

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Рабочая программа дисциплины (модуля)  
**ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ**

Направление и направленность (профиль)  
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2024

Форма обучения  
очная

Владивосток 2024

Рабочая программа дисциплины (модуля) «Технология программирования» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия (утв. приказом Минобрнауки России от 19.09.2017г. №920) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

*Васильев Б.К., кандидат химических наук, доцент, Кафедра информационных технологий и систем, boris.vasiliev@vvsu.ru*

Утверждена на заседании кафедры информационных технологий и систем от 29.05.2024 , протокол № 9

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кийкова Е.В.

<b>ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ</b>	
Сертификат	1575633692
Номер транзакции	000000000D0AC67
Владелец	Кийкова Е.В.

## 1 Цель, планируемые результаты обучения по дисциплине (модулю)

Целью освоения дисциплины «Технология программирования» является теоретическая и практическая подготовка студентов в области технологий разработки программ в такой степени, чтобы при менеджменте программного проекта или в процессе участия в его реализации они могли выбирать необходимые технические, алгоритмические, программные и технологические решения, уметь объяснить принципы их функционирования и правильно их использовать. Иметь представление о каждом этапе жизненного цикла программы от проектирования до внедрения и сопровождения. Знать современные стандарты качества программного обеспечения и перспективные направления развития технологии разработки ПО.

Задачи освоения дисциплины состоят: в освоении основных положений технологии разработки ПО, формулировка практических рекомендаций по организации работы коллективов программистов, руководства такими коллективами, формировании у студентов знаний по дисциплине, связанных с процессом разработки ПО, включая связи с предметной областью, реализацию, организацию производства, контроль сроков исполнения и качества, ознакомлении с техническими программными и технологическими решениями, используемыми при разработке ПО; а также в приобретении практических навыков работы в коллективе программистов, умения находить правильные технологические решения по выбору средств разработки и структуры программного проекта, методов тестирования и контроля исполнения использование современных инструментальных и методологических средств.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1к : Использует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	РД1	Знание	тестирования и обеспечения качества, организации работы команды разработчиков
			РД2	Умение	составлять тестовые наборы данных, декомпозицию программного проекта
			РД3	Знание	организации процесса разработки ПО (включая все стадии), организации верификации, тестирования и проверки стабильности ПО, управления качеством
			РД4	Навык	использования способов разработки отдельных модулей, их сборки и создания пользовательского интерфейса, выполнения интеграции проекта, тестирования и сопровождения

				РД5	Умение	самостоятельно изучить предметную область, составить техническое задание, выбрать необходимые математические модели и способы их алгоритмической реализации, выбрать программные и инструментальные средства для разработки ПО, организовать верификацию, тестирование и проверку стабильности ПО
--	--	--	--	-----	--------	---

## 2 Место дисциплины (модуля) в структуре ОПОП

Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана.

## 3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обучения	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудо-емкость (З.Е.)	Объем контактной работы (час)					СРС	Форма аттестации	
					Всего	Аудиторная			Внеаудиторная			
						лек.	прак.	лаб.	ПА			КСР
09.03.04 Программная инженерия	ОФО	Б1.Б	3	3	55	18	36	0	1	0	53	ДЗ

## 4 Структура и содержание дисциплины (модуля)

### 4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке	РД1, РД2, РД3, РД4, РД5	2	4	0	6	отчет о выполнении практической работы

2	Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	РД1, РД2, РД3, РД4, РД5	2	4	0	8	отчет о выполнении практической работы
3	Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	РД1, РД2, РД3, РД4, РД5	2	6	0	8	отчет о выполнении практической работы
4	Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматики языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	РД1, РД2, РД3, РД4, РД5	4	6	0	8	отчет о выполнении практической работы
5	Групповая разработка, управление версиями. Параллельная и конкурентная разработка. Различные способы организации коллектива разработчиков. Основные и вспомогательные подразделения на предприятии и их задачи	РД1, РД3	2	4	0	8	отчет о выполнении практической работы
6	Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	РД1, РД2, РД3, РД4, РД5	2	6	0	8	отчет о выполнении практической работы
7	Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	РД1, РД2, РД3, РД4, РД5	4	6	0	7	отчет о выполнении практической работы
<b>Итого по таблице</b>			<b>18</b>	<b>36</b>	<b>0</b>	<b>53</b>	

