

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

**УТВЕРЖДАЮ**

Заведующий кафедрой  
электроники



Усков Г.К.

20.05.2025 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.01.01 Технологии разработки программного обеспечения

**1. Код и наименование направления подготовки/специальности:**

09.04.01 Информатика и вычислительная техника

**2. Профиль подготовки/специализация:**

Автоматизированные информационно-измерительные системы

**3. Квалификация выпускника: магистр**

**4. Форма обучения: очная**

**5. Кафедра, отвечающая за реализацию дисциплины: электроники**

**6. Составители программы:**

Коровченко Игорь Сергеевич, к.ф.-м.н., доцент

**7. Рекомендована: НМС физического факультета 20.05.2025, № протокола: 5**

**8. Учебный год: 2025/2026**

**Семестр(ы)/Триместр(ы): 1, 2**

## 9. Цели и задачи учебной дисциплины

Целью является изучение методов проектирования и производства программного продукта, принципов построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения; методов организации работы в коллективах разработчиков программного обеспечения, формирование навыков проектирования, реализации, оценки качества и анализа эффективности программного обеспечения.

Задачи:

- Дать студентам необходимые знания о методах разработки регламентов соблюдения требований информационной безопасности совместно с соответствующими службами организации для всего жизненного цикла баз данных в области автоматизированных информационно-измерительных систем;
- Дать студентам необходимые знания о методах разработки внутренних правил, методик и регламентов проведения работ по разработке компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем.

## 10. Место учебной дисциплины в структуре ООП:

Дисциплина относится к вариативной части блока Б1.

## 11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-2	Способен организовывать процесс разработки компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем	ПК-2.1	Владеть методами разработки регламентов соблюдения требований информационной безопасности совместно с соответствующими службами организации для всего жизненного цикла баз данных в области автоматизированных информационно-измерительных систем	<b>Знать:</b> - методы организации работы в коллективах разработчиков программного обеспечения в части обеспечения должного уровня безопасности; - методы и средства разработки программной документации в части требований к безопасности.  <b>Уметь:</b> - использовать методы для получения кода с заданной степенью качества в области безопасности программного обеспечения; - пользоваться ремонтной и эксплуатационной технической документацией в области безопасности программного обеспечения.  <b>Владеть:</b> - методами выработки требований к программному обеспечению в части его безопасности для пользователей; - методами проектирования программного обеспечения с использованием специализированных программных пакетов для тестирования на уязвимости в части безопасности и исправления уязвимостей.
		ПК-2.6	Владеть методами разработки внутренних правил, методик и регламентов проведения работ по разработке	<b>Знать:</b> - модели процесса разработки программного обеспечения; - основные принципы процесса разработки программного обеспечения; - основные подходы к интегрированию программных модулей;

			компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем	<p>- основные методы и средства эффективной разработки;</p> <p>- концепции и реализации программных процессов;</p> <p>- методы организации работы в коллективах разработчиков программного обеспечения;</p> <p>- методы и средства разработки программной документации.</p> <p><b>Уметь:</b></p> <p>- использовать методы для получения кода с заданной функциональностью и степенью качества;</p> <p>- пользоваться ремонтной и эксплуатационной технической документацией.</p> <p><b>Владеть:</b></p> <p>- владеть основными методологиями процессов разработки программного обеспечения;</p> <p>- методами выработки требований к программному обеспечению;</p> <p>- методами проектировании программного обеспечения с использованием специализированных программных пакетов.</p>
--	--	--	--	---

## 12. Объем дисциплины в зачетных единицах/час – 6 / 216

Форма промежуточной аттестации *экзамен, экзамен*

## 13. Трудоемкость по видам учебной работы

Вид учебной работы	Всего	Трудоемкость		
		По семестрам		
		1	2	
Аудиторные занятия	118	64	54	
в том числе:	лекции	50	32	18
	практические	68	32	36
	лабораторные			
Самостоятельная работа	26	8	18	
Форма промежуточной аттестации: экзамен – 36 час.	72	36	36	
Итого:	216	108	108	

