

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Владимирский государственный университет имени
Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

УТВЕРЖДАЮ
Проректор по учебно-методической
работе



А.А. Панфилов

« 22 » 01 2016 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Практикум по решению задач на ЭВМ

Направление подготовки 44.03.01 Педагогическое образование

Профиль подготовки «Информатика»

Уровень высшего образования бакалавриат

Форма обучения заочная

Семестр	Трудоём- кость зач. ед, час.	Лекций, час.	Практ. зан., час.	Лаборат. работ, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
6	8 / 288	-	-	8	280	ЗАЧЕТ
7	2 / 72	-	-	14	58	ЗАЧЕТ
Итого	10 / 360	-	-	22	338	2 ЗАЧЕТА

Владимир, 2016

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цели:

1. Формирование у студентов навыков работы с современными технологиями в программировании для решения прикладных задач.
2. Развитие операционного мышления направленного на выбор оптимальных действий, на умение планировать свою деятельность и предвидеть ее результаты.
3. Формирование опыта работы в коллективе, в частности рефлексии.

Задачи дисциплины:

- Сформировать навыки работы с программной платформой .NET Framework и реализации ООП парадигмы.
- Развитие принципов разработки алгоритмов и программ, их оптимизации.
- Изучение и использование различных методов программирования.
- Формирование опыта разработки алгоритмов и решения задач.
- Освоение понятий и принципов ООП в рамках платформы .NET Framework.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Практикум по решению задач на ЭВМ» относится к вариативной части учебного плана по направлению «Педагогическое образование».

Для освоения дисциплины студенты используют знания и умения, сформированные в процессе изучения таких дисциплин, как «Современные ИТ», «Программирование», «Теория алгоритмов», «Теоретическая информатика».

Освоение данной дисциплины способствует подготовке студентов к итоговой государственной аттестации.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка компетенции
ОК-6	способностью к самоорганизации и самообразованию;
ПК-1	готовностью реализовывать образовательные программы по учебным предметам в соответствии с требованиями общеобразовательных стандартов;
ПК-12	способность руководить учебно-исследовательской деятельностью обучающихся.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать:

- теоретическую основу, важнейшие особенности и возможности программной платформы .NET Framework 4.5 (ОК-6 / ПК-1 / ПК-12);

- синтаксис языка программирования C# и реализация принципов ООП средствами .NET Framework (ПК-1);
- методы и приемы реализации алгоритмов на базе объектной и компонентной модели проекта (ОК-6 / ПК-1).

Уметь:

- моделировать научные и практические задачи средствами .NET Framework (ОК-6 / ПК-1);
- применять новые технологии на основе практических задач (ПК-1 / ПК-12);
- проецировать полученные знания для реализации педагогических задач в процессе обучения основам алгоритмизации и программирования (ОК-6 / ПК-1);
- осуществлять согласованную работу в коллективе из нескольких человек в целях достижения поставленной учебной задачи (ПК-12).

Владеть:

- приемами исследования математических задач средствами .NET Framework (ПК-1 / ПК-12);
- навыками работы со справочными системами по технологии NET Framework и языку программирования C# (ПК-1).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоёмкость дисциплины составляет 10 зачетных единиц, 360 часов.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические	Лабораторные	Контрольные работы	СРС	КП/КР		
1	Статические компоненты. Структуры и перечисления	6				2			80		
2	Строки и файлы Обработка исключений	6				2			60	1/50%	
3	Наследование и полиморфизм	6				2			80		

4	Интерфейсы. Делегаты. События. Обобщения.	6			2		60		1/50%	
Всего (6 семестр)					8		280		2/25%	ЗАЧЕТ
1	Перегрузка операторов.	7			4		8		1/25%	
2	Методы расширения.	7			2		4			
3	Динамически подключаемые библиотеки.	7			2		6		1/50%	
4	Инкапсуляция на уровне класса.	7			2		10		1/50%	
5	Разработка проектов WindowsForms.	7			4		30		2/50%	
Всего (7 семестр)					14		58		5/35.7%	ЗАЧЕТ
ВСЕГО					22		338		7/31.8%	2 ЗАЧЕТА

Темы и содержание лабораторных занятий

6 семестр

Тема 1. Статические компоненты, структуры, перечисления

- Задачи, приводящие к необходимости введение статических компонентов.
- Статичные поля класса и методы класса.
- Статичные классы.
- Использование структур в качестве параметров методов.
- Использование структур в иерархии типов данных.
- Перечисления.
- Обработка данных перечислимого типа.
- Решение задач.

