

Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Алтайский государственный технический университет им. И.И. Ползунова»

**СОГЛАСОВАНО**

Декан ФИТ

А.С. Авдеев

## **Рабочая программа дисциплины**

Код и наименование дисциплины: **Б1.О.17 «Тестирование и отладка программного обеспечения»**

Код и наименование направления подготовки (специальности): **09.03.04**

**Программная инженерия**

Направленность (профиль, специализация): **Разработка программно-информационных систем**

Статус дисциплины: **обязательная часть**

Форма обучения: **очная**

<b>Статус</b>	<b>Должность</b>	<b>И.О. Фамилия</b>
Разработал	старший преподаватель	Я.Б. Ерёмин
Согласовал	Зав. кафедрой «ПМ»	Е.Г. Боровцов
	руководитель направленности (профиля) программы	Е.Г. Боровцов

г. Барнаул

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Содержание компетенции	Индикатор	Содержание индикатора
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1	Формализует задачу и предлагает алгоритмическое решение
		ОПК-6.2	Проектирует программные продукты с применением основ информатики
		ОПК-6.3	Осуществляет разработку и тестирование программных продуктов

## 2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.	Объектно-ориентированное программирование, Основы программной инженерии, Типы и структуры данных
Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения.	Выполнение и защита выпускной квалификационной работы, Преддипломная практика

## 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Общий объем дисциплины в з.е. /час: 5 / 180

Форма промежуточной аттестации: Экзамен

Форма обучения	Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
очная	24	24	0	132	62

## 4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Форма обучения: очная

**Лекционные занятия (24ч.)**

- 1. Основы информатики и программирования в сфере тестирования программ. Необходимость процессов тестирования и верификации программного обеспечения. Специфицирование программного обеспечения и его тестирование методом черного ящика. Роли людей, процессов, методов, инструментов и технологий обеспечения качества. {беседа} (4ч.)[1,2,13,14]**  
Формализация задач. Алгоритмическое решение. Тестирование и верификация, основные определения. Тестировщик или QA-инженер. Тестирование черного и белого ящика. Регрессивное тестирование. Процесс тестирования и V-модели. Спецификация в виде инвариантов, предусловий и постусловий. Спецификация в виде Use-Case диаграмм языка UML. Работа тестировщика. Работа с баг-трекинговой системой. Краткая справка по командной работе на GitHub.
- 2. Концепции и атрибуты качества на уровне кода. Отладка кода. Модульное тестирование и документирование проекта {лекция-пресс-конференция} (4ч.)[1,2,5,8]**  
Отладка кода и отладчики. Место процесса модульного тестирования в процессе разработки. Документирование проекта на уровне кода. Обеспечение надёжности в процессе модульного тестирования. Фреймворки модульного тестирования. Некоторые советы по модульному тестированию.
- 3. Применение основных методов разработки в тестировании. Методологии разработки \*DD (TDD, BDD, MDD). Разработка через тестирование. Конструирование программ на основе предварительных тестов {лекция с разбором конкретных ситуаций} (4ч.)[1,6]**  
Применение основных методов и инструментов разработки прямо в процессе тестирования.  
Что такое методология разработки. Немного о MDD (Model Driven Development). Разработка через тестирование (TDD, Test Driven Development). Разработка, управляемая поведением (BDD, Behavior Driven Development). Настройка инструментов и примеры.
- 4. Функциональное автоматизированное тестирование {лекция-пресс-конференция} (4ч.)[1,2,4,9,10,11,12]**  
Зачем нужно функциональное тестирование и зачем его автоматизировать. Подходы к автоматизации. Тестирование настольных (desktop) приложений с помощью Lanit NeuroControl. Тестирование Web-приложений с помощью Selenium и Postman.
- 5. Методы анализа программ. Статические проверки и динамический анализ программ. {беседа} (4ч.)[1,15,16,17]**  
Статический анализ, его необходимость. Линтеры. PVS-Studio. Cppcheck. Онаписании собственных статических анализаторов. Динамический анализ спомощью Valgrind. Встроенные в IDE профилировщики и их возможности.
- 6. Сложности и подходы к тестированию игрового и научного ПО {лекция-пресс-конференция} (4ч.)[1,6,7,9]**  
Тестирование игровой логики на примере

проекта World of Tanks, особенности архитектуры игровых и научных приложений, не позволяющие применять обычные методы тестирования, которые используются с ПО общего назначения. Альтернативные способы повышения качества ПО.

#### **Лабораторные работы (24ч.)**

- 1. Создание спецификации для существующего ПО(4ч.)[1,2,13,14]** Разработка спецификации на существующее приложение, например курсовой проект студента. Спецификация в виде сценариев использования. Спецификация в виде UML Use Case диаграмм.
- 2. Работа с баг-трекинговой системой, составление баг-репорта. Написание модульных тестов на собственный код студента.(4ч.)[1,2,7,8]**
- 3. Применение методологий TDD для разработки программного обеспечения {разработка проекта} (8ч.)[1,6]** Действия по индивидуальным заданиям, выполнение требований задания, фиксация изменений (коммит) в системе контроля версий git каждого шага разработки (тест, код)
- 4. Анализ программного обеспечения (статический и динамический).(4ч.)[1,15,16,17]**
- 5. Функциональное тестирование web API.(4ч.)[1,2,12]** Работа в среде Postman и взаимодействие с API для авторизации, аутентификации, получения данных, их обновления, добавления, и удаления.