### 13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
<b>1. Лекции</b>			
1.1	Введение в технологии разработки программных средств	Основные понятия и определения. Жизненный цикл программных средств. Работа с профессиональной литературой в зависимости от поставленной задачи.	–
1.2	Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла	Стратегии разработки программных средств и систем: базовые стратегии разработки ПС; каскадная стратегия разработки; инкрементная стратегия; эволюционная стратегия. Модели ЖЦ, реализующие каскадную стратегию разработки ПС: Общие сведения о каскадных моделях; классическая каскадная модель; каскадная модель с обратными связями; каскадная модель по ГОСТ Р ИСО/МЭК ТО 15271-2002; V-образная модель. Модели быстрой разработки ПО: базовая модель RAD; RAD-модель, основанная на моделировании предметной области; RAD-модель параллельной разработки ПО;	–

		<p>модель быстрой разработки по ГОСТ Р ИСО/МЭК ТО 15271-2002. Модели ЖЦ, реализующие инкрементную стратегию разработки ПС: общие сведения об инкрементных моделях; инкрементная модель с уточнением требований на начальных этапах разработки; варианты инкрементной модели по ГОСТ Р ИСО/МЭК ТО 15271-2002; инкрементная модель экстремального программирования. Модели ЖЦ, реализующие эволюционную стратегию разработки ПС: общие сведения; эволюционная модель по ГОСТ Р ИСО/МЭК ТО 15271-2002; структурная эволюционная модель быстрого прототипирования; эволюционная модель прототипирования по ГОСТ Р ИСО/МЭК ТО 15271-2002; спиральная модель Боэма; упрощенные спиральные модели. Анализ выбранного стиля программирования. Разработка проекта программного обеспечения. Разработка структурного алгоритма. Разработка программного продукта с использованием объектно-ориентированного программирования. Разработка справочной системы. Тестирование методом «белого ящика», Тестирование методом «черного ящика». Способы анализа граничных решений, Способы создания диаграмм причин-следствий. Нисходящее тестирование интеграций, Восходящее тестирование интеграций. Анализ предметной области, Автоматизированное тестирование. Отладка и оптимизация программ. Работа в составе бригады. Работа с литературой, подготовка к занятиям, доработка и усовершенствование программного кода, адаптация программного обеспечения под другие браузеры и платформы.</p>	
1.3	Выбор модели жизненного цикла для конкретного проекта	<p>Классификация проектов по разработке ПС. Процедура выбора модели ЖЦ ПС. Адаптация модели ЖЦ разработки ПС к условиям конкретного проекта. Знакомство с интегрированным средством Star UML / Rational Rose. Основы UML. Изучение постановки задачи. Создание диаграмм: вариантов использования и действующих лиц; последовательности; кооперативной диаграммы; состояний для класса Заказ; активности для варианта использования «Выполнить поставку Заказа».</p>	–
1.4	Классические методологии разработки программных средств	<p>Структурное программирование. Модульное проектирование ПС. Методы восходящего проектирования. Методы расширения ядра. Метод Джексона. Оценка структурного разбиения ПС. Пакеты и классы: уточнение методов и свойств классов; описание связей между классами; использование текста в информации классов на естественных языках. Работа с литературой, подготовка к занятиям, доработка и усовершенствование моделей.</p>	–
1.5	CASE-технологии структурного анализа и проектирования программных средств	<p>Общие сведения о CASE-технологиях. Методология функционального моделирования IDEF0. Методология структурного анализа потоков данных DFD. Методология информационного моделирования IDEF1X. Методологии, ориентированные на данные. Построение диаграммы компонентов. CASE-технологии структурного анализа и проектирования программных средств.</p>	–
1.6	Методология объектно-ориентированного анализа и проектирования сложных систем	<p>Основы объектно-ориентированного анализа и проектирования. Математические основы объектно-ориентированного анализа и проектирования. Основы языка UML. Построение диаграммы размещения. Кодогенерация модельных элементов. Построение диаграмм UML. Методология объектно-ориентированного анализа и проектирования сложных систем.</p>	–
<b>2. Практические занятия</b>			
2.1	Введение в технологии разработки программных средств	<p>Общие принципы формирования технического задания.</p>	–
2.2	Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла	<p>Классификация проектов. Выбор стратегии разработки программного обеспечения для небольших, средних, больших и очень больших проектов с точки зрения разных подходов и команд. Выбор стратегии развития</p>	–

