

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»
ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ
Кафедра Компьютерных технологий



УТВЕРЖДАЮ:

проректор по научно-методической
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ТЕСТИРОВАНИЕ И ВНЕДРЕНИЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль подготовки: Информатика и вычислительная техника

Образовательная программа: бакалавриат

Квалификация: академический бакалавр

Форма обучения: очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения

Донецк 2020

УТВЕРЖДАЮ:

Декан физико-технического факультета

Фоменко С.А.

«17» апреля 2020 г.



Программа учебной дисциплины **«Тестирование и внедрение программного обеспечения»** составлена на основе Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённого приказом Министерства образования и науки ДНР от «21» января 2016 г. №31»; «Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики», утверждённого приказом Министерства образования и науки ДНР №1171 от «10» ноября 2017 г.»; учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.01 Информатика и вычислительная техника, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Доцент кафедры компьютерных технологий

A handwritten signature in blue ink, likely belonging to T.V. Shariy.

Шарий Т.В.

Программа учебной дисциплины утверждена на заседании кафедры компьютерных технологий

Протокол № 12 от «2» апреля 2020 г.

Зав. кафедрой компьютерных технологий

A handwritten signature in blue ink, likely belonging to T.V. Ermolenko.

Ермоленко Т.В.

Программа учебной дисциплины одобрена учебно-методической комиссией физико-технического факультета

Протокол № 5 от «15» апреля 2020 г.

Председатель учебно-методической комиссии
физико-технического факультета

A handwritten signature in blue ink, likely belonging to V.N. Kotenko.

Котенко В.Н.

1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Тестирование и внедрение программного обеспечения» относится к вариативной части профессионального блока и состоит из двух содержательных модулей: модуль 1 – «Модульное тестирование», модуль 2 – «Интеграционное тестирование».

Основывается на базе дисциплин: «Основы программирования», «Программирование», «Технологии разработки программного обеспечения».

Является основой для изучения следующих дисциплин: «Выпускная квалификационная работа».

2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	09.03.01 Информатика и вычислительная техника			
Профиль	Информатика и вычислительная техника			
Образовательная программа	Бакалавриат			
Квалификация	Академический бакалавр			
Количество содержательных модулей	2			
Дисциплина базовой / вариативной части образовательной программы	Профессиональный блок. Вариативная часть			
Формы контроля (МК, экзамен, зачет)	Модульный контроль, зачет			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачётных единиц (кредитов)	2	2	3	3
Год подготовки	3	2	3	2
Семестр	6	4	6	4
Количество часов	72	72	72	72
- лекционных	16	16	4	4
- практических, семинарских				
- лабораторных	32	32	8	8
- самостоятельной работы	24	24	60	60
в т. ч. индивидуальное задание				
Недельное количество часов, т. ч.	4	4	4	4
аудиторных	3	3	0,75	0,75

3. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цель – формирование у студентов базовых знаний и навыков, касающихся современного процесса тестирования и внедрения программного обеспечения.

Задачи – изучение процесса разработки через тестирование; изучение паттернов модульного тестирования; изучение и применение на практике изоляционных фреймворков; изучение основ интеграционного тестирования; изучение основ тестирования производительности, внедрения программного обеспечения.

Требования к результатам освоения дисциплины:

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по данному направлению подготовки:

а) общекультурных (ОК):

способность работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия (ОК-6);

способность к самоорганизации и самообразованию (ОК-7).

б) общепрофессиональных (ОПК):

знание современных методов построения и анализа алгоритмов, основ численных методов и умение их использовать на практике (ОПК-4).

в) профессиональных (ПК):**проектно-конструкторская деятельность:**

пользоваться методиками использования программных средств для решения практических задач (ПК-2);

использовать и самостоятельно разрабатывать интерфейсы взаимодействия человека и ЭВМ (ПК-3);

знание принципов программирования, средств современных языков программирования, структур данных (ПК-5).

Проектно-технологическая деятельность:

знание современных технологий и инструментальных способов разработки сложных программных систем (инженерии программного обеспечения), умение их использовать на всех этапах жизненного цикла программ (ПК-14).

