

Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет»

УТВЕРЖДАЮ

Первый проректор Белорусско-Российского  
университета

  
О.В. Машин

20.10 2023

Регистрационный № УД-090301/Б.р.О.7 /1

**ПРАКТИКА НАПИСАНИЯ ПРОГРАММНОГО КОДА**  
(наименование дисциплины)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Направление подготовки** 09.03.01 Информатика и вычислительная техника

**Направленность (профиль)** Разработка программно-информационных систем

**Квалификация** Бакалавр

	Форма обучения
	Очная
Курс	2
Семестр	3
Лекции	16
Лабораторные работы, часы	34
Экзамен, семестр	3
Курсовая работа	3
Контактная работа по учебным занятиям, часы	50
Самостоятельная работа, часы	94
Всего часов / зачетных единиц	144/4

Кафедра-разработчик программы: Программное обеспечение информационных технологий  
(название кафедры)

Составитель: О.В. Сергиенко, ст. препод., Н.Н. Горбатенко, канд. техн. наук, доц.,  
Прокопенко С.А., ст. препод.

(И.О. Фамилия, ученая степень, ученое звание)

Рабочая программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, учебным планом Рег. № 090301-2.1 от 28.04.2023.

Рассмотрена и рекомендована к утверждению в качестве рабочего варианта на заседании кафедры «Программное обеспечение информационных технологий» 20.09.2023, протокол № 2.


Зав. кафедрой ПОИТ

 В. В. Кутузов

Одобрена и рекомендована к утверждению Научно-методическим советом Белорусско-Российского университета

18 октября 2023 г., протокол № 2.

Зам. председателя  
Научно-методического совета

 С.А. Сухоцкий

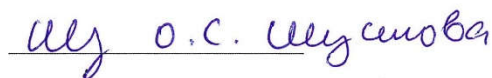
Рецензент:

Акиншева И. В., заведующий кафедрой программного обеспечения информационных технологий МГУ имени А. А. Кулешова, канд. техн. наук

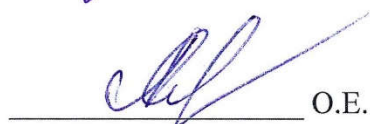
(И.О. Фамилия, должность, ученая степень, ученое звание рецензента)

Рабочая программа согласована:

Ведущий библиотекарь

 О.С. Шустова

Начальник учебно-методического  
отдела

 О.Е. Печковская

# 1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## 1.1 Цель учебной дисциплины

Целью учебной дисциплины является приобретение студентами навыков написания программного кода на основе принципов процедурной декомпозиции решаемой задачи.

## 1.2 Планируемые результаты изучения дисциплины

В результате освоения учебной дисциплины студент должен

### **знать:**

- основы алгоритмизации;
- технологию разработки программного обеспечения на основе принципов процедурной декомпозиции;
- основные конструкции языка высокого уровня;
- основные приемы разработки, отладки, тестирования и документирования программы.

### **уметь:**

- составлять алгоритмы;
- выполнять процедурную декомпозицию решаемой задачи;
- реализовывать алгоритмы в виде программ на языке высокого уровня;

### **владеть:**

- методами процедурной декомпозиции решаемой задачи;
- навыками самостоятельной разработки, отладки, тестирования и документирования программы.

## 1.3 Место учебной дисциплины в системе подготовки студента

Дисциплина относится к Блоку 1 "Дисциплины (модули)", обязательная часть блока 1.

Перечень учебных дисциплин, изучаемых ранее, усвоение которых необходимо для изучения данной дисциплины:

- Программирование;
- Информатика.

Перечень учебных дисциплин, которые будут опираться на данную дисциплину:

- Объектно-ориентированное программирование.

Кроме того, знания, полученные при изучении дисциплины на лекционных и лабораторных занятиях будут применены при прохождении преддипломной практики, а также при подготовке выпускной квалификационной работы и дальнейшей профессиональной деятельности.

