

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
ФТД.В.01 Гибкие технологии разработки программного обеспечения

1. Код и наименование направления подготовки/специальности:

11.03.02 Инфокоммуникационные технологии и системы связи

2. Профиль подготовки/специализация:

Инфокоммуникационные технологии и системы связи

3. Квалификация (степень) выпускника: бакалавр

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: 2028/2029

Семестр(ы): 7

9. Цели и задачи учебной дисциплины: познакомить студентов с методами разработки коммерческого, медицинского и промышленного программного обеспечения.

10. Место учебной дисциплины в структуре ООП:

Дисциплина является одной из основополагающих для приобретения необходимых специалисту профессиональных навыков.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-6	Способен разрабатывать и тестировать программное обеспечение для решения задач в рамках профессиональной деятельности	ПК-6.2	Осуществляет обоснованный выбор языка высокого уровня для разработки программного обеспечения радиоэлектронных средств	Умеет проектировать сложные системы с использованием современного специализированного программного обеспечения для работы в команде Учитывает особенности современных языков программирования для корректной реализации программного обеспечения Проектирует современное программное обеспечение с учетом актуальных практик разработки программного обеспечения
		ПК-6.4	Создает программный код, используя современные среды разработки программных продуктов	Применяет итерации в качестве основы для постоянного совершенствования программы, добавления новых функций и поддержки программного обеспечения Использует спринты как основу взаимодействия с командой разработки Использует гибкие методологии разработки программного обеспечения Применяет методы оценки трудозатрат по написанию кода программ

12. Объем дисциплины в зачетных единицах/час. — 2 / 72.

Форма промежуточной аттестации зачет.

13. Виды учебной работы

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
Аудиторные занятия	32	7	
в том числе: лекции	16	16	
практические			
лабораторные	16	16	
Самостоятельная работа	40	40	
Форма промежуточной аттестации (зачет – 0 час. / экзамен – ___ час.)			
Итого:	72	72	

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Этапы жизненного цикла программного обеспечения	Подготовка. Проектирование. Создание. Поддержка.
1.2	Основные модели разработки программного обеспечения	Каскадная модель, или “водопад”, V-образная модель, разработка через тестирование, инкрементная модель, итеративная (или итерационная) модель^ спиральная модель.
1.3	Основные методологии разработки программного обеспечения	Экстремальное программирование (Extreme Programming, XP) бережливая разработка программного обеспечения (Lean), фреймворк для управления проектами Scrum, разработка, управляемая функциональностью (Feature-driven development, FDD), разработка через тестирование (Test-driven development, TDD), методология “чистой комнаты” (Cleanroom Software Engineering), итеративно-инкрементальный метод разработки (OpenUP), методология разработки Microsoft Solutions Framework (MSF), метод разработки динамических систем (Dynamic Systems Development Method, DSDM), метод управления разработкой Kanban.
1.4	Применение гибких технологий разработки программного обеспечения	Применение Kanban, применение Scrum
2. Практические занятия		
3. Лабораторные работы		
3.1	Этапы жизненного цикла программного обеспечения	Исследование крупных проектов на GitHub
3.2	Основные модели разработки программного обеспечения	Применение модели разработки для медицинского программного обеспечения
3.3	Основные методологии разработки программного обеспечения	Совместная работа над проектом в системе Kanban
3.4	Применение гибких технологий разработки программного обеспечения	Совместная работа над проектом в системе Scrum

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Этапы жизненного цикла программного обеспечения	4	0	4	10	188

2	Основные модели разработки программного обеспечения	4	0	4	10	18
3	Основные методологии разработки программного обеспечения	4	0	4	10	18
4	Применение гибких технологий разработки программного обеспечения	4	0	4	10	18
	Итого:	16	0	16	40	72

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 предполагает 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 зачету.