Научно-исследовательская деятельность:

базовые знания научно-методических основ и стандартов в области компьютерной инженерии, проводить эксперимент по проверке корректности решений, рассчитывать экономическую эффективность (ПК-15);

умение готовить и проводить доклады с использованием современных компьютерных средств, писать научно-технические отчеты, оформлять результаты исследований в виде статей (ПК-16).

Научно-педагогическая деятельность:

готовить конспекты лекций, проводить повышение квалификации сотрудников (ПК-17).

Сервисно-эксплуатационная деятельность:

инсталлировать, настраивать и сопровождать программное и аппаратное обеспечение информационных и автоматизированных систем (ПК-21).

В результате изучения учебной дисциплины студент должен**Знать:**

- принципы разработки приложений через тестирование (TDD, Test-Driven Development);
- основы и инструменты разработки модульных тестов; основы и области применения изоляционных фреймворков в модульном тестировании;
- технологии интеграционного тестирования;
- основы тестирования производительности приложений;
- основы внедрения программного обеспечения.

Уметь:

- писать модульные тесты (unit-тесты) для отдельных функций;
- проектировать и кодировать unit-тесты с заглушками, применять изоляционные фреймворки;
- писать тесты для web-приложений и приложений с графическим интерфейсом пользователя;
- производить функциональное и нагрузочное тестирование приложений, оформлять документацию по нему.

Владеть:

– навыками разработки тестов с помощью инструментов для модульного, интеграционного и нагрузочного тестирования.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Порядковый номер и тема	Краткое содержание темы
	Содержательный модуль 1. Модульное тестирование
Тема 1. Основные понятия этапа тестирования приложений.	Тест, дефект, верификация, валидация. Виды тестирования программного обеспечения. Функциональное тестирование.
Тема 2. Разработка через тестирование.	Методология «Разработка через тестирование»: Test Driven Development (TDD) и Behavior Driven Development (BDD).
Тема 3. Unit-тестирование	Основные понятия unit-тестирования. Паттерны и антипаттерны xUnit. Принцип «AAA» (Arrange-Act-Assert). Возможности MSTest, NUnit. Unit-тестирование в условиях случайности.
Тема 4. Изоляционные фреймворки.	Изоляционные фреймворки в unit-тестировании. Заглушки в unit-тестировании: стабы и моки. Библиотека Moq.
Тема 5. Тестирование в управлении IT-проектами	Гибкие методологии управления IT-проектами. Скрам. Спринт, бэклог, диаграмма выгорания.
	Содержательный модуль 2. Интеграционное тестирование
Тема 6. Интеграционное тестирование	Интеграционное тестирование. Тестирование графического интерфейса пользователя (Coded UI Tests).
Тема 7. Тестирование web-приложений	Тестирование web-приложений. Ядро Selenium WebDriver, браузерный компонент Selenium IDE.
Тема 8. Тестирование производительности	Тестирование производительности. Нагрузочное тестирование. Инструмент JMeter.
Тема 9. Внедрение программного обеспечения	Приемочное тестирование. Внедрение программного обеспечения. Виды установки ПО на клиентских компьютерах. Создание установщиков и развертывание программ с помощью программы InnoSetup и средств MS Visual Studio. Разворачивание приложения в облаке.
Тема 10. Тестирование и непрерывная интеграция	Автоматизация разработки, тестирования и управления проектами. Непрерывная интеграция (Continuous Integration, CI). MSBuild, NAnt, Jenkins, Travis CI.

Курс дисциплины «Тестирование и внедрение программного обеспечения» предусматривает следующие **формы организации учебного процесса**:

1. лекции;
2. лабораторные занятия;
3. самостоятельная работа студента.

По источнику передачи и восприятия учебной информации используются словесные (лекция, беседа), наглядные (слайды, иллюстрации, коды программ), практические (исследования, упражнения, лабораторные работы) методы.

По характеру познавательной деятельности студентов используются объяснительно-иллюстративные и репродуктивные методы, проблемное преподавание, частично-поисковый и исследовательский методы.