## 1.4 Требования к освоению учебной дисциплины

Освоение данной учебной дисциплины должно обеспечивать формирование следующих компетенций:

Коды формируемых компетенций	Наименования формируемых компетенций
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК-9	Способен осваивать методики использования программных средств для решения практических задач

## 2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Вклад дисциплины в формирование результатов обучения выпускника (компетенций) и достижение обобщенных результатов обучения происходит путём освоения содержания обучения и достижения частных результатов обучения, описанных в данном разделе.

### 2.1 Содержание учебной дисциплины

Номер тем	Наименование тем	Содержание	Коды формируемых компетенций
1	Система контроля версий Git	<p>Понятие системы контроля версий как как системы, регистрирующей изменения в одном или нескольких файлах, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.</p> <p>Виды систем контроля версий. Локальные. Централизованные. Распределенные.</p> <p>Понятие Git как распределенной системы контроля версий для отслеживания изменений в файлах и координации работы над этими файлами среди большой команды людей. Преимущества использования Git.</p> <p>Git. Терминология. Репозиторий. Fork. Clone. Ветка/Branch. Master. Commit. Pull. Push. Merge. Pull request. Code review. Обновиться из upstream. Обновиться из origin.</p> <p>Установка Git. Создание профиля в репозитории GitHub.</p> <p>Git. Настройка конфигурации.</p> <p>Основные команды Git. git init. git status. git add. git commit. git diff. git log. git show. git branch. git checkout. git merge. git rebase. git remote add. git pull origin. git push. git fetch origin. git revert. git reset.</p> <p>Конфликты в работе Git и их разрешение. git commit –m. git merge –abort.</p> <p>Работа с Git в среде Visual studio.</p>	ОПК-8 ОПК-9
2	Рефакторинг кода	<p>Понятие рефакторинга программного кода как процесса изменения кода, не меняя его функциональности.</p> <p>Проблемы в коде, требующие рефакторинга. Дублирование кода. Длинный метод. Большой класс. Длинный список параметров. «Жадные» функции. Избыточные временные переменные. Классы данных. Не сгруппированные данные.</p> <p>Методы рефакторинга. Инкапсуляция поля (Encapsulate Field). Выделение класса (Extract Class). Выделение интерфейса (Extract Interface). Выделение локальной переменной (Extract Local Variable). Выделение метода (Extract Method). Генерализация типа (Generalize Type). Встраивание (Inline). Введение фабрики (Introduce Factory). Введение параметра (Introduce Parameter). Подъем поля/метода (Pull Up). Спуск поля/метода (Push Down). Замена условного оператора полиморфизмом (Replacing a conditional operator with a polymorphism).</p> <p>Инструментальные средства Visual studio для выполнения рефакторинга кода.</p>	ОПК-8 ОПК-9
3	Отладка и профилирование кода в среде Visual Studio	<p>Назначение отладчика кода. Основные приемы работы с отладчиком кода.</p> <p>Профилирование кода в среде Visual Studio. Основные приемы работы с профилировщиком кода.</p>	ОПК-8 ОПК-9
4	Разработка кода через тестирование	<p>Сущность разработки кода через тестирование (Test-Driven Development). Преимущества TDD. Главные принципы TDD.</p> <p>Назначение unit-тестов. Создание unit-тестов в Visual Studio.</p>	ОПК-8 ОПК-9
5	Соглашения и стандарты по стилю кода C#-программ	<p>Общие соглашения о коде C#. Рекомендации по именованию. Рекомендации по использованию управляющих инструкций. Рекомендации по использованию методов. Рекомендации по использованию параметров. Рекомендации по проектированию методов. Рекомендации по использованию исключений.</p>	ОПК-8 ОПК-9