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
6.	Чакон С. Git для профессионального программиста / С. Чакон, Б. Страуб. – СПб.: Питер, 2018. – 496 с.
7.	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).
8.	Удачная модель ветвления для Git. – URL: [https://habr.com/ru/post/106912/] (Дата обращения: 01.06.2020).
9.	Что такое Терминал Windows?. – URL: [https://docs.microsoft.com/ru-ru/windows/terminal/] (Дата обращения: 01.06.2020).
10.	Join GitHub. – URL: [https://github.com/join] (Дата обращения: 01.06.2020).
11.	Understanding the GitHub flow. – URL: [https://guides.github.com/introduction/flow/] (Дата обращения: 01.06.2020).
12.	Ст. 1261 ГК РФ ч.4. Программы для ЭВМ. – URL: [http://www.consultant.ru/document/cons_doc_LAW_64629/ce1359ed5b9bd99896d7a496c7887e7c223a2c8c/] (Дата обращения: 01.06.2020).
13.	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).
14.	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).
15.	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).
16.	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).
17.	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).
18.	Open Source Initiative (OSI). – URL: [https://opensource.org] (Дата обращения: 01.06.2020).
19.	Open Source Licenses by Category. – URL: [https://opensource.org/licenses/category] (Дата обращения: 01.06.2020).
20.	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
21.	Электронная библиотека Зональной научной библиотеки Воронежского госуниверситета : электронно-библиотечная система. – URL : https://lib.vsu.ru/zgate?Init+elib.xml,simple_elib.xsl+rus
22.	Электронно-библиотечная система "БиблиоТех" : электронно-библиотечная система. – URL : https://lib.vsu.ru/?p=4&t=2d&id=1486
23.	Электронно-библиотечная система "Издательство "Лань" : электронно-библиотечная система. – URL : https://lib.vsu.ru/?p=4&t=2d&id=1308
24.	Электронно-библиотечная система "Университетская библиотека online" : электронно-библиотечная система. – URL : https://lib.vsu.ru/?p=4&t=2d&id=1307
25.	Электронно-библиотечная система "Консультант студента" : электронно-библиотечная система. – URL : https://lib.vsu.ru/?p=4&t=2d&id=1306
26.	Электронно-библиотечная система «ЮРАЙТ» : электронно-библиотечная система. – URL : https://lib.vsu.ru/?p=4&t=2d&id=1457

16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник
1.	Макконнелл, Стив. Совершенный код. Мастер-класс = Code Complete : пер. с англ. / Стив Макконнелл .— Москва : Русская редакция, 2012 .— XX, 867 с.
2.	Фаулер, Мартин. Архитектура корпоративных программных приложений : Пер. с англ. / Мартин Фаулер .— М. : Вильямс, 2004 .— 539, [4]с.
3.	Мартин, Роберт К. Быстрая разработка программ : Принципы, примеры, практика / Роберт К. Мартин, Джеймс В. Ньюкирк, Роберт С. Косс ; Пер. с англ. А.П. Сергеева, Т.А. Шамренко; Под ред. А.П. Сергеева .— М. : Вильямс, 2004 .— 739, [5] с.
4.	Всё о QA: 80 бесплатных материалов по грамотному тестированию – URL: https://tproger.ru/digest/free-software-testing-books/ (дата обращения: 15 июня 2019)

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

При реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологии на базе образовательного портала "Электронный университет ВГУ" по адресу edu.vsu.ru, а также другие доступные ресурсы сети Интернет.

18. Материально-техническое обеспечение дисциплины:

Учебная аудитория (ауд. 407): специализированная мебель, компьютеры

Microsoft Windows, Linux, OpenOffice, браузер Google Chrome, Python

Помещение для самостоятельной работы обучающихся (ауд. 423): специализированная мебель, компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ВГУ

Microsoft Windows, Linux, OpenOffice, браузер Google Chrome

Помещение для самостоятельной работы обучающихся (ауд. 410): специализированная мебель, компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ВГУ

Microsoft Windows, Linux, OpenOffice, браузер Google Chrome, MatLab, NI LabView, Python

Компьютерный класс (ауд. 412) специализированная мебель, принтер, сканер, компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ВГУ