В зависимости от основной дидактической цели и задач используются методы устного изложения знаний, закрепление учебного материала, самостоятельной работы студентов по осмыслению и усвоению нового материала, работы по применению знаний на практике и выработке умений и навыков, проверки и оценки знаний, умений и навыков.

Используются следующие методы контроля:

1. устный контроль (экспресс-опрос на лекциях);
2. проверка конспектов;
3. защита лабораторных работ;
4. проверка самостоятельных работ;
5. модульная контрольная работа.

Тематический план

	Содержательный модуль 1																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма обучения												Заочная форма обучения										
	Нормативный срок обучения						Ускоренный срок обучения						Нормативный срок обучения						Ускоренный срок обучения				
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа
Тема 1. Основные понятия этапа тестирования приложений.	6	2		2	2		6	2		2	2		9	1		2	6		9	1		2	6
Тема 2. Разработка через тестирование.	5	1		2	2		5	1		2	2		6				6		6				6
Тема 3. Unit-тестирование	10	2		6	2		10	2		6	2		9	1		2	6		9	1		2	6
Тема 4. Изоляционные фреймворки.	10	2		6	2		10	2		6	2		9	1		2	6		9	1		2	6
Тема 5. Тестирование в управлении IT-проектами	5	1			4		5	1			4		6				6		6				6
Итого по 1-му содержательному модулю	36	8		16	12		36	8		16	12		39	3		6	30		39	3		6	30

	Содержательный модуль 2																							
Названия содержательных модулей и тем	Количество часов																							
	Очная форма обучения											Заочная форма обучения												
	Нормативный срок обучения					Ускоренный срок обучения					Нормативный срок обучения					Ускоренный срок обучения								
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.				
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа	
Тема 6. Интеграционное тестирование	8	2		4	2		8	2		4	2		6				6		6				6	
Тема 7. Тестирование web-приложений	10	2		6	2		10	2		6	2		9	1		2	6		9	1		2	6	
Тема 8. Тестирование производительности	8	2		4	2		8	2		4	2		6				6		6				6	
Тема 9. Внедрение программного обеспечения	5	1		2	2		5	1		2	2		6				6		6				6	
Тема 10. Тестирование и непрерывная интеграция	5	1			4		5	1			4		6				6		6				6	
Итого по 2-му содержательному модулю	36	8		16	12		36	8		16	12		33	1		2	30		33	1		2	30	
Всего часов	72	16		32	24		72	16		32	24		72	4		8	60		72	4		8	60	

5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
1.	Разработка через тестирование	2
2.	Unit-тестирование	4
3.	Изоляционные фреймворки	2
4.	Интеграционное тестирование	2
5.	Тестирование web-приложений	2
6.	Тестирование производительности	2
7.	Внедрение программного обеспечения	2
	ВСЕГО	16

Темы лабораторных занятий

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
1.	Юнит-тестирование с помощью MS Testing Framework и NUnit	10
2.	Заглушки в юнит-тестировании. Изоляционные фреймворки	6
3.	Интеграционное тестирование. Coded UI Tests и тестирование web-приложений	10
4.	Нагрузочное тестирование. Внедрение программного обеспечения	6
	ВСЕГО	32

6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов.

Самостоятельная работа студентов по курсу «Тестирование и внедрение программного обеспечения» предусматривает изучение дополнительной технической литературы и интернет-источников, рекомендуемых этой программой; самостоятельную разработку алгоритмов и текстов программ лабораторных работ, изучение дополнительного инструментария. При желании студент может подготовить реферат или доклад по одной из приведенных ниже тем:

1. Паттерны и антипаттерны unit-тестирования.
2. Изоляционные фреймворки.
3. Тестирование web-приложений с помощью Selenium.
4. Coded UI Tests (тестирование приложений с графическим интерфейсом).
5. Нагрузочное тестирование.
6. Тестирование и непрерывная интеграция.