#### 4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

*Тема 1 Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке.*

Содержание темы: Понятие технологии программирования: Особенности промышленного программирования, "программирование для себя" (Just for fun) и "программирование на заказ". Языки программирования как средство выражения алгоритмов и как средство получения выполняемого кода. Распределение памяти в языке Си. Жизненный цикл программного обеспечения (ПО). Общая организация проекта. Модели разработки ПО. Основные технологические подходы: каскадный, каркасный, сборочный,

адаптивный (экстремальное программирование). Трудности проектирование, их разновидности и преодоление. Примеры. Понятие АД. АД, ориентированные на предметную область (пример из геометрии). Понятие технологии программирования: Особенности промышленного программирования, "программирование для себя" (Just for fun) и "программирование на заказ". Языки программирования как средство выражения алгоритмов и как средство получения выполняемого кода. Распределение памяти в языке Си. Жизненный цикл программного обеспечения (ПО). Общая организация проекта. Модели разработки ПО. Основные технологические подходы: каскадный, каркасный, сборочный, адаптивный (экстремальное программирование). Трудности проектирование, их разновидности и преодоление. Примеры. Понятие АД. АД, ориентированные на предметную область (пример из геометрии).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

*Тема 2 Значение предметной области. Различные модели процесса разработки ПО. АД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения.*

Содержание темы: Значение предметной области. Каскадная и спиралевидная модели процесса разработки ПО. Возникновение и исследование идеи. Иллюстрации на предыдущем примере АД, оценка осуществимости: Как оценить сложность задачи? Реальность ее решения в заданные сроки при заданных рамочных ограничениях. Планирование: Сетевой и ленточный графики, треугольник – сроки, работы, ресурсы. Анализ требований и выработка спецификаций ПО. Проектирование архитектуры продукта. Выбор средств реализации. Инструментарий разработчика. Технологии RAD и готовые компоненты. Библиотеки алгоритмов и подпрограмм.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

*Тема 3 Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных.*

Содержание темы: Тестирование, обеспечение качества: Критерии качества и их метрики. Стандарты ISO 9000, 9001. Стандартизация информационных технологий. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных. Примеры из конкретной предметной области. Групповая разработка (постановка задачи).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

*Тема 4 Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматики языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений.*

Содержание темы: Средства автоматизации решения традиционно считающихся трудными задач. Разработка синтаксических анализаторов. Понятия грамматики языка, лексического и синтаксического разбора. Способы описания грамматики. Нотация БНФ, генератор распознавателей yacc, bison. Лингвистический подход при разработке приложений. Примеры малых языков и преимущества их использования.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

*Тема 5 Групповая разработка, управление версиями. Параллельная и конкурентная разработка. Различные способы организации коллектива разработчиков. Основные и вспомогательные подразделения на предприятии и их задачи.*

Содержание темы: Групповая разработка, управление версиями. Единый репозиторий проекта. Системы RCS, CVS. Методы нумерации версий. Параллельная и конкурентная разработка. Организация коллектива разработчиков: Матричный метод, метод главного специалиста, вертикальные и горизонтальные координации управления проектом. Основные и вспомогательные подразделения и их задачи.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к текущему и промежуточному тестированию.

*Тема 6 Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.*

Содержание темы: Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности. Требования, предъявляемые к ПО и документации для реализации успешного сопровождения.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

*Тема 7 Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем.*

Содержание темы: Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем: Перевод устаревших программ на новые языки и платформы, возвратное проектирование – извлечение знаний из текста программы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическая работа.

Виды самостоятельной подготовки студентов по теме: подготовка к промежуточному тестированию, практическим работам.

## **5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)**

### **5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы**

В ходе изучения дисциплины «Технология программирования» студенты могут посещать аудиторные занятия (лекции, практические занятия, консультации). Особенность изучения дисциплины «Технология программирования» состоит в том, что студенты в ходе выполнения практических работ разбирают и анализируют способы проектирования, изготовления, интеграции и тестирования многомодульного ПО.

Особое место в овладении частью тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а второстепенные и более легкие вопросы, а также вопросы, специфичные для направления подготовки, могут быть изучены студентами самостоятельно.

В соответствии с учебным планом направления подготовки процесс изучения дисциплины может предусматривать проведение лекций, практических занятий, консультаций, а также самостоятельную работу студентов. Обязательным является проведение практических занятий в специализированных компьютерных аудиториях, оснащенных персональными компьютерами с установленными операционными системами и средствами разработки приложений различного назначения.