Тема 2. Строки и файлы. Обработка исключений

- Методы класса System.String. Примеры обработки строк.
- Класс StringBuilder.
- Пространство имен System.IO.
- Запись и чтение текстовых файлов.
- Перехват исключений. Класс Excerpton. Обработка многочисленных исключений.
- Решение задач.

Тема 3. Наследование и полиморфизм

- Основы наследования.
- Виртуальные и абстрактные классы.
- Решение задач.

Тема 4. Интерфейсы. Делегаты. События. Обобщения

- Интерфейсы и их роль в иерархии типов.

- Делегаты.
- События как методы обратного вызова.
- Связь событий и делегатов.
- Решение практических задач.
- Назначение обобщений и их применение.
- Обобщённые классы.
- Решение практических задач.

7 семестр

Тема 1. Перегрузка операторов

- Понятие перегрузки, ее применение в реальных задачах.
- Синтаксис перегрузок.
- Изучение возможности перегрузки операторов на базе реализации модуля класса комплексных чисел.

Тема 2. Методы расширения

- Роль методов расширения, дополнительный функционал.
- Методика использования методов расширения.
- Примеры создания и использования методов расширения.
- Решение задач.

Тема 3. Динамически подключаемые библиотеки

- Технология динамически подключаемых библиотек.
- Механизмы создания и подключения динамических библиотек.
- Вопросы производительности и безопасности технологии DLL.
- Решение практических задач.

Тема 4. Инкапсуляция на уровне класса

- Уровни доступа к компонентам класс и классам.
- Запечатанные классы.
- Методика работы с закрытыми полями, принцип «черного ящика»/
- Свойства.
- Решение задач.

Тема 5. Разработка проектов WindowsForms

- GUI приложения.
- Разработка GUI с помощью Visual Studio, SharpDevelop и сборка в ручном режиме.
- Структура шаблона GUI приложения.
- Визуальный конструктор формы.
- Компоненты класса Control.
- Разработка приложений.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Изучение курса «ПРЗ на ЭВМ» предполагает сочетание лабораторных занятий и самостоятельной работы студентов.

На лабораторных занятиях, общий объем которых указан в тематическом плане, студенты изучают теоретический минимум, выполняют задания (индивидуально / попарно или в группах из нескольких человек), консультируются по самостоятельной работе с преподавателем.

Самостоятельная работа предполагает более детальное знакомство с теоретическим материалом и предварительную подготовку к новым лабораторным работам.

При изучении учебного материала данной дисциплины следующие технологии обучения: учебные групповые дискуссии: обсуждения задач (методы, приемы решения, выбор оптимального способа решения, количество возможных случаев для рассмотрения и т.п.), мозговой штурм, презентация микроисследований и их обсуждение, технология проблемного обучения.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6.1 ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Вопросы к зачету 6-го семестра

1. Статические компоненты
2. Задачи, приводящие к необходимости введение статических компонентов.
3. Модификатор `static`. Статичные поля класса.
4. Статичные методы класса.
5. Статичные классы. Примеры использования.
6. Структуры и перечисления.
7. Использование структур в качестве параметров методов.
8. Использование структур в иерархии типов данных.
9. Перечислимый тип как механизм повышения качества логики построения приложения.
10. Обработка данных перечислимого типа.
11. Строки и файлы.
12. Методы класса `System.String`. Примеры обработки строк.
13. Класс `StringBuilder`.
14. Регулярные выражения.
15. Пространство имен `System.IO`.

16. Запись и чтение текстовых файлов.
17. Работа с жестким диском.
18. Обработка исключений и перехват исключений.
19. Класс Exception.
20. Обработка многочисленных исключений.
21. Наследование и полиморфизм
22. Конструкторы класса в наследовании.
23. Виртуальные методы.
24. Абстрактные классы.
25. Интерфейсы и их роль в иерархии типов.
26. Интерфейсные свойства.
27. Делегаты.
28. Групповой вызов и адресация делегируемых методов.
29. События как методы обратного вызова.
30. Связь событий и делегатов.
31. Обобщения.
32. Обобщённые классы.
33. Стек на обобщенном типе.