№ n/n	Название темы	Количество часов
1.	Основные понятия этапа тестирования приложений	2
2.	Разработка через тестирование	2
3.	Unit-тестирование	2
4.	Изоляционные фреймворки	2
5.	Тестирование в управлении IT-проектами	4
6.	Интеграционное тестирование	2
7.	Тестирование web-приложений	2
8.	Тестирование производительности	2
9.	Внедрение программного обеспечения	2
10.	Тестирование и непрерывная интеграция	4
	ВСЕГО	24

7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальные задания предусмотрены к каждой лабораторной работе. Полный перечень всех заданий приведен в УМКД дисциплины.

Образец индивидуального задания

(Из лабораторной работы №1). Написать тесты с помощью NUnit для решения задачи, в соответствии с вариантом, с Assert'ами простых условий и исключений.

Вариант 1.

В Production Code написать класс Triangle с методом double Area(), который вычисляет площадь треугольника. Стороны задаются в методе SetSides(double a, double b, double c), при этом если одна из сторон задана отрицательной, бросается исключение FormatException; если стороны не могут образовать треугольник, то бросается исключение ArgumentException. В Test Code написать тесты, которые проверяют соответствие результатов метода спецификации требований.

Вариант 2.

В Production Code написать класс StringFormatter с методом string SafeString(string s), который во входной строке экранирует все одинарные кавычки (дублирует их для защиты от sql-инъекций в Postgre), а если строка пустая, то метод возвращает строку «NULL». В Test Code написать тесты, которые проверяют корректность работы метода.

Вариант 3.

В Production Code написать класс ArrayProcessor со статическим методом int[] SortAndFilter(int[] a), который сортирует входной массив целых чисел и удаляет из него все элементы, кроме двузначных. В Test Code написать тесты, проверяющие соответствие работы метода спецификации требований.

8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. Основные понятия этапа тестирования программного обеспечения.
2. Виды тестирования.
3. Функциональное тестирование.
4. Разработка через тестирование.
5. Unit-тестирование.
6. Библиотеки NUnit, MSTest.
7. Заглушки в unit-тестировании. Моки и стабы.
8. Изоляционные фреймворки.

9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «Донецкий национальный университет»
Физико-технический факультет

Направление подготовки	09.03.01 - Информатика и вычислительная техника
Профиль	Информатика и вычислительная техника
Программа подготовки	бакалавриат
Семестр	6
Учебная дисциплина	Тестирование и внедрение программного обеспечения

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №1 ВАРИАНТ №1

1. Аббревиатура TDD расшифровывается как:

- а) Test Design Development
- б) Test Driven Design
- в) Test Design Dependency
- г) Test Driven Development

2. Тестирование ПО минимальным набором тестов для проверки на явные ошибки:

- а) юнит-тестирование
- б) smoke-тестирование
- в) бета-тестирование
- г) тестирование белого ящика

3. Уровни тестирования:

- а) тестирование черного ящика, белого ящика, серого ящика
- б) функциональное, модульное, системное тестирование
- в) модульное, интеграционное, системное тестирование
- г) функциональное, нагрузочное, конфигурационное тестирование

4. Назначение изоляционных фреймворков состоит в:

- а) создании интерфейса внешней зависимости на основе заглушки
- б) создании заглушки внешней зависимости на основе интерфейса
- в) создании внешней зависимости на основе интерфейса
- г) создании интерфейса заглушки на основе внешней зависимости

5. Выберите вариант, в котором все четыре термина относятся к xUnit:

- а) TestFixture, TestCase, SetUp, TearDown
- б) TestFixture, TestScript, SetUp, TearDown
- в) TestData, TestCase, Assert, TearDown
- г) TestFixture, TestBlock, SetUp, Assert

6. Задан класс Foo со статическим методом string FormatDays(int days).

Примеры работы метода:

Foo.FormatDays(1)	->	“1 день”
Foo.FormatDays(2)	->	“2 дня”
Foo.FormatDays(12)	->	“12 дней”
Foo.FormatDays(-12)	->	“12 дней”

Напишите юнит-тест для метода FormatDays и сам метод. Укажите секции кода, в соответствии с принципом AAA.