		Рекомендации по использованию комментариев.	
6	Стратегии оптимизации кода	Понятие оптимизации кода как процесса улучшения производительности и эффективности программы кода. Виды стратегий оптимизации кода. Использование эффективных алгоритмов. Уменьшение количества операций. Использование более эффективных структур данных. Избегание избыточных проверок. Использование локальных переменных. Использование компилятора с оптимизацией. Использование кэшей. Использование алгоритмов с меньшей сложностью. Уменьшение числа обращений к памяти. Оптимизация циклов.	ОПК-8 ОПК-9
7	Проектирование приложения при процедурном стиле программирования	Сущность процедурного стиля программирования. Основные этапы проектирования приложения при процедурном стиле программирования. Анализ предметной области. Определение требований к программе (функциональные, нефункциональные, требования к качеству и безопасности). Определение архитектуры программы (выбор платформы, языка программирования, базы данных). Разработка структурной схемы программы методом нисходящего проектирования с использованием стратегии пошагового уточнения на основе требований к программе. Разработка интерфейса пользователя. Разработка тест-плана для проверки программы на соответствие требованиям.	ОПК-8 ОПК-9

## 2.2 Учебно-методическая карта учебной дисциплины

№ недели	Лекции (наименование тем)	Часы	Практические (семинарские) занятия	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
Семестр 3									
Модуль 1									
1	Тема 1. Система контроля версий Git	2			Л.р.№ 1. Практика написания методов	2	1		
2					Л.р.№ 1. Практика написания методов	2	1	ЗЛР	6
3	Тема 1. Система контроля версий Git	2			Л.р.№ 2. Обработка исключительных ситуаций	2	1	ЗЛР	6
4					Л.р.№ 3. Форматирование кода	2	1		
5	Тема 2. Рефакторинг кода	2			Л.р.№ 4. Написание комментариев	2	1	ЗЛР	6
6					Л.р.№ 5. Отладка и тестирование программного кода	2	1		
7	Тема 3. Отладка и профилирование кода в среде Visual Studio	2			Л.р.№ 6. Применение стратегий оптимизации кода	2	1	ЗЛР	6
8					Л.р.№ 7. Применение методик оптимизации логики кода	2	1	ЗЛР ПКУ	6 30
Модуль 2									
9	Тема 4. Разработка кода через тестирование	2			Л.р.№ 8. Решение и тестирование задач	2	1	ЗЛР	4
10					Л.р.№ 8. Решение и тестирование задач	2	1	ЗЛР	4
11	Тема 5. Соглашения и стандарты по стилю кода C#-программ	2			Л.р.№ 9. Выбор оптимальной стратегии и решение задачи	2	1	ЗЛР	4
12					Л.р.№ 9. Выбор оптимальной стратегии и решение задачи	2	1	ЗЛР	4
13	Тема 6. Стратегии оптимизации кода	2			Л.р.№ 10. Анализ условия, постановка задания и решение задачи	2	2		

14				Л.р.№ 10. Анализ условия, постановка задания и решение задачи	2	2	ЗЛР	4
15	Тема 7. Проектирование приложения при процедурном стиле программирования	2		Л.р.№ 11. Решение комплексной задачи	2	2	ЗЛР	5
16				Л.р.№ 11. Решение комплексной задачи	2	2		
17				Л.р.№ 11. Решение комплексной задачи	2	2	ЗЛР ПКУ	5 30
1-17	Выполнение курсовой работы					36		
18-20						36	ПА (экзамен.)	40
Итого		16			34	94		100

Принятые обозначения:

*Текущий контроль* –

ЗЛР – защита лабораторной работы;

ПКУ – промежуточный контроль успеваемости.

ПА - Промежуточная аттестация.

Итоговая оценка определяется как сумма текущего контроля и промежуточной аттестации и соответствует баллам:

Экзамен

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

### 2.3 Требования к курсовой работе

Целями курсовой работы являются:

- углубление, закрепление и систематизация теоретических знаний и практических умений, полученных в процессе обучения;
- оценка степени усвоения студентами основных положений дисциплины;
- приобретение навыков самостоятельной разработки программного обеспечения в интегрированной среде Microsoft Visual Studio.NET;
- получение навыков оформления программной документации и использования справочной литературы, ГОСТов при создании программных продуктов.

