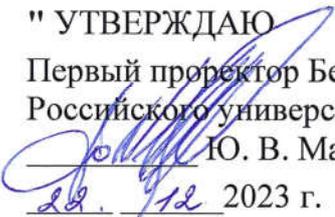


Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

" УТВЕРЖДАЮ

Первый проректор Белорусско-
Российского университета

 Ю. В. Машин

22. 12 2023 г.

Регистрационный № УД-010304/Б.Р.В.Ч/р.

ПРАКТИКИ НАПИСАНИЯ ПРОГРАММНОГО КОДА

РАБОЧАЯ ПРОГРАММА

Направление подготовки 01.03.04 Прикладная математика

Направленность (профиль) Разработка программного обеспечения

Квалификация бакалавр

	Форма обучения
	Очная
Курс	3
Семестр	5
Лекции, часы	16
Лабораторные занятия, часы	34
Зачёт, семестр	5
Контактная работа по учебным занятиям, часы	50
Самостоятельная работа, часы	58
Всего часов / зачетных единиц	108/3

Кафедра-разработчик программы: Программное обеспечение информационных технологий

(название кафедры)

Составители: канд.техн.наук, доц. Н.Н. Горбатенко

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

Могилев, 2023

Рабочая программа составлена в соответствии с федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки 01.03.04 – “Прикладная математика” (уровень бакалавриата), утвержденным приказом № 11 от 10.01.2018 и учебным планом, утвержденным 28.04.2023 г., рег. номер 010304-2.1

Рассмотрена и рекомендована к утверждению кафедрой ПОИТ
6.12.2023 г., протокол № 5.

Зав. кафедрой  В.В. Кутузов

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

20 декабря 2023 г., протокол № 3.

Зам. председателя
Научно-методического совета

 С.А. Сухоцкий

Рецензент:

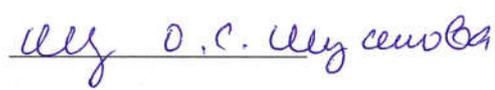
Акиншева И. В., заведующий кафедрой программного обеспечения информационных технологий МГУ имени А. А. Кулешова, канд. техн. наук

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

Рабочая программа согласована
Зав. каф. Высшей математики

 В.Г. Замураев

Ведущий библиотекарь

 О.С. Шушова

Начальник учебно-методического
отдела

 О.Е. Печковская

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель учебной дисциплины

Целью учебной дисциплины является ознакомление студентов с практиками написания программного кода на основе системы контроля версий, рефакторинга кода и паттернов (шаблонов программирования), а также с основными принципами их использования.

1.2 Задачи учебной дисциплины

Студент, изучивший дисциплину, должен **знать**:

- систему контроля версий Git;
- рефакторинг программного кода;
- шаблоны программирования.

Студент, изучивший дисциплину, должен **уметь**:

- использовать систему контроля версий Git;
- использовать рефакторинг кода;
- использовать основные паттерны при разработке программного обеспечения.

Студент, изучивший дисциплину, должен **владеть**:

- навыками использования системы контроля версий Git;
- навыками использования рефакторинга программного кода;
- навыками использования паттернов программирования.

1.3 Место дисциплины в системе подготовки студента

Дисциплина «Практики написания программного кода» относится к блоку 1 Дисциплины (модули) «Часть Блока 1, формируемая участниками образовательных отношений».

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

- программирование.

Перечень учебных дисциплин (циклов дисциплин), которые будут опираться на данную дисциплину:

- базы данных;
- проектирование программного обеспечения;
- тестирование и отладка программного обеспечения.

Кроме того, знания, полученные при изучении дисциплины на лекциях и лабораторных работах 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
ПК-3	Способен осуществлять непосредственное руководство процессами разработки программного обеспечения
ПК-4	Способен организовывать процессы разработки программного обеспечения

2 СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1 Содержание учебной дисциплины

Номер тем	Наименование тем	Содержание	Коды формируемых компет.
1	2	3	4
1	Система контроля версий Git	<p>Понятие системы контроля версий. 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>	ПК-3 ПК-4
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>	ПК-3 ПК-4

		<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>	
3	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>	ПК-3 ПК-4
4	Основные принципы объектно-ориентированного проектирования SOLID	<p>Принцип единственной обязанности (SRP).</p> <p>Принцип открытости/закрытости (OCP).</p> <p>Принцип подстановки Лисков (LSP).</p> <p>Принцип разделения интерфейсов (ISP).</p> <p>Принцип инверсии зависимости (DIP).</p>	ПК-3 ПК-4
5	Паттерны проектирования GoF	<p>Порождающие паттерны.</p> <p>Структурные паттерны.</p> <p>Паттерны поведения.</p>	ПК-3 ПК-4

