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.
«__» 05 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.36 ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Направление подготовки **01.03.02 Прикладная математика и информатика**

Направленность (профиль) Математическое моделирование в естествознании и технологиях

Программа подготовки академическая

Форма обучения очная

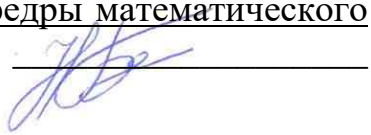
Квалификация (степень) выпускника бакалавр

Краснодар 2019

Рабочая программа дисциплины «ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ» составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению **01.03.02 Прикладная математика и информатика**, утвержденным приказом Министерства науки и высшего образования Российской Федерации № 9 от 10 января 2018 г.

Программу составил:

Бессарабов Н.В., канд. техн. наук, доцент, доцент кафедры математического моделирования КубГУ



Рабочая программа дисциплины «Технологии программирования» утверждена на заседании кафедры математического моделирования
протокол № 11 «10 » апреля 2019 г.

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



Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 1 « 15» мая 2019 г.

Председатель УМК факультета
канд. экон. наук, доцент Коваленко А.В.



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 ПК-4 (Способен активно участвовать в разработке системного и прикладного программного обеспечения) и ПК-5 (Способен применять основные алгоритмические и программные решения в области информационно-коммуникационных технологий, а также участвовать в их разработке).

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:

– Программирование на Java
– БД и СУБД
– Oracle
– XML

Для полноценного освоения курса студент должен в достаточно большом объёме 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

Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
		знать	**уметь**	**владеть**
ПК-4	Способен активно участвовать в разработке системного и прикладного программного обеспечения	– состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач. – методики проектирования ПО; – технологии распределённой коллективной работы; – языки программирования; – языки баз данных; – сетевые технологии; – методы тестирования; – определение качества ПО.	– анализировать предметную область, выделять основные бизнес-процессы; – составлять техническое задание на разработку ПО; – составлять документацию ПО.	– навыками составления требований к ПО; – технологиями распределённой коллективной работы; – методами и средствами разработки и оформления технической документации – устойчивыми навыками тестирования.
ПК-5	Способен применять основные алгоритмические и программные решения в области информационно-коммуникационных технологий, а также участвовать в их разработке	– стандарты программной инженерии; – модели жизненного цикла ПО; – принципы планирование ЖЦ ПО;	– ставить задачу и разрабатывать алгоритм ее решения, разрабатывать основные программные документы; – проводить формализацию решения прикладных задач; – планировать ЖЦ ПО; – собирать, обрабатывать и интерпретировать данные; – работать с электронными библиотеками и пакетами программ для версионинга, тестирования и групповой работы; – работать с современными системами программирования, включая объектно-ориентированные – составлять документацию ПО.	– методами и средствами описания прикладных процессов и информационного обеспечения решения прикладных задач; – методами и средствами разработки и оформления технической документации. – устойчивыми навыками тестирования; – навыками работы с версиями документов.

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

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 академических часов 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 экзамен. Программой дисциплины предусмотрены 14 часов лекционных занятий, 28 часов лабораторных занятий, 48 часов самостоятельной работы и подготовка к экзамену – 53,7 часа.

Вид учебной работы		Всего часов	Семестр (часы)
			7
Контактная работа (всего)		**42,3**	**42,3**
В том числе:			
Занятия лекционного типа		14	14
Занятия семинарского типа (семинары, практические занятия)		–	–
Лабораторные занятия		28	28
Иная контактная работа:			
Контроль самостоятельной работы (КСР)		–	–
Промежуточная аттестация (ИКР)		0,3	0,3
Самостоятельная работа (всего)		**48**	**48**
В том числе:			
Курсовая работа		–	–
Проработка учебного (теоретического) материала		28	28
Подготовка к текущему контролю		20	20
Контроль: экзамен			
Подготовка к экзамену		53,7	53,7
Общая трудоемкость	**час.**	**144**	**144**
	в том числе контактная работа	**42,3**	**42,3**
	зач. ед	**4**	**4**

2.2 Структура дисциплины:

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

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа		Внеаудиторная работа	
			Л	ЛЗ	СРС	контроль
1	Программная инженерия.	7	1	–	2	4
2	Модели жизненного цикла ПО	7	1	1	2	3
3	Составление требований к программному обеспечению.	11	1	2	4	4

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа		Внеаудиторная работа	
			Л	ЛЗ	СРС	контроль
4	Проектирование и программирование	15	1	4	4	6
5	Шаблоны проектирования	18	2	4	6	6
6	Управление версиями. Модель.	11	1	2	4	4
7	Управление версиями. Subversion. Tortoise.	11	1	2	4	4
8	Документирование.	11	1	2	4	4
9	Тестирование.	16	1	3	6	6
10	Утилита JUnit. Рефакторинг.	11	1	2	4	4
11	Качество программного обеспечения.	11	1	2	4	4
12	Программная инженерия.	14,7	2	4	4	4,7
Промежуточная аттестация (ИКР)		0,3	–	–	–	–
Итого:		**144**	**14**	**28**	**48**	**53,7**

2.3 Содержание разделов дисциплины:

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Программная инженерия.	Проект SWEBOK. 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 SWEBOK. 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. 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 связи с отсутствием лекционной части курса при проведении лабораторных работ используется набор презентаций, которые позволяют быстро ввести студентов в проблематику изучаемого материала. С этой же целью используются задания на самостоятельную работу.

Вводная информационно-объяснительная часть лабораторной работы должна подкрепляться примерами, в том числе, с использованием доски (лучше интерактивной) 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.

Для достаточно хорошо подготовленных студентов можно предлагать проблемы для решения методом «мозговых атак», но при низком уровне такой подход совершенно не эффективен.

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.

Групповые индивидуальные задания формируют навыки исследовательской работы в коллективе.

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

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

4. 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.

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Основные требования к результатам освоения дисциплины представлены в таблице в виде признаков сформированности компетенций. Требования формулируются в соответствии со структурой, принятой в ФГОС ВО: знать, уметь, владеть.

Название компетенции (или ее части)	Структура компетенции	Основные признаки сформированности компетенции
ПК-4 Способен активно участвовать в разработке системного и прикладного программного обеспечения	Знать: – состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач. – методики проектирования ПО; – технологии распределённой коллективной работы; – языки программирования; – языки баз данных; – сетевые технологии; – методы тестирования; определение качества ПО.	Знает состав и структуру прикладных процессов, компоненты информационного обеспечения решения прикладных задач, методики проектирования ПО, технологии распределённой коллективной работы; языки программирования; языки баз данных; сетевые технологии; методы тестирования; определение качества ПО.
	Уметь: – анализировать предметную область, выделять основные бизнес-процессы; – составлять техническое задание на разработку ПО; составлять документацию ПО.	Умеет анализировать предметную область, выделять основные бизнес-процессы; составлять техническое задание на разработку ПО; составлять документацию ПО.
	Владеть: – навыками составления требований к ПО; – технологиями распределённой коллективной работы; – методами и средствами разработки и оформления технической документации устойчивыми навыками тестирования.	Владеет навыками составления требований к ПО; технологиями распределённой коллективной работы; методами и средствами разработки и оформления технической документации, устойчивыми навыками тестирования.
ПК-5 Способен применять основные алгоритмические и программные решения в области	Знать: – стандарты программной инженерии; – модели жизненного цикла ПО; принципы планирование ЖЦ ПО;	Знает стандарты программной инженерии; модели жизненного цикла ПО; принципы планирование ЖЦ ПО;