Ниже перечислены предназначенные для самостоятельного изучения студентами те вопросы из лекционных тем, которые во время проведения аудиторных занятий изучаются недостаточно или изучение которых носит обзорный характер.

В рамках общего объема часов, отведенных для изучения дисциплины, предусматривается выполнение следующих видов самостоятельных работ студентов (СРС): изучение теоретического материала при подготовке к защите практических работ, итоговое повторение теоретического материала. Для самостоятельного изучения дисциплины выносятся часть материала по всем темам дисциплины с самоконтролем по контрольным вопросам и возможностью консультации у ведущего преподавателя. Для закрепления материала и приобретения навыков расчета рекомендуется выполнение следующих задач:

- Разработка многомодульного программного проекта «калькулятор с возможностью построения графиков произвольных функций одной и двух переменных»
- Ознакомление с методами автоматической генерации синтаксических распознавателей (bison, Zubr, eli). Анализ грамматики известных языков программирования (C, Pascal, yacc, prolog).
- Установка средств разработки программного обеспечения open source (компилятор gcc, отладчик), используя пакет исходных текстов («тарбол»).
- Разработка плана работ и сетевого графика для собственного программного проекта.

## **5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

## **6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)**

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания,



методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

## **7 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **7.1 Основная литература**

1. Лаврищева Е. М. ПРОГРАММНАЯ ИНЖЕНЕРИЯ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ СЛОЖНЫХ СИСТЕМ 2-е изд., испр. и доп. Учебник для вузов [Электронный ресурс] , 2020 - 432 - Режим доступа: <https://urait.ru/book/programmnaia-inzheneriya-i-tehnologii-programmirovaniya-slozhnyh-sistem-452137>

2. Романов, Е. Л. Программная инженерия : учебное пособие / Е. Л. Романов. — Новосибирск : НГТУ, 2017. — 395 с. — ISBN 978-5-7782-3455-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118221> (дата обращения: 30.09.2024). — Режим доступа: для авториз. пользователей.

### **7.2 Дополнительная литература**

1. Мишова, В. В. Технологии программирования: практикум : для студентов, обучающихся по направлению подготовки 51.03.06 Библиотечно-информационная деятельность, профиль Технология автоматизированных библиотечно-информационных систем, квалификация (степень) выпускника бакалавр, формы обучения: очная, заочная / В. В. Мишова. — Кемерово : Издательство КемГИК, 2016. — 87 с. : ил. — Библиогр.: с.84-85. — ISBN 978-5-8154-0360-4. — URL: <https://lib.rucont.ru/efd/614334> (дата обращения: 30.09.2024)

2. Основы алгоритмизации и программирования : учебное пособие / составители А. А. Прокин, В. И. Харитонов. — Саранск : МГУ им. Н.П. Огарева, 2023. — 164 с. — ISBN 978-5-7103-4619-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/397916> (дата обращения: 30.09.2024). — Режим доступа: для авториз. пользователей.

3. Родионова, Т. Е. Технологии программирования : учебное пособие / Т. Е. Родионова. — Ульяновск : УлГТУ, 2018. — 115 с. — ISBN 978-5-9795-1789-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/165070> (дата обращения: 30.09.2024). — Режим доступа: для авториз. пользователей.

4. Сборник задач по программированию / Э.С. Бадмаева, О.А. Лобсанова. — Улан-Удэ : Бурятский государственный университет, 2019. — 96 с. — ISBN 978-5-9793-1426-6. — URL: <https://lib.rucont.ru/efd/706661> (дата обращения: 30.09.2024)

5. Технологии программирования : учебное пособие / А. В. Гайдель, А. В. Благов, В. И. Проценко, А. С. Широкаев. — Самара : Самарский университет, 2020. — 108 с. — ISBN 978-5-7883-1554-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/189025> (дата обращения: 30.09.2024). — Режим доступа: для авториз. пользователей.