2.2 Учебно-методическая карта учебной дисциплины

№ недели	Лекции (наименование тем)	Часы	Лабораторные занятия	Часы	Самостоятельная работа, часы	Форма контроля знаний	Баллы (max)
1	2	3	4	5	6	7	8
Семестр 5							
Модуль 1							
1	Тема 1. Система контроля версий Git	2	Л. Р. № 1 Система контроля версий Git	2	1	ЗЛР	3
2			Л. Р. № 1 Система контроля версий Git	2	1	ЗЛР	3
3	Тема 1. Система контроля версий Git	2	Л. Р. № 1 Система контроля версий Git	2	1	ЗЛР	3
4			Л. Р. № 2 Рефакторинг программного кода	2	1	ЗЛР	3

5	Тема 2. Рефакторинг кода		Л. Р. № 2 Рефакторинг программно-го кода			ЗЛР	3
6			Л. Р. № 2 Рефакторинг программно-го кода			ЗЛР	5
7	Тема 3. Разработка кода через тести-рование	2	Л. Р. № 3 Создание юнит-тестов	2	1	ЗЛР	5
8			Л. Р. № 3 Создание юнит-тестов	2	1	ТЗ ПКУ	5 30

Модуль 2

9	Тема 4. Основные принципы объ-ектно-ориентированного проекти-рования SOLID	2	Л. Р. № 4 Разработка программ с ис-пользованием принципов SOLID	2	1	ЗЛР	3
10		2	Л. Р. № 4 Разработка программ с ис-пользованием принципов SOLID	2	1	ЗЛР	3
11	Тема 4. Паттерны проектирования GoF		Л. Р. № 5 Разработка программ с ис-пользованием порождающих паттернов	2	1	ЗЛР	3
12		2	Л. Р. № 5 Разработка программ с ис-пользованием порождаю-щих паттернов	2	1	ЗЛР	3
13	Тема 4. Паттерны проектирования GoF		Л. Р. № 6 Разработка программ с ис-пользованием структурных паттернов.	2	1	ЗЛР	3
14		2	Л. Р. № 6 Разработка программ с ис-пользованием структурных паттернов.	2	2	ЗЛР	3
15	Тема 4. Паттерны проектирования GoF		Л. Р. № 7 Разработка программ с ис-пользованием паттернов по-ведения.	2	2	ЗЛР	3
16		2	Л. Р. № 7 Разработка программ с ис-пользованием паттернов по-ведения.	2	2	ЗЛР	4
17			Л. Р. № 7 Разработка программ с ис-пользованием паттернов поведения.	2	2	ТЗ ПКУ	5 30
17						ПА(зач)	40
Итого		16		34	22		100

Принятые обозначения:

ЗЛР – защита лабораторной работы;

ПКУ – промежуточный контроль успеваемости;

ПА – промежуточная аттестация.

ТЗ – тестовое задание

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

Зачет

Оценка	Зачтено	Не зачтено
Баллы	51-100	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 занятий	Виды аудиторных занятий		Всего часов
		Лекции	Лабораторные занятия	
1	Мультимедиа	Темы 1-5	–	16
2	С использованием ЭВМ	–	Лаб.р. №№ 1 – 8	34
	ИТОГО	16	34	50

4 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ, ПРОМЕЖУТОЧНОЙ И ИТОГОВОЙ АТТЕСТАЦИИ СТУДЕНТОВ

Оценочные средства контроля знаний студентов входят в состав учебно-методического комплекса дисциплины и хранятся на кафедре. Оценочные средства по дисциплине «Практики написания программного кода» включают:

№ п/п	Вид оценочных средств	Наличие (+/-)	Количество комплектов
1	Вопросы к зачету	+	1
2	Вопросы и задания к защите лабораторных работ	+	9
3	Тестовые задания	+	1

5 МЕТОДИКА И КРИТЕРИИ ОЦЕНКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

5.1 Уровни сформированности компетенций

№ п/п	Уровни сформированности компетенции	Содержательное описание уровня*	Результаты обучения**
-------	-------------------------------------	---------------------------------	-----------------------

<i>Компетенция ПК 3. Способен осуществлять непосредственное руководство процессами разработки программного обеспечения</i>			
<i>ПК-3.4 Способен использовать методологии разработки программного обеспечения и технологии программирования при руководстве разработкой программного кода</i>			
1	1 Пороговый уровень	Знает принципы разработки программного обеспечения	Знает принципы формализация и алгоритмизация поставленных задач и разработки технических спецификаций на программные компоненты
2	2 Продвинутый уровень	Способен осуществлять управление разработкой программных средств и проектов	Умеет проводить анализ требований к программному обеспечению, проверять и проводить отладку существующего программного кода
3	3 Высокий уровень	Умеет проводить анализ требований к программному обеспечению, проверять и проводить отладку существующего программного кода	Владеет навыками написания программного кода с использованием языков программирования, определения и манипулирования данными
<i>Компетенция ПК 4. Способен организовывать процессы разработки программного обеспечения.</i>			
<i>Индикатор ПК-4.2 Способен применять методологии разработки программного обеспечения при управлении информацией в процессе разработки программного обеспечения и при разработке внутренних правил, методик и регламентов проведения работ</i>			
	1 Пороговый уровень	Знает методологию разработки программного обеспечения на основе паттернов программирования.	Знает и понимает основные практики написания программного кода с использованием паттернов программирования.
	2 Продвинутый уровень	Умеет разрабатывать и тестировать программные компоненты с использованием паттернов программирования, пользуясь подсказкой преподавателя.	Умеет использовать основные паттерны программирования при написании программного кода.
	3 Высокий уровень	Умеет самостоятельно разрабатывать и тестировать программные компоненты с использованием паттернов программирования	Уметь оценивать эффективность применения различных паттернов при написании программного кода.