		программного обеспечения для небольших, средних, больших и очень больших проектов с точки зрения разных подходов и команд.	
2.3	Выбор модели жизненного цикла для конкретного проекта	Формирование технического задания для создания программного обеспечения в рамках традиционных и гибких подходов. Формирование технического задания для модернизации существующего программного обеспечения в рамках традиционных и гибких подходов, исходя из задачи заказчика.	–
2.4	Классические методологии разработки программных средств	Разработка программного обеспечения на основе разных парадигм, принципов и методов.	–
2.5	CASE-технологии структурного анализа и проектирования программных средств	Автоматизация процессов разработки программного обеспечения. Программные и аппаратные инструменты для автоматизации процессов разработки программного обеспечения.	–
2.6	Методология объектно-ориентированного анализа и проектирования сложных систем	Основы работы с UML. Создание архитектуры программного обеспечения на основе анализа технических требований.	–
<b>3. Лабораторные занятия</b>			

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Введение в технологии разработки программных средств	8	8		4	20
2	Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла	8	8		4	20
3	Выбор модели жизненного цикла для конкретного проекта	8	8		4	20
4	Классические методологии разработки программных средств	8	8		4	20
5	CASE-технологии структурного анализа и проектирования программных средств	9	9		4	22
6	Методология объектно-ориентированного анализа и проектирования сложных систем	9	27		6	42
Итого:		50	68		26	144

### 14. Методические указания для обучающихся по освоению дисциплины:

Для успешного освоения дисциплины необходима регулярная и планомерная работа с конспектом лекций и литературой.

После окончания лекции нужно просматривать конспект для определения материала, вызывающего затруднения для понимания. После этого следует обратиться к рекомендуемой в настоящей программе литературе с целью углубленного изучения проблемного вопроса. В общем случае работа лишь с одним литературным источником часто является недостаточной для полного понимания, поэтому необходимо просматривать несколько источников для выбора того, который наиболее полно и доступно освещает изучаемый материал. В случае если проблемы с пониманием остались, необходимо обратиться к преподавателю на ближайшей лекции с заранее сформулированными вопросами.

Для успешного освоения лекционного курса рекомендуется регулярно повторять изученный материал.

Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа может включать в себя следующие составляющие:

- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;
- работа над темами для самостоятельного изучения;
- участие в работе студенческих научных конференций, олимпиад;
- подготовка экзамену.

Кроме литературы из основного списка рекомендуется самостоятельно использовать дополнительную. Независимо от вида учебника, работа с ним должна происходить в течение всего семестра. Эффективнее работать с учебником не после, а перед лекцией.

В процессе самостоятельной работы следует занимать активную позицию и пользоваться не только рекомендованной литературой, но и самостоятельно найденными источниками. При изучении дисциплины рекомендуется использовать возможности сети Интернет для получения дополнительной информации по рассматриваемой теме.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей, вовремя подключаться к онлайн-занятиям, ответственно подходить к выполнению заданий для самостоятельной работы.

## 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

### а) основная литература:

№ п/п	Источник
1.	Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. – СПб.: Питер, 2020. – 352 с.
2.	Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста / Р. Мартин. – СПб.: Питер, 2020. – 464 с.
3.	Мартин Р. Чистый Agile. Основы гибкости / Р. Мартин. – СПб.: Питер, 2020. – 272 с.
4.	Мартин Р. Идеальный программист. Как стать профессионалом разработки ПО / Р. Мартин. – СПб.: Питер, 2020. – 224 с.
5.	Бек К. Экстремальное программирование. Разработка через тестирование / К. Бек. – СПб.: Питер, 2017. – 230 с.