Содержание курсового проекта включает две части:

- 1) теоретическая часть – обзор по теме проектирования, исследование актуальных вопросов в данной области, постановка задач, обоснование принятого решения;
- 2) практическая – реализация программной системы и ее отладка.

Примерная тематика курсовых проектов представлена в приложении 1.

Разбивка этапов курсовой работы, определение количества минимальных и максимальных баллов за каждый из них производится преподавателем. Примерный перечень этапов выполнения курсовой работы и количества баллов за каждый из них представлен в таблице.

№	Этап выполнения	Минимум	Максимум
1	Описание предметной области	6	10
2	Проектирование программного модуля	6	10
3	Разработка документации программного модуля	9	15
4	Разработка программного модуля	9	15
5	Тестирование программного модуля	6	10
	Итого за выполнение курсового проекта	36	60
	Защита курсового проекта	15	40

Итоговая оценка курсового проекта представляет собой сумму баллов за его выполнение и защиту и выставляется в соответствии со шкалой:

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Баллы	87-100	65-86	51-64	0-50

### 3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изучении дисциплины используется модульно-рейтинговая система оценки знаний студентов. Применение форм и методов проведения занятий при изучении различных тем курса представлено в таблице.

№ п/п	Форма проведения занятия	Вид аудиторных занятий			Всего часов
		Лекции	Практические занятия	Лабораторные занятия	
2	Мультимедиа	Темы 1-7			16
7	С использованием ЭВМ			Л.р.№№ 1-11	34
	<b>ИТОГО</b>				50

### 4 ОЦЕНОЧНЫЕ СРЕДСТВА

Используемые оценочные средства по учебной дисциплине представлены в таблице и хранятся на кафедре.

№ п/п	Вид оценочных средств	Количество комплектов
1	Вопросы к экзамену	1
2	Экзаменационные билеты	1
3	Задания к защите лабораторных работ	11
4	Вопросы к самостоятельной подготовке и к лабораторным работам	11

### 5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

#### 5.1 Уровни сформированности компетенций

№ п/п	Уровни сформированности компетенции	Содержательное описание уровня*	Результаты обучения**
	<i>ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения</i>		
	<i>ИОПК-8.2. Способен создавать и отлаживать программный код, пригодный для дальнейшего использования.</i>		
1	Пороговый уровень	Способен создавать программный код, отвечающий требованию задания, под непосредственным руководством преподавателя.	Знать и понимать основные практики написания программного кода.
2	Продвинутый уровень	Способен создать программный код,	Уметь использовать основные практики

		отвечающий требованию задания, изредка пользуясь консультацией преподавателя.	написания программного кода.
3	Высокий уровень	Способен самостоятельно создавать программный код, отвечающий требованию задания.	Уметь оценивать эффективность применения различных практик написания программного кода.
<i>ОПК-9 Способен осваивать методики использования программных средств для решения практических задач</i>			
<i>ИОПК-9.2. Способен применять современные методики разработки программного обеспечения</i>			
1	Пороговый уровень	Понимает основные концепции и методики разработки программного обеспечения	Знать и понимать основные концепции и методики разработки программного обеспечения
2	Продвинутый уровень	Умеет использовать на практике основные концепции и методики разработки программного обеспечения	Уметь использовать основные концепции и методики, связанные с разработкой программного обеспечения
3	Высокий уровень	Умеет оценивать результаты применения основных концепций, и методик разработки программного обеспечения	Уметь оценивать результаты применения основных концепций и методик разработки программного обеспечения

## 5.2 Методика оценки знаний, умений и навыков студентов

Результаты обучения	Оценочные средства
<i>Компетенция ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения</i>	
Пороговый уровень. Знать и понимать основные практики написания программного кода.	Контрольные вопросы к защите лабораторных работ Вопросы к зачету
Продвинутый уровень Уметь использовать основные практики написания программного кода.	Контрольные вопросы к защите лабораторных работ Вопросы к зачету
Высокий уровень Уметь оценивать эффективность применения различных практик написания программного кода.	Контрольные вопросы к защите лабораторных работ Вопросы к зачету
<i>Компетенция ОПК-9. Способен осваивать методики использования программных средств для решения практических задач</i>	
Пороговый уровень. Знать и понимать основные концепции и методики разработки программного обеспечения	Контрольные вопросы к защите лабораторных работ Вопросы к зачету