5.2 Методика оценки знаний, умений и навыков студентов

Результаты обучения	Оценочные средства
<i>Компетенция ПК 3. Способен осуществлять непосредственное руководство процессами разработки программного обеспечения</i>	
1 Пороговый уровень. Знает принципы формализация и алгоритмизация поставленных задач и разработки техни-	Вопросы к зачету Вопросы для защиты лабораторных работ

ческих спецификаций на программные компоненты	
2 Продвинутый уровень. Умеет проводить анализ требований к программному обеспечению, проверять и проводить отладку существующего программного кода	Вопросы к зачету Вопросы для защиты лабораторных работ Тестовые задания
3 Высокий уровень. Владеет навыками написания программного кода с использованием языков программирования, определения и манипулирования данными	Вопросы к зачету Вопросы для защиты лабораторных работ Тестовые задания
<i>Компетенция ПК 4. Способен организовывать процессы разработки программного обеспечения.</i>	
1 Пороговый уровень. Знает и понимает основные практики написания программного кода с использованием паттернов программирования.	Вопросы к зачету Вопросы для защиты лабораторных работ Тестовые задания
2 Продвинутый уровень. Умеет использовать основные паттерны программирования при написании программного кода.	Вопросы к зачету Вопросы для защиты лабораторных работ Тестовые задания
3 Высокий уровень. Уметь оценивать эффективность применения различных паттернов при написании программного кода.	Вопросы к зачету Вопросы для защиты лабораторных работ Тестовые задания

5.3 Критерии оценки лабораторных работ.

Каждая выполненная и защищенная лабораторная работа оцениваются в диапазоне от 3 до 5 баллов. При этом 2 балла начисляется за выполнение работы и от 1 до 3 баллов за оформление отчета и защиту работы в зависимости от качества оформления и уровня знаний студента по тематике работы. Если по окончании модуля лабораторная работа выполнена, но не защищена, то баллы по ней не начисляются и она попадает в разряд задолженности.

5.4 Критерии оценки тестового задания

Студент получает оценку «отлично» (5 баллов) если он правильно ответил на более 80% всех вопросов тестового задания.

Студент получает оценку «хорошо» (4 балла) если он правильно ответил не менее чем на 70% и не более 80% всех вопросов тестового задания.

Студент получает оценку «удовлетворительно» (3 балла) если он правильно ответил не менее чем на 50% и не более 70% всех вопросов тестового задания.

Студент получает оценку «неудовлетворительно» (2 балла) если он правильно ответил менее чем на 50% всех вопросов тестового задания.

5.5 Критерии оценки зачета.

На зачете знания, умения и навыки студентов оцениваются следующим образом:
«зачет» – если студент полно освоил учебный материал, владеет понятийным аппаратом, ориентируется в изученном материале, осознанно применяет знания для ре-

шения практических задач, грамотно излагает ответ, но содержание, форма ответа имеют отдельные неточности;

«не зачет» – если студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, искажающие их смысл, беспорядочно и неуверенно излагает материал, не может применять знания для решения практических задач.

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.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1 Основная литература

№ п/п	Автор, название, место издания, издательство, год издания учебной литературы	Гриф	Количество экземпляров
1	Макконнелл С. Совершенный код. Мастер-класс: пер. с англ. / С. Макконнелл. - М.: Русская редакция, 2020. - 896с.	–	15

7.2 Дополнительная литература

№ п/п	Автор, название, место издания, издательство, год издания учебной литературы	Гриф	Количество экземпляров
1	Йордан Эдвард. Объектно-ориентированный анализ и проектирование систем / Эдвард Йордан. - М.: ЛОРИ, 2007. – 264 с.	–	5
2	Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма и др. - СПб.: Питер, 2016. – 366 с.	–	5

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

7.3.1 Методические указания

1. Практики написания программного кода. Методические рекомендации к лабораторным работам для студентов по направлению подготовки 01.03.04 «Прикладная математика» дневной формы обучения / Составление Горбатенко Н.Н. Могилев, 2023 г., 48 с. (Электронный вариант)

7.3.2 Плакаты, мультимедийные презентации

Тема 1 – Система контроля версий Git
Тема 2 – Рефакторинг кода
Тема 3 - Принципы SOLID
Тема 4 - Паттерны проектирования GoF
Тема 5 - Шаблоны проектирования GRASP

7.3.3 Перечень программного обеспечения, используемого в учебном процессе

Используется свободно распространяемое программное обеспечение: система программирования MS Visual Studio 2019(Лабораторные работы № 1-8)

7.4 Материально-техническое обеспечение учебной дисциплины

Материально-техническое обеспечение дисциплины содержится в паспорте лаборатории «518/2», рег. номер № ПУЛ-4 518/2-23.