WinPro, Linux Mint, Open Office, AWR Studio, Anaconda, MicroCap Evaluation, Maxima, Octave, CoID, Cube, Lazarus

Помещение для самостоятельной работы обучающихся (ауд. 407): специализированная мебель, компьютеры с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ВГУ

WinPro, OfficeSTD, Интернет-браузер Google Chrome Mozilla Firefox, MatLab, NI LabView, Python

19. Фонд оценочных средств:

19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	ФОС* (средства оценивания)

ПК-6 Способен разрабатывать и тестировать программное обеспечение для решения задач в рамках профессиональной деятельности	ПК-6.2 Осуществляет обоснованный выбор языка высокого уровня для разработки программного обеспечения радиоэлектронных средств	1.1. Этапы тестирования программного обеспечения. 1.2. Техники тест дизайна. 1.3. Уровни тестирования. 1.4. Типы тестирования.	Тесты № 1-4 Лабораторные работы № 1-4
	ПК-6.4 Создает программный код, используя современные среды разработки программных продуктов	1.4. Типы тестирования.	
Промежуточная аттестация			КИМ

* В графе «ФОС» в обязательном порядке перечисляются оценочные средства текущей и промежуточной аттестаций.

19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Для оценивания результатов обучения на экзамене используются следующие показатели:

- 1) знание учебного материала и владение понятийным аппаратом операционных систем;
- 2) умение связывать теорию с практикой;
- 3) умение иллюстрировать ответ примерами, фактами, данными научных исследований в области разработки операционных систем;

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся владеет понятийным аппаратом функциональной электроники (теоретическими основами дисциплины), готов к использованию современных операционных систем на уровне продвинутого пользователя	Базовый уровень	Зачтено
Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен к использованию современных операционных систем на уровне обычного пользователя	Пороговый уровень	Не зачтено

19.3. 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

26.3.1 Перечень вопросов к экзамену:

1. Анализ продукта.
2. Работа с требованиями.
3. Разработка стратегии тестирования и планирование процедур контроля качества.
4. Создание тестовой документации.
5. Тестирование прототипа.
6. Основное тестирование.
7. Стабилизация.

8. Эксплуатация
9. Эквивалентное разделение.
10. Анализ граничных значений.
11. Причина / следствие.
12. Предугадывание ошибки.
13. Исчерпывающее тестирование.
14. Парное тестирование.
15. Модульное тестирование.
16. Интеграционное тестирование.
17. Системное тестирование.
18. Операционное тестирование.
19. Приемочное тестирование.
20. Функциональное тестирование.
21. Тестирование пользовательского интерфейса.
22. Тестирование безопасности.
23. Тестирование взаимодействия.
24. Нагрузочное тестирование.
25. Стрессовое тестирование.
26. Тестирование стабильности или надежности.
27. Объемное тестирование.
28. Тестирование установки.
29. Тестирование удобства пользования.
30. Тестирование на отказ и восстановление.
31. Конфигурационное тестирование.
32. Дымовое тестирование.
33. Регрессионное тестирование.
34. Повторное тестирование.
35. Тестирование сборки.
36. Санитарное тестирование или проверка согласованности/исправности.
37. Мутационное тестирование.

26.3.2 Перечень практических заданий

1. Unit-тесты в Java
2. Unit-тесты в Python
3. Интеграционное тестирование веб-приложений
4. UI-тестирование для веб-приложений

19.3.4 Тестовые задания

19.3.4 Перечень заданий для контрольных работ

19.3.5 Темы курсовых работ

19.3.6 Темы рефератов

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

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 форме тестирования. Критерии оценивания приведены выше.

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

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний и практические задания, позволяющие оценить степень сформированности умений и навыков.

При оценивании используются качественные шкалы оценок. Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся владеет теоретическими основами дисциплины, фрагментарно способен иллюстрировать ответ примерами, фактами, частично умеет применять теоретические знания для решения практических задач</i>	<i>Пороговый уровень</i>	<i>Зачет</i>
<i>Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки, не умеет применять теоретические знания для решения практических задач</i>	–	<i>Незачет</i>