Утверждено на заседании кафедры компьютерных технологий,
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой
Преподаватель

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	1
2	1
3	1
4	1
5	1
6	10
Всего	15

10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Экзамен не предусмотрен.

11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

Назначение изоляционных фреймворков состоит в:

- а) создании интерфейса внешней зависимости на основе заглушки
- б) создании заглушки внешней зависимости на основе интерфейса
- в) создании внешней зависимости на основе интерфейса
- г) создании интерфейса заглушки на основе внешней зависимости

Выберите вариант, в котором все четыре термина относятся к xUnit:

- а) TestFixture, TestCase, SetUp, TearDown
- б) TestFixture, TestScript, SetUp, TearDown
- в) TestData, TestCase, Assert, TearDown
- г) TestFixture, TestBlock, SetUp, Assert.

12. КРИТЕРИИ ОЦЕНИВАНИЯ

По курсу предполагается проведение промежуточной аттестации в виде модульного контроля, выполнение лабораторных работ.

*Распределение баллов, которые могут получить студенты
в процессе изучения дисциплины*

	Содержательный модуль №1				Содержательный модуль №2			Всего
	Лабораторные работы		Мод. контр. работа	Всего С.М. №1	Лабораторные работы		Всего С.М. №2	
	№1	№2			№3	№4		
Макс. балл	20	20	15	55	30	15	45	100

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (зачет)
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

Оценка за овладение курса выставляется по следующим критериям:

– Оценку «отлично» заслуживает студент, который обнаружил глубокие знания при ответах на теоретические вопросы по темам курса, а также выполнил практические задания в полном объеме и набрал более 90 баллов.

– Оценку «хорошо» заслуживает студент, сделавший ошибки в теоретических или практических ответах, которые могут быть интерпретированы как малозначительные для вопросов, которые рассматривались. Студент должен набрать более 75 баллов.

– Оценку «удовлетворительно» заслуживает студент, который выполнил задания неполно и с ошибками, но при этом набрал более 60 баллов.

– Оценку «неудовлетворительно» заслуживает студент, который не выполнил большинства теоретических и практических задач и набрал менее 60 баллов.

13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Для проведения лекционных занятий требуется аудитория на поток, оборудованная мультимедийным проектором и экраном, или интерактивной доской. Для проведения лабораторных занятий по дисциплине необходим оборудованный ПЭВМ или ноутбуками компьютерный класс с возможностью выхода в Интернет.

14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
<i>Основная литература</i>			
1.	Galezowski G. Test-driven Development [Текст] / G. Galezowski. – LeanPub, 2017. – 285 p.	-	Да
<i>Дополнительная литература</i>			
2.	Бек К. Экстремальное программирование. Разработка через тестирование / К. Бек. – СПб.: Питер, 2003. – 512 с.	1	Нет
3.	Ошероув Р. Искусство автономного тестирования с примерами на C# / Р. Ошероув. – М.: ДМК Пресс, 2016. – 360 с.	-	Нет
4.	Месарош Дж. Шаблоны тестирования xUnit: рефакторинг кода тестов. – М.: Вильямс, 2008. – 629 с.	-	Нет
5.	Калбертсон Р. Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. – М.: Вильямс, 2002. – 384 с.	-	Нет

15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Материалы курса «Технологии разработки программного обеспечения» (автор – Шарий Т.В.). URL: <https://github.com/ar1st0crat/SoftDevCourse/> (дата обращения 18.03.2020 г.).
2. Паттерны xUnit. URL: <http://xunitpatterns.com> (дата обращения – 18.03.2020 г.).

16. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. IDE Visual Studio Community версии 2017 или 2019.
2. Библиотека NUnit.
3. Библиотека Moq.
4. Selenium IDE для браузера Firefox, Selenium WebDriver.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2020 год.
Протокол № 12 от «2» апреля 2020 г.
Заведующий кафедрой Ермоленко Т.В.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2021 год.
Протокол № ____ от «____» _____ 2021 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2022 год.
Протокол № ____ от «____» _____ 2022 г.
Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2023 год.
Протокол № ____ от «____» _____ 2023 г.
Заведующий кафедрой