Продвинутый уровень Уметь использовать основные концепции и методики, связанные с разработкой программного обеспечения	Контрольные вопросы к защите лабораторных работ Вопросы к зачету
Высокий уровень Уметь оценивать результаты применения основных концепций и методик разработки программного обеспечения	Контрольные вопросы к защите лабораторных работ Вопросы к зачету

### 5.3 Критерии оценки лабораторных работ

Оценка знаний студентом материала каждой лабораторной работы осуществляется путём защиты им отчёта, где должны быть сформулирована цель работы, условие решаемой задачи, описан алгоритм решения задачи, приведен код программы, скриншот с результатами решения задачи, сделано заключение. При защите студент должен ответить на контрольные вопросы к лабораторной работе. Максимальное количество баллов, указанное в таблице 2.2 в колонке Баллы (max), студент получает, ответив на 100 % контрольных вопросов. Минимальное количество баллов, равное  $0,5 \cdot \text{Баллы}(\text{max})$ , студент получает, ответив не менее чем на 50 % контрольных вопросов. Остальная шкала баллов соответствует правильным ответам на вопросы пропорционально их количеству и сложности.

### 5.4 Критерии оценки зачета

Зачет по данной дисциплине проводится индивидуально (возможно использование информационно-коммуникационных технологий). Студенту предлагается за определённое время ответить на ряд вопросов, охватывающих все изученные темы. При ответе на каждый вопрос студент должен выбрать правильный ответ из нескольких предлагаемых или сформулировать собственный ответ, если зачет проводится в устной форме. Ответив не менее чем на 50 % поставленных вопросов, студент получает 15 баллов. Максимальное количество баллов студент получает, ответив на 100 % поставленных вопросов. Остальная шкала баллов соответствует правильным ответам на вопросы пропорционально их количеству и сложности.

## 6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Самостоятельная работа студентов (СРС) направлена на закрепление и углубление освоения учебного материала, развитие практических умений. СРС включает следующие виды самостоятельной работы студентов:

- самостоятельное изучение материала по учебникам и другим источникам;
- тестирование по предмету и выполнение контрольных работ;
- обзор литературы;
- закрепление изученного материала на групповых занятиях;
- работа со справочной литературой;
- подготовка к аудиторным занятиям;
- подготовка к сдаче зачета.

Подготовка к тестированию и написанию контрольной работы по соответствующему модулю дисциплины подразумевает изучение лекционного материала и выполнение практических работ, относящихся к соответствующему модулю.

Контроль результатов внеаудиторной самостоятельной работы студентов осуществляется в пределах времени, отведенного на обязательные учебные занятия по

дисциплине и внеаудиторную самостоятельную работу студентов по дисциплине, проходит в письменной форме.

Критериями оценки результатов внеаудиторной самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умение студента использовать теоретические знания при выполнении практических, творческих заданий;
- обоснованность и четкость изложения ответа.

Перечень контрольных вопросов и заданий для самостоятельной работы студентов хранится на кафедре.

Для СРС рекомендуется использовать источники, приведенные в п. 7.