6. Языки программирования. Часть 1 : лабораторный практикум. Направление подготовки 10.03.01 – Информационная безопасность. Бакалавриат / Е. А. Малиновская, Р. А. Рыскаленко. — Ставрополь : изд-во СКФУ, 2016. — 103 с. — URL: <https://lib.rucont.ru/efd/622893> (дата обращения: 30.09.2024)

### **7.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):**

1. <https://habrahabr.ru/flows/develop/> - Профессиональный ресурс: раздел посвященный разработке ПО.
2. Электронно-библиотечная система "ЛАНЬ"
3. Электронно-библиотечная система "РУКОНТ"
4. Электронно-библиотечная система издательства "Юрайт" - Режим доступа: <https://urait.ru/>
5. Open Academic Journals Index (ОАИ). Профессиональная база данных - Режим доступа: <http://oaji.net/>
6. Президентская библиотека им. Б.Н.Ельцина (база данных различных профессиональных областей) - Режим доступа: <https://www.prlib.ru/>
7. Информационно-справочная система "Консультант Плюс" - Режим доступа: <http://www.consultant.ru/>

**8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения**

Основное оборудование:

- Компьютеры
- Проектор
- Графическая станция Desten eXtreme 1024Q Монитор Acer P193 WAW
- Графическая станция Desten eXtreme 1024Q Монитор Acer P193 WAW
- Компьютер WSU5/270.64/4GB(сист
- Мультимедийный комплект №1 (Проектор Sanyo PLC-XD2600 потол. крепл.SMS CL F500, к/м Kramer WX-1N,коннектор VGA,экран Draper Star 178\*178, зап. лампа
- Система аудиовизуального представления информации
- Экран Projecta 160\*160

Программное обеспечение:

- Microsoft Office Professional Plus 2013 Russian
- VMware Workstation 9 for Linux andWindows

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Фонд оценочных средств  
для проведения текущего контроля  
и промежуточной аттестации по дисциплине (модулю)

**ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ**

Направление и направленность (профиль)

09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2024

Форма обучения  
очная

Владивосток 2024

## 1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1к : Использует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

## 2 Показатели оценивания планируемых результатов обучения

**Компетенция ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»**

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код результата	Тип результата	Результат	
ОПК-6.1к : Использует навыки и программирования, отладки и тестирования прототипов программно-технических комплексов задач	РД1	Знание	тестирования и обеспечения качества, организации работы команды разработчиков	знание принципов тестирования и обеспечения качества, особенностей организации работы команды разработчиков
	РД2	Умение	составлять тестовые наборы данных, декомпозицию программного проекта	составление тестовых наборов в данных, декомпозиции программного проекта
	РД3	Знание	организации процесса разработки ПО (включая все стадии), организации верификации, тестирования и проверки стабильности ПО, управления качеством	знание организации процесса разработки ПО, организации верификации, тестирования и проверки стабильности ПО, управления качеством
	РД4	Навык	использования способов разработки отдельных модулей, их сборки и создания пользовательского интерфейса, выполнения интеграции проекта, тестирования и сопровождения	использование способов разработки отдельных модулей, их сборки и создания пользовательского интерфейса, выполнения интеграции, тестирования и сопровождения проекта

	Р Д 5	У м е н е	самостоятельно изучить предметную область, составить техническое задание, выбрать необходимые математические модели и способы их алгоритмической реализации, выбрать программные и инструментальные средства для разработки ПО, организовать верификацию, тестирование и проверку стабильности ПО	самостоятельное изучение предметной области, составление технического задания, выбор необходимых математических моделей и способов их алгоритмической реализации, выбор программных и инструментальных средств для разработки ПО, организация верификации, тестирования и проверки стабильности ПО
--	-------------	-----------------------	---	--

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

### 3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения	Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС		
		Текущий контроль	Промежуточная аттестация	
Очная форма обучения				
РД1	Знание : тестирования и обеспечения качества, организации работы команды разработчиков	1.1. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке	Практическая работа	Тест
		1.2. Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	Практическая работа	Тест
		1.3. Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	Практическая работа	Тест
		1.4. Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматик и языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	Практическая работа	Тест

		1.5. Групповая разработка, управление версиями. Параллельная и конкурентная разработка. Различные способы организации коллектива разработчиков. Основные и вспомогательные подразделения на предприятии и их задачи	Практическая работа	Тест
		1.6. Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	Практическая работа	Тест
		1.7. Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	Практическая работа	Тест
РД2	Умение : составлять тестовые наборы данных, декомпозицию программного проекта	1.1. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке	Практическая работа	Тест
		1.2. Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	Практическая работа	Тест
		1.3. Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	Практическая работа	Тест
		1.4. Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматик и языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	Практическая работа	Тест
		1.6. Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	Практическая работа	Тест