#### **Самостоятельная работа (132ч.)**

- 1. Освоение учебной литературы по тестированию {разработка проекта} (60ч.)[1,2]**
- 2. Изучение литературы в виде пособия, а также предложенных публикаций и видео с конференций с целью изучения существующих методов тестирования, повышения качества и надёжности ПО.(40ч.)[1,6,7,9,10,11,12,15,16]**
- 3. Рефакторинг (улучшение) своего кода при выполнении лабораторных работ {разработка проекта} (16ч.)[1]** После изучения нового материала, предлагать и осуществлять рефакторинг ранее написанного ПО, так, чтобы оно было тестируемым, в том числе, для подготовки к экзамену.
- 4. Подготовка к экзамену(16ч.)[1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17]**
- 5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине**

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронно-библиотечным системам: Лань, Университетская библиотека он-лайн, электронной библиотеке АлтГТУ и к электронной

информационно-образовательной среде:

1. Старолетов С.М. Основы тестирования и верификации программного обеспечения [Электронный ресурс]: Учебное пособие.— Электрон. дан.—

Барнаул:

АлтГТУ,

2020.—

Режим

доступа:

[http://elib.altstu.ru/eum/download/pm/Staroletov\\_OsnTestVerifPO\\_up.pdf](http://elib.altstu.ru/eum/download/pm/Staroletov_OsnTestVerifPO_up.pdf),

авторизованный

## **6. Перечень учебной литературы**

### **6.1. Основная литература**

2. Старолетов С. М. Автоматное моделирование многокомпонентных программных систем, их тестирование и верификация: учебно-методическое пособие/ С. М. Старолетов, Е. Н. Крючкова. – Барнаул : Изд-во АлтГТУ, 2015. –

156 с. — Режим

доступа:

[http://new.elib.altstu.ru/eum/download/pm/avtomat\\_staroletov.pdf](http://new.elib.altstu.ru/eum/download/pm/avtomat_staroletov.pdf)

### **6.2. Дополнительная литература**

3. Липаев, В. В. Тестирование компонентов и комплексов программ : учебник / В. В. Липаев. — Москва : СИНТЕГ, 2010. — 393 с. — ISBN 978-5-89638-115-0. — Текст : электронный // Цифровой образовательный ресурс IPR

SMART : [сайт]. — URL: <https://www.iprbookshop.ru/27301.html> (дата обращения:

05.05.2023). — Режим доступа: для авторизир. пользователей

## **7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

4. Selenium - Web Browser Automation. — Режим доступа: <https://www.seleniumhq.org>

5. Doxygen. Generate documentation from source code. — Режим доступа: <http://www.doxygen.nl>

6. TDD в геймдеве или «кроличий ад». — Режим доступа: <https://habr.com/ru/companies/pixonic/articles/455076/>

7. Почему юнит-тесты не работают в научных приложениях. — Режим доступа: <https://habr.com/ru/articles/92038/>

8. Анатомия юнит-теста. — Режим доступа: <https://habr.com/ru/articles/554808/>

9. Автотесты в World of Tanks: боты на страже качества. — Режим доступа: <https://www.youtube.com/watch?v=eyJVEmuxFL4>

10. Тестирование десктоп-приложений с использованием нейронных сетей. — Режим доступа: <https://www.youtube.com/watch?v=GNb-C4qYSQw>

11. Fuzzing-тестирование: ищем баги в JIT-компиляторе и не только. — Режим

доступа: [https://www.youtube.com/watch?v=S2vBz\\_ZPAbc](https://www.youtube.com/watch?v=S2vBz_ZPAbc)

12. Postman Beginner's Course - API Testing. — Режим доступа: <https://www.youtube.com/watch?v=VywxIQ2ZXw4>

13. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. — Режим доступа: <https://habr.com/ru/articles/566218/>

14. UML Use Case Diagrams. — Режим доступа: <https://www.uml-diagrams.org/use-case-diagrams.html>

15. Топ-10 ошибок, найденных PVS-Studio в проектах на ASP.NET Core. — Режим доступа: <https://habr.com/ru/companies/pvs-studio/articles/654395/>

16. Cppcheck is a static analysis tool for C/C++ code. — Режим доступа: <http://cppcheck.sourceforge.net>

17. Valgrind is an instrumentation framework for building dynamic analysis tools. — Режим доступа: <http://www.valgrind.org>

## **8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации**

Содержание промежуточной аттестации раскрывается в комплекте контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

Фонд оценочных материалов (ФОМ) по дисциплине представлен в приложении А.

## **9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное взаимодействие обучающегося с преподавателем через личный кабинет студента.

<b>№пп</b>	<b>Используемое программное обеспечение</b>
1	Dev-C++
2	Dia
3	Eclipse IDE
4	LibreOffice
5	Selenium IDE
6	Selenium WebDriver
7	Windows
8	Антивирус Kaspersky

№пп	Используемые профессиональные базы данных и информационные справочные системы
1	IEEE Xplore - Интернет библиотека с доступом к реферативным и полнотекстовым статьям и материалам конференций. Бессрочно без подписки ( <a href="https://ieeexplore.ieee.org/Xplore/home.jsp">https://ieeexplore.ieee.org/Xplore/home.jsp</a> )
2	Бесплатная электронная библиотека онлайн "Единое окно к образовательным ресурсам" для студентов и преподавателей; каталог ссылок на образовательные интернет-ресурсы ( <a href="http://Window.edu.ru">http://Window.edu.ru</a> )
3	Крупнейший веб-сервис IT-проектов и их совместной разработки ( <a href="https://github.com/">https://github.com/</a> )
4	На сайте проекта OpenNet размещается информация о Unix системах и открытых технологиях для администраторов, программистов и пользователей ( <a href="http://www.opennet.ru/">http://www.opennet.ru/</a> )
5	Национальная электронная библиотека (НЭБ) — свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. ( <a href="http://нэб.рф/">http://нэб.рф/</a> )

## 10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Наименование специальных помещений и помещений для самостоятельной работы
учебные аудитории для проведения учебных занятий
помещения для самостоятельной работы

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».