### б) дополнительная литература:

№ п/п	Источник
1.	Чакон С. Git для профессионального программиста / С. Чакон, Б. Страуб. – СПб.: Питер, 2018. – 496 с.
2.	Сергеев С. 15 базовых советов по Git для эффективной работы каждый день / С. Сергеев. – URL: [https://habr.com/ru/company/manychat/blog/511946/] (Дата обращения: 01.06.2020).
3.	Удачная модель ветвления для Git. – URL: [https://habr.com/ru/post/106912/] (Дата обращения: 01.06.2020).
4.	Что такое Терминал Windows?. – URL: [https://docs.microsoft.com/ru-ru/windows/terminal/] (Дата обращения: 01.06.2020).
5.	Join GitHub. – URL: [https://github.com/join] (Дата обращения: 01.06.2020).
6.	Understanding the GitHub flow. – URL: [https://guides.github.com/introduction/flow/] (Дата обращения: 01.06.2020).
7.	Ст. 1261 ГК РФ ч.4. Программы для ЭВМ. – URL: [http://www.consultant.ru/document/cons\_doc\_LAW\_64629/ce1359ed5b9bd99896d7a496c7887e7c223a2cbc/] (Дата обращения: 01.06.2020).
8.	Ст. 1262 ГК РФ ч.4. Государственная регистрация программ для ЭВМ и баз данных. – URL: [http://www.consultant.ru/document/cons\_doc\_LAW\_64629/d0887a7ca3da6c85fbbce19815b9b1ead5e67687/] (Дата обращения: 01.06.2020).
9.	Статья 1261 ГК РФ. Программы для ЭВМ и Комментарий к Ст. 1261 ГК РФ. – URL: [http://gkodeksrf.ru/ch-4/rzd-7/gl-70/st-1261-gk-rf] (Дата обращения: 01.06.2020).
10.	Фомин М. Искусство софта: как работает авторское право на программное обеспечение? / М. Фомин. – URL: [https://www.computerra.ru/254823/iskusstvo-softa-kak-rabotaet-avtorskoe-pravo-na-programmnoe-obespechenie/] (Дата обращения: 01.06.2020).
11.	Авторские права на программы для ЭВМ. – URL: [https://sumip.ru/biblioteka/avtorskoye-pravo/obekty-avtorskogo-prava/avtorskim-prava-na-programmy-dlya-evm/] (Дата обращения: 01.06.2020).
12.	Джола В.К. Международное нормативно-правовое регулирование охраны прав на программы для ЭВМ / В.К. Джола, А.Б. Никишов. – URL: [https://urfac.ru/?p=2340] (Дата обращения: 01.06.2020).
13.	Open Source Initiative (OSI). – URL: [https://opensource.org] (Дата обращения: 01.06.2020).

14.	Open Source Licenses by Category. – URL: [https://opensource.org/licenses/category] (Дата обращения: 01.06.2020).
15.	Лицензирование программного обеспечения. – URL: [https://habr.com/ru/post/275995/] (Дата обращения: 01.06.2020).

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)\*:

№ п/п	Ресурс
1.	Электронная библиотека Зональной научной библиотеки Воронежского госуниверситета : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/zgate?lnit+elib.xml,simple_elib.xsl+rus">https://lib.vsu.ru/zgate?lnit+elib.xml,simple_elib.xsl+rus</a>
2.	Электронно-библиотечная система "БиблиоТех" : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1486">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1486</a>
3.	Электронно-библиотечная система «ЮРАЙТ» : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1457">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1457</a>
4.	Электронно-библиотечная система BOOK.ru.(изд-во "КноРус") : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1436">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1436</a>
5.	Национальный цифровой ресурс "РУКОНТ" : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1401">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1401</a>
6.	Электронно-библиотечная система "ZNANIUM.COM" (изд-во "ИНФРА-М") : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1360">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1360</a>
7.	Электронно-библиотечная система ibook.ru : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1344">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1344</a>
8.	Электронно-библиотечная система IPRbooks : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1343">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1343</a>
9.	Электронно-библиотечная система «КнигаФонд» : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1336">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1336</a>
10.	Электронно-библиотечная система IQLib : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1310">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1310</a>
11.	Электронно-библиотечная система "Издательство "Лань" : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1308">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1308</a>
12.	Электронно-библиотечная система "Университетская библиотека online" : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1307">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1307</a>
13.	Электронно-библиотечная система "Консультант студента" : электронно-библиотечная система. – URL : <a href="https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1306">https://lib.vsu.ru/?p=4&amp;t=2d&amp;id=1306</a>

## 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1.	Зубра А. С. Культура умственного труда студента : пособие для студентов вузов / А.С.Зубра. — 2-е изд., испр. и доп. — Мн. : Дикта, 2007. — 228с.
2.	Горцевский А.А. Организация самостоятельной работы студента / А.А. Горцевский, М.И. Любицына. — Л. : ЛГУ, 1958. — 50 с.
3.	Скрипченко Ю.С. Объектно-ориентированное программирование в примерах и задачах : учебное пособие / Ю.С. Скрипченко, Н.А. Тюкачев, В.Г. Хлебостроев. — Воронеж : Воронеж. гос. ун-т, 2006. — 160 с.
4.	Орлов С. А. Теория и практика языков программирования : [учебник по направлению "Информатика и вычисл. техника"] / С.А. Орлов. — 2-е изд. — Санкт-Петербург [и др.] : Питер, 2017. — 685 с.
5.	Себеста Роберт У. Основные концепции языков программирования / Роберт У. Себеста ; Пер. с англ. Д.А. Ключина, А.В. Назаренко ; Под ред. Д.А. Ключина 5-е изд. — М. и др. : Вильямс, 2001. — 668 с.