Вопросы к зачету 7-го семестра

1. Понятие перегрузки операторов. Математические задачи, реализуемые с помощью перегрузки.
2. Синтаксис перегрузок.
3. Перегрузка операторов на базе реализации модуля класса комплексных чисел.
4. Перегрузка операторов на базе реализации модуля класса матриц.
5. Методы расширения. Роль методов расширения, дополнительный функционал.
6. Методика использования методов расширения.
7. Примеры создания и использования методов расширения.
8. Динамически подключаемые библиотеки. Технология DLL.
9. Механизмы создания и подключения динамических библиотек.
10. Вопросы производительности и безопасности технологии DLL.
11. Инкапсуляция на уровне класса. Уровни доступа к компонентам класс и классам.
12. Запечатанные классы.
13. Методика работы с закрытыми полями, принцип «черного ящика».
14. Свойства.
15. Разработка проектов WindowsForms.
16. GUI приложения.
17. Разработка GUI с помощью Visual Studio, SharpDevelop и сборка в ручном режиме.
18. Структура шаблона GUI приложения.
19. Визуальный конструктор формы. Компоненты класса Control.
20. Реализация меню.

6.2 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Вопросы для самостоятельной работы

6 семестр

1. Статичные классы в пространстве имен System.
2. Строковые операции. Исследование задач, требующих работы со строками.
3. Регулярны выражения. Преимущества и недостатки.
4. Абстрактные классы. Реализация математических моделей.
5. Наследование и полиморфизм. Целесообразность наследования при реализации собственных классов и сборок.
6. Делегаты. Принципы построения. Связь с указателями на методы в языке C++. Вопросы безопасности.
7. Интерфейсы и абстрактные классы.
8. Обобщения как средство расширения возможностей работы с компонентами.

7 семестр

1. Реализация абстрактной модели векторных пространств и основных операций над векторами средствами C#.
2. Модель рационального числа. Реализация операций над рациональными числами
3. Проблемы использования DLL.
4. Использование библиотеки .NET Framework для получения информации о компьютере.
5. Основные классы для работы с сетью Интернет. Примеры подключения и обработки запросов.
6. Работа с графическими примитивами.
7. Технология GDI+.
8. Программирование GUI в рамках школьного курса. Проблемы и трудности в реализации подхода.

Примеры заданий для проектной деятельности

6 семестр

1. Реализуйте класс Complex, позволяющий работать с комплексными числами в удобной форме. Класс позволяет:
 - создавать комплексные числа (с помощью параметризованного конструктора);
 - складывать, вычитать, умножать и делить числа (с помощью статических методов);
 - вычислять модуль и аргумент числа;

- выводить результат в удобной форме.
2. Реализуйте класс `Matrix`, позволяющий работать с квадратными матрицами. Класс позволяет:
- создавать матрицы по указанной размерности или массиву;
 - складывать, вычитать, умножать, делить матрицы, домножать матрицу на число.
 - выводить результат в удобной форме.

7 семестр

Разработать класс, реализующий основные операции над квадратными матрицами. С помощью перегрузки операторов определить следующие операции:

- сложения;
- разности;
- умножения матрицы на матрицу;
- умножение матрицы на число;
- изменение знака на противоположный.

Перегрузить оператор равенства, проверяющий матрицы на равенство.

Предполагается, что операции совершаются над матрицами одинаковой размерности. В противном случае возвращается ложный результат.

В классе определить два конструктора: первому передается двумерный массив, отвечающий за матрицу, а второму передается число (размерность матрицы) и создается матрица из нулей.

Пример дополнительного материала для организации самостоятельной работы

6 семестр

Обработка поведения элементов перечисления

Перечисления задают логику построения, но не определяют поведение элементов объекта в каждом конкретном значении. Эта работа должна быть проделана отдельно.

Обычно все элементы перечисления равнозначны в плане реализации. Т.е. алгоритм их обработки одинаков, а разница лишь в итоговом значении.

Так, в нашей задаче определено два перечисления, отвечающие за тип фигуры и ее цвет, то однотипная обработка очевидна. Если мы хотим на выводе получать названия цветов и типов фигур на родном языке, то необходимо добавить функцию, которая для каждого значения задаст корректное соответствие.