		1.7. Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	Практическая работа	Тест
РДЗ	Знание : организации процесса разработки ПО (включая все стадии), организации верификации, тестирования и проверки стабильности ПО, управления качеством	1.1. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке	Практическая работа	Тест
		1.2. Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	Практическая работа	Тест
		1.3. Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	Практическая работа	Тест
		1.4. Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматик и языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	Практическая работа	Тест
		1.5. Групповая разработка, управление версиями. Параллельная и конкурентная разработка. Различные способы организации коллектива разработчиков. Основные и вспомогательные подразделения на предприятии и их задачи	Практическая работа	Тест
		1.6. Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	Практическая работа	Тест

		1.7. Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	Практическая работа	Тест
РД4	Навык : использования с пособов разработки отдельных модулей, их сборки и создания пользовательского интерфейса, выполнения интеграции проекта, тестирования и сопровождения	1.1. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке	Практическая работа	Тест
		1.2. Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	Практическая работа	Тест
		1.3. Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	Практическая работа	Тест
		1.4. Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматик и языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	Практическая работа	Тест
		1.6. Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	Практическая работа	Тест
		1.7. Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	Практическая работа	Тест
		РД5	Умение : самостоятельно изучить предметную область, составить техническое задание, выбрать необходимые математические модели и способы их алгоритмической ре	1.1. Основные понятия технологии программирования, особенности программного проекта. Способы преодоления сложностей при разработке



	лизации, выбрать программные и инструментальные средства для разработки ПО, организовать верификацию, тестирование и проверку стабильности ПО	1.2. Значение предметной области. Различные модели процесса разработки ПО. АТД, ориентированные на предметную область, оценка осуществимости проекта, графики выполнения	Практическая работа	Тест
		1.3. Тестирование, обеспечение качества: Критерии качества и их метрики. Статическое и динамическое тестирование. Методы белого и черного ящиков. Создание тестовых наборов данных	Практическая работа	Тест
		1.4. Средства автоматизации при разработке синтаксических анализаторов. Понятия грамматик и языка, лексического и синтаксического разбора. генераторы распознавателей yacc, bison. Лингвистический подход при разработке приложений	Практическая работа	Тест
		1.6. Сопровождение: Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	Практическая работа	Тест
		1.7. Разработка интерфейса пользователя: решаемые задачи и средства. Целесообразность и метафоричность интерфейса. Виды интерфейсов. Средства для разработки интерфейсов. Реинжиниринг программных систем	Практическая работа	Тест

#### 4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Тестовые задания	Практические работы	Итого
Практические занятия		60	60
Самостоятельная работа		20	20
Промежуточная аттестация	20		20
Итого	20	80	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках

дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями и умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

## 5 Примерные оценочные средства

### 5.1 Примеры заданий для выполнения практических работ

Тема 1. Разработка приложения – игры в «крестики-нолики» на поле заданного размера, содержащего графический интерфейс и содержащего аналитическую часть для проверки завершающего хода.

Тема 2. Разработка приложения, считывающего структурированные данные из файла и заполняющего структуры данных.

Тема 3. Разработка приложения, демонстрирующего многомерный куб (4 или 5 измерений) на 2-мерных проекциях.

Тема 4. Реализовать программу, включающую графический интерфейс инженерного калькулятора с использованием специальных функций и «построителя графиков» функций одной и двух переменных по заданной формуле.

Тема 5. Реализовать программу, включающую язык манипуляций с матрицами и матричные операции.

Тема 6. Выполнить переработку графического интерфейса практических работ 1-4. Реализовать функции контроля времени.

#### *Краткие методические указания*

На выполнение одной практической работы отводится не менее двух двухчасовых занятий (включая затраты времени на проведение промежуточного теста на последнем в учебном периоде практическом занятии). После выполнения каждой практической работы студент должен представить отчет о ее выполнении, а также, по указаниям преподавателя, выполнить дополнительные задания по теме практической работы.

#### *Шкала оценки*

№	Баллы	Описание
5	73–80	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.

№	Баллы	Описание
4	61–72	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	49–60	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	33–48	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0–32	Студентом проявляется полное или практически полное отсутствие умений и навыков.

## 5.2 Примеры тестовых заданий

1. Среди этапов жизненного цикла ПО нет этапа

- 1) кодирования;
- 2) интеграции;
- 3) архивации;
- 4) сопровождения.

2. Критический путь на сетевом графике проекта можно найти

- 1) методом динамического программирования;
- 2) при помощи алгоритма Флойда;
- 3) методом линейного программирования;
- 4) перебором вариантов.