Название компетенции (или ее части)	Структура компетенции	Основные признаки сформированности компетенции
информационно-коммуникационных технологий, а также участвовать в их разработке	Уметь: – ставить задачу и разрабатывать алгоритм ее решения, разрабатывать основные программные документы; – проводить формализацию решения прикладных задач; – планировать ЖЦ ПО; – собирать, обрабатывать и интерпретировать данные; – работать с электронными библиотеками и пакетами программ для версионинга, тестирования и групповой работы; – работать с современными системами программирования, включая объектно-ориентированные составлять документацию ПО.	Умеет ставить задачу и разрабатывать алгоритм ее решения, разрабатывать основные программные документы; проводить формализацию решения прикладных задач; планировать ЖЦ ПО; собирать, обрабатывать и интерпретировать данные; работать с электронными библиотеками и пакетами программ для версионинга, тестирования и групповой работы; работать с современными системами программирования, включая объектно-ориентированные составлять документацию ПО.
	Владеть: – методами и средствами описания прикладных процессов и информационного обеспечения решения прикладных задач; – методами и средствами разработки и оформления технической документации. – устойчивыми навыками тестирования; навыками работы с версиями документов.	Владеет методами и средствами описания прикладных процессов и информационного обеспечения решения прикладных задач; методами и средствами разработки и оформления технической документации, устойчивыми навыками тестирования; навыками работы с версиями документов.

4.1 Фонд оценочных средств для проведения текущего контроля

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 экзамен

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 CMM.
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.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации 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.

Методические рекомендации к сдаче экзамена

Экзамен 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. Экзамены проводятся по расписанию, в сроки, предусмотренные календарным графиком учебного процесса. Расписание экзаменов доводится до сведения студентов не менее чем за две недели до начала экзаменационной сессии.

Экзамен проводится только при предъявлении студентом зачетной книжки и при условии выполнения всех контрольных мероприятий, предусмотренных учебным планом и рабочей программой по изучаемой дисциплине. Студентам на экзамене предоставляется право выбрать один из билетов. Время подготовки к ответу составляет 60 минут. По истечении установленного времени студент должен ответить на вопросы экзаменационного билета и продемонстрировать выполненное задание. Результаты экзамена оцениваются по четырехбалльной системе («отлично», «хорошо», «удовлетворительно», «неудовлетворительно») и заносятся в экзаменационную ведомость и зачетную книжку. В зачетную книжку заносятся только положительные оценки.

Критерии выставления оценок

Оценка «*отлично*»:

– 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 задач;

– умение ориентироваться в теориях, концепциях и направлениях дисциплины;

–творческая самостоятельная работа на лабораторных занятиях, активное участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;

– высокий уровень сформированности заявленных в рабочей программе компетенций.

Оценка «*хорошо*»:

– достаточно полные и систематизированные знания по дисциплине;

– умение ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им оценку;

– использование научной терминологии, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;

– владение инструментарием по дисциплине;

– самостоятельная работа на лабораторных занятиях, , средний уровень культуры исполнения заданий;

– средний уровень сформированности заявленных в рабочей программе компетенций.

Оценка «*удовлетворительно*»:

– 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 лабораторных занятиях, допустимый уровень культуры исполнения заданий;

– достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций.

Оценка «*неудовлетворительно*»:

– 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 основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1 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 «Лань»,

5.2 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.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. http://swebok.sorlik.ru
2. http://svnhowto.com/
3. http://svnbook.red-bean.com/index.ru.html
4. http://docbook.ru/

7. 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 CMM.
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.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень информационных технологий

– Проверка индивидуальных заданий и консультирование посредством электронной почты.

– Использование электронных презентаций при проведении лекционных и лабораторных занятий.

8.2 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.

8.3 Перечень информационных справочных систем:

1. Электронная библиотечная система "Юрайт" (http://www.biblio-online.ru).
2. Электронная библиотечная система "Университетская библиотека ONLINE" (http://www.biblioclub.ru).
3. Электронная библиотечная система издательства "Лань" (http://e.lanbook.com).
4. Электронная библиотечная система eLIBRARY.RU (http://www.elibrary.ru).

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащенность
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)

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащенность
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).

Осуществление учебного процесса курса предполагает наличие необходимого для реализации данной программы перечня материально-технического обеспечения: аудитории, оборудованные видеопроекционным оборудованием для презентаций (цифровой проектор, экран, ноутбук); компьютерные классы для проведения лабораторных занятий.

Факультет компьютерных технологий и прикладной математики, оснащен компьютерными классами, установлена локальная сеть, все компьютеры факультета подключены к сети Интернет, имеется лицензионное программное обеспечение. Студенты и преподаватели вуза имеют постоянный доступ к электронному каталогу учебной, методической, научной литературе, периодическим изданиям и архиву статей.