Перечисления

```
using System;  
  
enum FigureType { Triangle, Rectangle, Circle };  
enum FigureColor { Red, Green, Blue, Black };
```

```

class Figure
{
    public FigureType Type { get; set; }
    public FigureColor Color { get; set; }

    public Figure(FigureType type, FigureColor color)
    {
        Type = type;
        Color = color;
    }

    // метод выводит общие данные по объекту
    public void Info()
    {
        Console.WriteLine("Тип: " + GetFigureType(Type));
    }

    // метод возвращает название указанного цвета на русском языке
    private string GetFigureType(FigureType type)
    {
        switch (type)
        {
            case FigureType.Triangle:
                return "Треугольник";
            case FigureType.Rectangle:
                return "Квадрат";
            default:
                return "Круг";
        }
    }
}

class Program
{
    static void Main()
    {
        // создание объекта и инициализация его свойств
        Figure fig = new Figure(
            FigureType.Rectangle,
            FigureColor.Black
        );

        fig.Info();

        Console.ReadKey();
    }
}

```

Обратите внимание, что методу `GetFigureType` определен закрытый доступ. Таким образом мы скрываем его работу внутри класса и запрещаем внешний вызов метода (что и нежелательно).

Разумеется, аналогичный механизм необходимо проделать и для элементов перечисления `FigureColor`.

Инкапсуляция по смыслу

Последнее, что необходимо отметить в нашей задаче – достаточно ограниченное распространение типов `FigureType` и `ColorType`. Класс `Figure` весьма узкопрофилирован, и нет смысла распространять перечисления на другие классы.

Разумная идея – внедрить перечисления `FigureType` и `ColorType` внутри класса `Figure`.

Перечисления

```
using System;

class Figure
{
    public enum FigureType { Triangle, Rectangle, Circle };
    public enum FigureColor { Red, Green, Blue, Black };

    // свойства, конструктор, вывод и т.д.
    // . . .
}

class Program
{
    static void Main()
    {
        // создание объекта и инициализация его свойств
        Figure fig = new Figure(
            Figure.FigureType.Rectangle,
            Figure.FigureColor.Black
        );

        fig.Color = Figure.FigureColor.Green;

        fig.Info();

        Console.ReadKey();
    }
}
```

Перечисления в примере являются уже элементами класса, поэтому его необходимо будет указывать при каждом обращении к значению перечисления.

7 семестр

Класс `ImageBox`

Элемент управления `PictureBox` применяется для отображения графических изображений. Изображение может быть в формате BMP, JPEG, GIF, PNG, метафайла или пиктограммы.

Свойство `Image` задает ссылку на изображение:

```
public Image Image { get; set; }
```

Из одноименного класса `Image` отдельно выделим статический метод `FromFile`, загружающий изображение по указанному пути.

Свойство `SizeMode` определяет способ отображения внутри рамки:

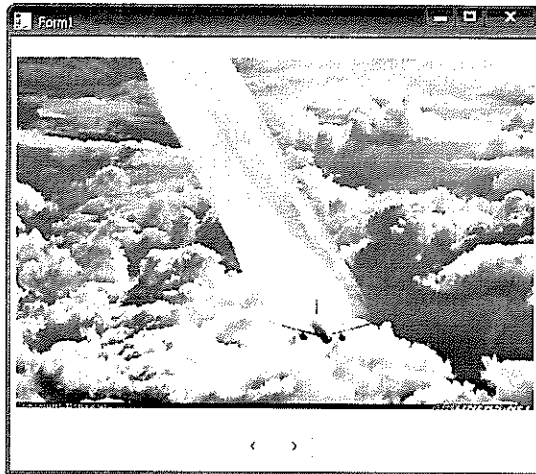
```
public PictureBoxSizeMode SizeMode { get; set; }
```

где перечисление `PictureBoxSizeMode` задает следующие варианты:

```
public enum PictureBoxSizeMode = {  
    AutoSize, CenterImage, Normal, StretchImage, Zoom  
};
```

Form1.cs

Следующее приложение позволяет просматривать изображения в текущей директории.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
  
namespace WF_2  
{  
    public partial class Form1 : Form  
    {  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private string[] files; // массив путей к файлам изображений  
        private int index = 0; // индекс текущего изображения  
  
        private void Form1_Load(object sender, EventArgs e)  
        {  
            // директория с изображениями  
            string path =  
                @"D:\Мои документы\Visual Studio 2010\Projects\WF_2\WF_2\Pictures";  
            // получить массив из имен файлов с расширением ".jpg"  
            files = System.IO.Directory.GetFiles(path, "*.jpg");  
            // загрузить первое изображение
```