3. Ленточный график используется для

- 1) установления зависимостей между видами работ;
- 2) определения времени выполнения проекта;
- 3) наглядного представления хода работ;
- 4) оптимизации распределения ресурсов.

4. Наибольшей сложностью отличается модель процесса разработки ПО

- 1) спиралевидная;
- 2) модель RAD;
- 3) каскадная;
- 4) V-образная.

5. Минимальная временная сложность программы сортировки составляет

- 1)  $O(N)$ ;
- 2)  $O(N^2)$ ;
- 3)  $O(N^{1/2})$ ;
- 4)  $O(N \log_2 N)$ .

6. Временная сложность программы формирования частотного словаря текста

- 1)  $O(N)$ ;
- 2)  $O(N \log_2 N)$ ;
- 3)  $O(N^2)$ ;
- 4)  $O(\log_2 N)$ .

7. Сложность программы обычно выражается

- 1) количеством человеко-дней, затраченных на разработку;
- 2) количеством строк кода;
- 3) размером исполняемого модуля в байтах;
- 4) количеством операторов в программе.

8. Типовое значение числа Ингве

- 1) 5-9;
- 2) 2-4;
- 3) 7-11;
- 4) 9-14.

9. Использование настраиваемого представления интерфейса недопустимо в приложениях, предназначенных для

- 1) программа управления роботом-спасателем;
- 2) обработки данных;
- 3) офисных работ;
- 4) управления производством.

10. Для сопровождения индустриального ПО чаще всего используется

- 1) техническая спецификация ПО;
- 2) исходный текст программ;
- 3) специальное руководство по сопровождению;
- 4) руководство системного оператора.

11. Количество используемых элементов отображения изменяющихся параметров в интерфейсе не должно

- 1) влиять на работу оператора;
- 2) быть слишком маленьким;
- 3) изменяться в ходе эксплуатации;
- 4) превышать число Ингве.

12. Мобильный интерфейс нельзя создать при помощи

- 1) VisualC;
- 2) Delphi;
- 3) FLTK;
- 4) QTdesigner.

13. В библиотеке FLTK к элементам управления, не имеющим метода "value" является

- 1) slider;
- 2) roller;
- 3) counter;
- 4) browser.

14. Промышленное использование переноса программ, написанных на Delphi в ОС Linux

- 1) из-за неоптимальности кода используется редко;
- 2) сейчас используется повсеместно;
- 3) невозможно из-за несовместимости графических подсистем Windows и Linux;
- 4) не делается из-за отсутствия компилятора.

15. Для разработки графического пользовательского интерфейса наиболее всего подходит

методология

- 1) функционального программирования;
- 2) объектно-ориентированного программирования;
- 3) императивного программирования;
- 4) логического программирования.

16. Метод черного ящика не применяется чтобы подтвердить

- 1) правильное обнаружение ошибок и их обработку;
- 2) правильную работу продукта при больших потоках данных;
- 3) правильный выбор размера критических секций кода;
- 4) правильность вычислений.

17. Целью тестирования методом белого ящика не является

- 1) отслеживание всех ветвей выполняемого кода;
- 2) обнаружение ошибок в работе интегрированного продукта;
- 3) доказательство правильной работы модулей;
- 4) проверка правильности функционирования интерфейса.

18. При нисходящем способе проектирования на ранних этапах предпочтителен метод

- 1) системного тестирования;
- 2) черного ящика;

- 3) белого ящика;
- 4) интеграционного тестирования.

19. При восходящем способе проектирования на ранних этапах предпочтителен метод

- 1) черного ящика;
- 2) системного тестирования;
- 3) интеграционного тестирования;
- 4) белого ящика.

20. При использовании метода главного программиста

- 1) тестированием занимается член команды;
- 2) тестирование не производится из-за ненужности;
- 3) тестирование выполняет сам главный программист;
- 4) тестирование выполняет дублер.

*Краткие методические указания*

Промежуточный тест проводится в электронной форме во время последнего в учебном периоде практического занятия. Тест состоит из 20 тестовых заданий. На выполнение теста отводится 20 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем.

*Шкала оценки*

№	Баллы	Описание
5	19–20	Процент правильных ответов от 95% до 100%
4	16–18	Процент правильных ответов от 80 до 94%
3	13–15	Процент правильных ответов от 65 до 79%
2	9–12	Процент правильных ответов от 45 до 64%
1	0–8	Процент правильных ответов менее 45%