## 17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ, электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины для проведения текущего контроля и в качестве информационного ресурса используются технологии электронного обучения и дистанционные образовательные технологии на базе образовательного портала "Электронный университет ВГУ" по адресу [edu.vsu.ru](http://edu.vsu.ru), а также другие доступные ресурсы сети Интернет.

## 18. Материально-техническое обеспечение дисциплины:

Мультимедийная аудитория (ауд. 401): специализированная мебель, компьютеры, мультимедиа-проектор, настенный экран для проектора, аудио колонки

WinPro 8, Linux Debian, Open Office, Google Chrome

Помещение для самостоятельной работы обучающихся (ауд. 401): специализированная мебель, мультимедиа-проектор, настенный экран для проектора, аудио колонки,

компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ВГУ

WinPro 8, Linux Debian, Open Office, Google Chrome, Visual Studio Code, StarUML, Maxima, Octave, MATLAB, JVM, Scala, Haskel, Closure, Java, Kotlin, Python, Go, GCC, CLANG, ReactiveX, VHDL, Verilog, ReactiveX, VHDL, Verilog, SimulIDE Circuit Simulator, Wokwi Simulator, NI LabView, Arduino Studio, MicroCap Evaluation

## 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Введение в технологии разработки программных средств. Стратегии разработки программных средств и систем и реализующие их модели жизненного цикла. Выбор модели жизненного цикла для конкретного проекта. Классические методологии разработки программных средств. CASE-технологии структурного анализа и проектирования программных средств. Методология объектно-ориентированного анализа и проектирования сложных систем.	ПК-2 Способен организовывать процесс разработки компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем	ПК-2.1 Владеть методами разработки регламентов соблюдения требований информационной безопасности совместно с соответствующими службами организации для всего жизненного цикла баз данных в области автоматизированных информационно-измерительных систем	Комплект тестовых заданий № 1
			ПК-2.6 Владеть методами разработки внутренних правил, методик и регламентов проведения работ по разработке компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем	Комплект тестовых заданий № 2

## 20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в формах выполнения практико-ориентированных заданий - лабораторных работ и тестирования на портале Электронный университет ВГУ.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

### 20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: комплекта тестовых заданий. Пример тестового задания:

#### Комплект тестовых заданий № 1

1. Программный продукт – это

- а) программа для удовлетворения нужд разработчиков, предназначенная для продажи
- б) комплекс взаимосвязанных программ для решения определенной проблемы массового спроса, подготовленный к реализации как любой вид промышленной продукции
- в) программная реализация решения задачи на компьютере**
- г) результат разработки какого-либо технического задания

2. Отличительной особенностью программных продуктов является

- а) системность
- б) простота
- в) универсальность
- г) надежность**

3. Сопровождение программного продукта – это

- а) снабжение программного продукта необходимой документацией
- б) обнаружение и исправление ошибок
- в) поддержка работоспособности программного продукта, переход на его новые версии, внесение изменений, исправление обнаруженных ошибок и т.д.**
- г) проверка работоспособности каждой разработанной функции, процедуры, модуля

4. Мобильность программных продуктов – это

- а) независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.**
- б) точность выполнения предписанных функций обработки
- в) способность к внесению изменений
- г) обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства

5. В условиях существования рынка программных продуктов важными его характеристиками являются:

- а) количество продаж, наличие программ-конкурентов, длительность продаж
- б) стоимость, количество продаж, время нахождения на рынке, известность фирмы разработчика и программы**
- в) внешний интерфейс программы, количество продаж, наличие программ конкурентов
- г) модифицируемость, надежность, универсальность, известность фирмы – разработчика

6. Основными показателями качества программных продуктов является:

- а) алгоритмическая сложность, полнота и системность функций обработки, объем файлов программы
- б) стоимость, количество продаж, наличие программных продуктов аналогичного назначения

- в) **мобильность, надежность, эффективность, модифицируемость, коммуникативность, учет человеческого фактора**
- г) модифицируемость, надежность, наличие программных продуктов аналогичного назначения

7. При индивидуальной разработке фирма-разработчик создает программный продукт для...

- а) **конкретного заказчика**
- б) массового использования
- в) внедрения в специальные организации
- г) для удовлетворения собственных нужд

8. Модифицируемость программных продуктов означает...