## **7 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **7.1 Основная литература**

№ п/п	Библиографическое описание	Гриф	Количество экземпляров
1	С#. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2010. - 432с. - (Учебник для вузов).	Доп. МО и науки РФ в качестве учебника для студентов вузов	9

### **7.2 Дополнительная литература**

№ п/п	Библиографическое описание	Гриф	Количество экземпляров
1	Языки программирования: лабораторный практикум для вузов / П. В. Казаков. - Брянск: БГТУ, 2011. - 84с. - (Информационные системы и технологии).	Утв. РИС в качестве лабораторного практикума для студ. вузов	1
2	С# 2005 для профессионалов: [Пер. с англ.] / Нейгел Кристиан, Иввен Билл, Глинн Джей и др. - М.: Диалектика, 2007. - 1376с.	—	2
3	С#. Начала программирования: учебник / Э. А. Ишкова. - М.: Бином-Пресс, 2010. - 336с.	—	1

### **7.3 Перечень ресурсов сети Интернет по изучаемой дисциплине**

[https://professorweb.ru/my/csharp/charp\\_theory/level1/index.php](https://professorweb.ru/my/csharp/charp_theory/level1/index.php)

**7.4 Перечень наглядных и других пособий, методических рекомендаций по проведению учебных занятий, а также методических материалов к используемым в образовательном процессе техническим средствам**

#### **7.4.1 Методические рекомендации**

1 Методические рекомендации к выполнению лабораторных работ по дисциплине «Практика написания программного кода» / Сост. Сергиенко О. В., Горбатенко Н.Н. БРУ, Могилев, 2022 г., 30 с.

#### **7.4.2 Информационные технологии**

**Мультимедийные презентации по лекционному курсу:**

Тема 1. Система контроля версий Git

- Тема 2. Рефакторинг кода
- Тема 3. Отладка и профилирование кода в среде Visual Studio
- Тема 4. Разработка кода через тестирование
- Тема 5. Соглашения и стандарты по стилю кода C#-программ
- Тема 6. Стратегии оптимизации кода
- Тема 7. Проектирование приложения при процедурном стиле программирования

#### **7.4.3 Перечень программного обеспечения, используемого в образовательном процессе**

Свободно распространяемая система программирования MS Visual Studio .NET 2018 и выше. (Лабораторные работы № 1-12)

### **8 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Материально-техническое обеспечение дисциплины содержится в паспорте лаборатории « а. 517/2», рег. номер ПУЛ-4/517.2-23 , « а. 518/2». рег. номер ПУЛ-4/518.2-23 , «а. 519/2 », рег, номер ПУЛ-4/519.2-23

## ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

по учебной дисциплине Практика написания программного кода

направление подготовки 09.03.01 Информатика и вычислительная техника

направленность (профиль) Автоматизированные системы обработки информации и управления

на 2024/2025 учебный год

№№ пп	Дополнения и изменения	Основание
1	7.4.1 Методические рекомендации считать в новой редакции: 1. Практика написания программного кода: методические рекомендации к лабораторным работам предназначены для студентов направлений подготовки 09.03.04 «Программная инженерия», 09.03.01 «Информатика и вычислительная техника» очной формы обучения / О. В. Сергиенко, Н. Н. Горбатенко – Могилев: Белорусско-Российский университет, 2023. – 32 с.	Издание новых методических рекомендаций в соответствии со сводным планом изданий университета на 2023 год.

Рабочая программа пересмотрена и одобрена на заседании кафедры  
«Программное обеспечение информационных технологий»

(название кафедры-разработчика программы)

(протокол № 8 от 29 02 2024)

Заведующий кафедрой  
канд. техн. наук., доцент  
(ученая степень, ученое звание)

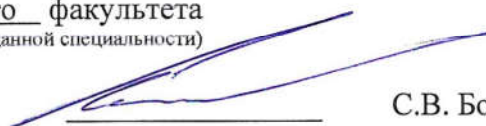


В.В. Кутузов

УТВЕРЖДАЮ

Декан электротехнического факультета  
(название факультета, выпускающего по данной специальности)

канд. техн. наук., доцент  
(ученая степень, ученое звание)



С.В. Болотов

16, 04 2024

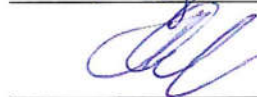
СОГЛАСОВАНО:

Ведущий библиотекарь



О.С. Шустова

Начальник учебно-методического  
отдела



О.Е. Печковская

16, 04 2024