```
        this.pictureBox1.Image = Image.FromFile(files[0]);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (index > 0) index--; else index = files.Length - 1;
        this.pictureBox1.Image = Image.FromFile(files[index]);
    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (index < files.Length - 1) index++; else index = 0;
        this.pictureBox1.Image = Image.FromFile(files[index]);
    }
}
}
```

При настройке свойств формы объекту PictureBox рекомендуется задать якорь со всех сторон, чтобы при изменении размера формы изображение масштабировалось автоматически.

Класс необходимо расширить двумя полями. Первое хранит пути к изображениям: их можно получить с помощью метода GetFiles() класса Directory. Второе поле – индекс текущего изображения; он будет увеличиваться либо уменьшаться при прокрутке. На этом уровне переменные будут общедоступны остальным управляющим элементам формы, в частности кнопкам.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Основная литература:

1. Златопольский Д.М. Программирование. Типовые задачи, алгоритмы, методы [Электронный ресурс]/ Златопольский Д.М.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2015.— 224 с. Режим доступа: <http://www.iprbookshop.ru/12264>
2. Объектно-ориентированное программирование с примерами на С#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с.: 70x100 1/16. - (Высшее образование: Бакалавриат) (Обложка) ISBN 978-5-00091-144-0. Режим доступа: <http://znanium.com/bookread2.php?book=529350>
3. Базовые средства программирования на Visual Basic в среде VisualStudio Net. Практикум: Учебное пособие / Шакин В.Н. - М.: Форум, НИЦ ИНФРА-М, 2015. - 288 с.: 70x100 1/16. - (Высшее образование: Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-00091-054-2. Режим доступа: <http://znanium.com/bookread2.php?book=502047>

Дополнительная литература

1. Практикум на ЭВМ. Часть 1 [Электронный ресурс]: учебное пособие/ — Электрон. текстовые данные.— М.: Евразийский открытый институт, 2012.— 263 с. Режим доступа: <http://www.iprbookshop.ru/14644>
2. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0355-1, 500 экз. Режим доступа: <http://znanium.com/bookread2.php?book=429576>
3. Туркин О.В. VBA. Практическое программирование [Электронный ресурс]/ Туркин О.В.— Электрон. текстовые данные.— М.: СОЛОН-ПРЕСС, 2010.— 128 с. Режим доступа: <http://www.iprbookshop.ru/8701>

Программное обеспечение и Интернет-ресурсы

1. Портал: Компьютерные технологии, <http://ru.wikipedia.org/wiki>, 2016.
2. Официальный сайт поддержки компании Microsoft: <https://msdn.microsoft.com>, 2016.
3. <http://professorweb.ru/>, электронные материалы по технологии .NET, 2016.

Периодические издания

1. Журнал «Информатика и образование»: <http://infojournal.ru/>
2. Журнал «Информационные технологии»: <http://novtex.ru/IT/>
3. Журнал «Информационное общество»: <http://www.infosoc.iis.ru/index.html>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

- Компьютерный класс на основе ЭВМ ПК IntelCore с доступом в сеть Интернет, маркерная и интерактивная доски, переносной ноутбук, наушники, колонки.
- Мультимедийный комплекс в составе: Ноутбук с выходом в сеть Интернет, мультимедиа проектор, экран белый матовый, доска маркерная.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 44.03.01 «Педагогическое образование», профиль «Информатика»

Рабочую программу составил асс. Якубович Д.А.

(ФИО, подпись)



Рецензент (представитель работодателя) учитель высшей категории МБОУ СОШ №15

г.Владимир Козлова С.А.

(место работы, должность, ФИО, подпись)

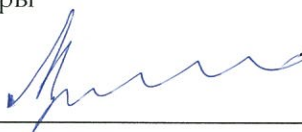


Программа рассмотрена и одобрена на заседании кафедры

Протокол № 5а от 15.01.16 года

Заведующий кафедрой ИИТО, проф. Медведев Ю.А.

(ФИО, подпись)



Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 44.03.05 Педагогическое образование

Протокол № 1 от 22.01.16 года

Председатель комиссии Артамонова М.В.

(ФИО, подпись)



**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____