- а) независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.
- б) точность выполнения предписанных функций обработки
- в) **способность к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п.**
- г) обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства

9. Жизненный цикл программы – это

- а) временной интервал, начиная с момента замысла программы и кончая прекращением всех видов его пользований
- б) временной интервал, начиная с момента введения программы в эксплуатацию
- в) промежуток времени, который определяет наиболее эффективное использование создаваемой программы
- г) **временная характеристика разработки программного продукта**

10. Программы малого Жизненного Цикла – это программы

- а) когда время разработки программы значительно меньше времени эксплуатации программы
- б) когда время разработки программы значительно больше времени использования программы
- в) **когда время разработки программы равно времени эксплуатации программы**
- г) нет правильного ответа

11. На этапе сбора и анализа требований заказчик должен

- а) выяснить, прежде всего, необходимость обеспечения безопасности системы и данных
- б) **выяснить, прежде всего, функции, которые должен выполнять программный продукт**
- в) выяснить, прежде всего, сроки написания программы
- г) собрать литературу по разрабатываемому программному продукту

12. Самая распространенная модель Жизненного цикла программного продукта это

- а) итерационная**
- б) V - образная
- в) спиральная
- г) каскадная

13. Классическая модель ЖЦПО характеризуется следующими основными особенностями

- а) последовательным выполнением входящих в ее состав этапов**
- б) наличием обратных связей между этапами
- в) отсутствием временного перекрытия этапов
- г) отсутствием (или определенным ограничением) возврата к предыдущим этапам
- д) наличием результата после каждого этапа разработки

14. Выберите правильную последовательность этапов спиральной модели жизненного цикла программного продукта:

- а) техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка
- б) кодирование, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка
- в) кодирование, техническое проектирование, уточнение функциональных характеристик, тестирование и отладка
- г) определение требований, анализ, реализация и тестирование, внедрение**

15. V – образная модель ЖЦ разработки ПО предполагает:

- а) отсутствие временного перекрытия этапов
- б) наличие обратной связи**
- в) возможность сокращения времени разработки ПО
- г) возможность увеличения жизненного цикла программного продукта

16. На втором этапе каскадной модели ЖЦ разработки ПО (Требования ПО) осуществляется...

- а) составление концептуальной структуры системы
- б) определение функциональности программного компонента**
- в) составление детальной спецификации архитектуры системы
- г) составление набора тестовых данных

17. Проверка корректности требований при использовании V – образной модели ЖЦ разработки ПО осуществляется...

- а) после каждого этапа разработки
- б) после разработки всей системы
- в) после разработки черновой версии системы
- г) после разработки набора тестовых данных**

18. Выберите правильную последовательность этапов жизненного цикла программного продукта:

- а) техническое проектирование, сопровождение ПП, сбор и анализ требований заказчика, кодирование, уточнение функциональных характеристик, тестирование и отладка
- б) сбор и анализ требований, проектирование системы, кодирование, создание программной документации, сопровождение
- в) кодирование, сбор и анализ требований заказчика, техническое проектирование, уточнение функциональных характеристик, сопровождение ПП, тестирование и отладка
- г) **сбор и анализ требований заказчика, уточнение функциональных характеристик, техническое проектирование, кодирование, тестирование и отладка, сопровождение ПП**

19. Во вспомогательные процессы ЖЦ программного продукта входит:

- а) **документирование, верификация, аттестация, обеспечение качества, совместная оценка, разрешение проблем, аудит**
- б) управление, создание инфраструктуры, усовершенствование, обучение
- в) разработка, приобретение, поставка, эксплуатация, сопровождение
- г) кодирование, тестирование, сопровождение

20. Одним из достоинств классического жизненного цикла программного продукта является

- а) **дает план и временной график по всем этапам проекта**
- б) в конце всей работы заказчику будут доступны результаты проекта
- в) системный анализ каждого элемента программы
- г) отсутствие временного перекрытия этапов разработки программного продукта

21. Итерационная модель ЖЦПО характеризуется следующими основными особенностями:

- а) последовательным выполнением входящих в ее состав этапов
- б) наличием обратных связей между этапами
- в) отсутствием временного перекрытия этапов
- г) отсутствием (или определенным ограничением) возврата к предыдущим этапам
- д) **возможность проведение корректировки после каждого этапа**

22. В конце каждого витка спирали спиральной модели ЖЦ разработки ПО получаем...

