и классов.
15. Особенности проектирования интерфейса пользователя.
16. Безопасное программирование.
17. Что такое шаблоны проектирования. Виды шаблонов.
18. Документирование. Работа с Docbook.
19. Управление версиями. Модели “блокирование – изменение – разблокирование” и “копирование-изменение-слияние”. Рабочие копии. Коллективная работа.
20. Цели тестирования и отладки. Объекты и особенности процесса тестирования.
21. Виды тестирования.
22. Критерии качества тестирования.
23. Метод ручной инспекции кода; метод эквивалентов и граничных условий.
24. Тесты и тестовые процедуры (определения, принципы создания).
25. Классификация ошибок с точки зрения процесса разработки.
26. Основные программные и эксплуатационные документы
27. Общее и детальное планирование тестирования.
28. Метрики для оценки свойств программного продукта.
29. Основные факторы качества программного продукта (по ГОСТ Р ИСО/МЭК 912693).

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Материально-техническое обеспечение по дисциплине

По всем видам учебной деятельности в рамках дисциплины используются аудитории, кабинеты и лаборатории, оснащенные оборудованием.

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащённость
1.	Лабораторные занятия	Компьютерный класс, укомплектованный компьютерами с лицензионным программным обеспечением, необходимой мебелью (доска, столы, стулья). (Аудитории: 101, 102, 106, 106а, 105/1, 107(2), 107(3), 107(5), А301).
2.	Групповые (индивидуальные) консультации	Аудитория для семинарских занятий, укомплектованная компьютерами с лицензионным программным обеспечением и выходом в интернет, необходимой мебелью (доска, столы, стулья) и демонстрационным оборудованием. (Аудитории: 129, 131).
3.	Текущий контроль, промежуточная аттестация	Аудитория для семинарских занятий, текущего контроля и промежуточной аттестации, укомплектованная необходимой мебелью (доска, столы, стулья) (аудитории: 129, 131, 133, А305, А307, 147, 148, 149, 150, 100С, А301б, А512), компьютерами с лицензионным программным обеспечением и выходом в интернет (106, 106а, А301)
4.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения, обеспеченный доступом в электронную информационно-образовательную среду университета, необходимой мебелью (столы, стулья). (Аудитории: читальный зал, 102а).

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.