- а) **готовый программный продукт**
- б) одну версию программного продукта
- в) версию программного продукта с набором тестовых данных
- г) черновую модель программного продукта

23. Спиральная модель ЖЦ разработки ПО предполагает:

- а) **отсутствие временного перекрытия этапов**
- б) наличие обратной связи

в) возможность сокращения времени разработки ПО

24. На втором этапе каскадной модели ЖЦ разработки ПО (Требования к ПО) осуществляется...

- а) **определение функциональности программного компонента**
- б) составление детальной спецификации архитектуры системы
- в) составление концептуальной структуры системы
- г) написание программного кода

25. Происходит ли интеграция отдельных компонент системы при разработке ПП по экстремальной модели ЖЦ?

- а) **да**
- б) Нет

26. Какую модель жизненного цикла разработки ПО целесообразнее использовать, если нет четко определенных требований к будущей системе?

- а) каскадную
- б) спиральную
- в) V – образную
- г) **итерационную**

27. Программное средство - это

- а) программа для удовлетворения нужд разработчиков, предназначенная для продажи
- б) программа, предназначенная для многократного применения на различных объектах и разработанная любым способом
- в) **программная реализация решения задачи на компьютере**
- г) результат разработки какого-либо технического задания

28. Качество ПП - это

- а) **совокупность свойств этого продукта, которые удовлетворяют определенным потребностям пользователей в соответствии с его назначением;**
- б) те свойства данного продукта, благодаря которым программный продукт может функционировать в любой программной среде;
- в) совокупность свойств программного продукта, которые удовлетворяют требованиям ЕСПД и базовым международным стандартам.

29. Изучаемость ПП включает в себя:

- а) **удобочитаемость, тестируемость, информативность;**
- б) внедряемость, понятность, удобочитаемость;
- в) документированность, понятность, удобочитаемость

30. Функциональная пригодность программного продукта включает в себя:

- а) точность, защищенность, надежность;
- б) эффективность и внедряемость;
- в) **понятность, стабильность, надежность.**

31. Понятность ПП заключается в ...

- а) **наличии в составе программы информации необходимой и достаточной для понимания назначения программы, существующих ограничений, входных и выходных данных и результатов обработки;**
- б) степени, которой пользователь может изучить назначение ПП, результат ее работы и текст этой программы;
- в) быстрой модификации с целью приспособления к изменяющимся условиям функционирования.

32. Программа является надежной, если...

- а) выдаваемый результат работы имеет допустимые значения отклонений от аналогичных отклонений;
- б) она продолжает свою работу при возникновении сбоев;
- в) **она при всех одинаково вводимых данных обеспечивает полную повторяемость результата.**

33. Программа является эффективной, если...

- а) она правильно работает при любых допустимых вариантах исходных данных;
- б) **объем требуемых ресурсов для ее выполнения не превышает допустимой границы;**
- в) она работает должным образом не только автономно, но и как часть информационной системы.

34. Программа является совместимой, если...

- а) **она работает должным образом не только автономно, но и как часть информационной системы;**
- б) ее качества могут быть продемонстрированы на практике;
- в) она допускает быструю модификацию с целью приспособления к изменяющимся условиям функционирования.

Описание технологии проведения.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета – в форме тестирования. Критерии оценивания приведены ниже. Тест выполняется на практическом занятии в виде письменной работы с последующей проверкой преподавателем.

Результаты текущей аттестации учитываются преподавателем при проведении промежуточной аттестации (экзамена).

Критерии оценки компетенций (результатов обучения) при выполнении теста:

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
80–100% правильных ответов	Повышенный уровень	Отлично

66–79% правильных ответов	Базовый уровень	Хорошо
50–65% правильных ответов	Пороговый уровень	Удовлетворительно
0–49% правильных ответов	–	Неудовлетворительно

## 20.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется в форме экзамена. Согласно П ВГУ 2.1.07 – 2024 Положению о проведении промежуточной аттестации обучающихся по образовательным программам высшего образования, оценка на экзамене может быть выставлена по результатам текущей успеваемости обучающегося в течение семестра и на основании процедуры и критериев оценивания, представленных в рабочей программе, но не ранее чем на заключительном занятии.

Промежуточная аттестация по дисциплине (экзамен) осуществляется с помощью следующих оценочных средств: теоретических вопросов. В контрольно-измерительный материал включаются два теоретических вопроса, позволяющих оценить уровень полученных знаний, умений и навыков.

### Перечень вопросов к экзамену:

#### 1 семестр

1. Программные продукты: назначение, характеристики
2. Основные понятия программного обеспечения.
3. Программа, программное обеспечение, задачи и приложения. Технологические и функциональные задачи.
4. Процесс создания программ: постановка задачи, алгоритмизация, программирование.
5. Характеристика программного продукта и его специфика.
6. Классификация программных продуктов.
7. Понятие жизненного цикла. Основные и вспомогательные процессы жизненного цикла.
8. Модели жизненного цикла разработки программного продукта.
9. Качество программной системы. Критерии оценки качества программных систем, характеристики качества и показатели качества.
10. Общие характеристики качества программных систем.
11. Методы управления качеством, используемые в современных технологиях программирования. Аттестация программных систем.
12. Функциональные и нефункциональные требования к программной системе.
13. Методы первичного сбора требований. Анализ требований.
14. Правила формулировки непротиворечивых требований. Техническое задание.
15. Внутренняя организация программного обеспечения.

#### 2 семестр

1. Методы проектирования программного обеспечения и признаки их классификации.
2. Неавтоматизированное и автоматизированное проектирование алгоритмов и программ.
3. Структурное проектирование и его методы.
4. Принцип системного проектирования.
5. Нисходящее проектирование.
6. Модульное проектирование.
7. Объектно-ориентированное проектирование.
8. Проектирование интерфейса пользователя.
9. Кодирование.

10. Модульное программирование.
11. Структурное программирование.
12. Объектно-ориентированное программирование.
13. Стиль программирования.
14. Разработка справочной системы программного обеспечения.
15. Создание документации пользователя.
16. Основные принципы организации тестирования.
17. Виды тестирования.
18. Программные ошибки.
19. Методы структурного тестирования программного обеспечения.
20. Принцип «белого и черного ящика».
21. Пошаговое и монолитное тестирование модулей.
22. Нисходящее и восходящее тестирование программного обеспечения.
23. Методы функционального тестирования.
24. Метод эквивалентного разбиения.
25. Метод анализа граничных условий.
26. Метод функциональных диаграмм.
27. Комплексное тестирование.
28. Отладка программ.
29. Сопровождение программ.
30. Категории специалистов, занятых разработкой и эксплуатацией программ.
31. Принципы и методы коллективной разработки программных продуктов.
32. Организация коллективной работы программистов.

#### Описание технологии проведения

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Требования к выполнению заданий (шкалы и критерии оценивания).

Для оценивания результатов обучения на экзамене используются следующие **показатели**:

- 1) Знать о методах разработки регламентов соблюдения требований информационной безопасности совместно с соответствующими службами организации для всего жизненного цикла баз данных в области автоматизированных информационно-измерительных систем;
- 2) Знать о методах разработки внутренних правил, методик и регламентов проведения работ по разработке компьютерного программного обеспечения в области автоматизированных информационно-измерительных систем.

Для оценивания результатов обучения на экзамене используется **шкала**: «отлично», «хорошо», «удовлетворительно», «не удовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения:

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся в полной мере владеет понятийным аппаратом данной области науки (теоретическими основами дисциплины), способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач, готов к использованию современных подходов и средств реализации практических задач.	Повышенный уровень	Отлично
Обучающийся владеет понятийным аппаратом данной области	Базовый уровень	Хорошо

науки (теоретическими основами дисциплины), способен применять теоретические знания для решения практических задач, готов к использованию типовых подходов и средств реализации практических задач.		
Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен к использованию типовых подходов и средств реализации практических задач.	Пороговый уровень	Удовлетворительно
Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.	–	Неудовлетворительно

**ЛИСТ СОГЛАСОВАНИЙ****РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ****Направление/специальность**

09.04.01 Информатика и вычислительная техника

**Дисциплина**

Б1.В.01.01 Технологии разработки программного обеспечения

**Профиль подготовки/специализация**

Автоматизированные информационно-измерительные системы

**Форма обучения** очная**Учебный год** 2025/2026

## Ответственный исполнитель

Доцент кафедры электроники \_\_\_\_\_ .\_\_ 20\_\_

СОГЛАСОВАНО

Куратор ООП  
по направлению/специальности \_\_\_\_\_ .\_\_ 20\_\_

Начальник отдела обслуживания ЗНБ \_\_\_\_\_ .\_\_ 20\_\_

---

---

Программа рекомендована НМС физического факультета 28.10.2024 протокол № 8 от 28.10.2024 г.