

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

УТВЕРЖДАЮ

Проректор
по образовательной деятельности

А.А.Панфилов

« 26 » 06 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направление подготовки: 09.04.01 Информатика и вычислительная техника

Профиль/программа подготовки: Автоматизация проектирования электронной вычислительной аппаратуры

Уровень высшего образования: магистратура

Форма обучения: очная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточной аттестации (экзамен/зачет/зачет с оценкой)
2	5/180	18	36		90	Экзамен (36)
3	3/108	18	18		72	Зачет, курсовой проект
Итого	8/288	36	54		162	Экзамен (36), зачет, курсовой проект

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью дисциплины является рассмотрение круга специальных вопросов в области разработки программного обеспечения различного назначения. Дисциплина должна способствовать более глубокому пониманию теоретических и практических проблем и их решений при разработке программ с использованием различных парадигм программирования, платформ и языков.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина **Технология разработки программного обеспечения** относится к части, формируемой участниками образовательных отношений.

Пререквизиты дисциплины: дисциплина опирается на знания предметов основной образовательной программы высшего образования: математика, языки программирования.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП

Код формируемых компетенций	Уровень освоения компетенции	Планируемые результаты обучения по дисциплине характеризующие этапы формирования компетенций (показатели освоения компетенции)
<i>1</i>	<i>2</i>	<i>3</i>
<i>УК-2</i>	<i>частичное</i>	Знать: методы представления и описания результатов проектной деятельности Уметь: обосновывать практическую и теоретическую значимость полученных результатов Владеть: средствами описания и реализации проектных решений в процессе проектирования программного обеспечения
<i>ОПК-2</i>	<i>частичное</i>	Знать: способы разработки оригинальных алгоритмов и программных средств, в том числе с использованием современных интеллектуальных технологий Уметь: применять методы и технологию проектирования оригинальных алгоритмов и программ Владеть: навыками проектирования оригинальных алгоритмов и программ
<i>ОПК-5</i>	<i>частичное</i>	Знать: основные принципы разработки современных программных систем Уметь: разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем Владеть: средствами разработки программного обеспечения на ЭВМ
<i>ОПК-6</i>	<i>частичное</i>	Знать: способы разработки компонентов программно-аппаратных комплексов обработки информации и автоматизированного проектирования Уметь: разрабатывать программно-аппаратные комплексы обработки информации и автоматизированного проектирования и их компоненты Владеть: навыками разработки компонентов программно-аппаратных комплексов

<i>ОПК-7</i>	<i>частичное</i>	<p>Знать: как устроены вычислительные комплексы обработки информации и автоматизированного проектирования</p> <p>Уметь: адаптировать зарубежные комплексы обработки информации и автоматизированного проектирования к нуждам отечественных предприятий</p> <p>Владеть: методами и средствами владения и обработки информации на ЭВМ</p>
<i>ПК-1</i>	<i>частичное</i>	<p>Знать: стандарты на разработку пользовательских документов</p> <p>Уметь: разрабатывать пользовательские документы, а также стандартные технические документы на основе предоставленного материала</p> <p>Владеть: грамотным техническим языком и техническими средствами подготовки документации.</p>

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 8 зачетных единиц, 288 часов

№ п/п	Наименование тем и/или разделов/тем 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, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
1	Основные определения и термины	2	1-2	2	4		10	3/50	
2	Основные конструкции Scala	2	3-4	2	4		10	3/50	
3	Базовые типы и операции	2	5-6	2	4		10	3/50	Рейтинг-контроль №1
4	Функциональные объекты	2	7-8	2	4		10	3/50	
5	Встроенные конструкции управления	2	9-10	2	4		10	3/50	
6	Функции и замыкания	2	11-12	2	4		10	3/50	Рейтинг-контроль №2
7	Абстракция управления	2	13-14	2	4		10	3/50	
8	Композиция и наследование	2	15-16	2	4		10	3/50	
9	Trait-конструкции	2	17-18	2	4		10	3/50	Рейтинг-контроль №3
Всего за 2семестр:							90	27/50	Экзамен (36)
10	Списки	3	1-2	2	2		4	2/50	
11	Case-классы и сопоставление с образцом	3	3-4	2	2		4	2/50	
12	Коллекции	3	5-6	2	2		4	2/50	Рейтинг-контроль №1
13	Фог-выражения	3	7-8	2	2		4	2/50	
14	Функции высшего порядка	3	9-10	2	2		4	2/50	
15	Параллелизм	3	11-12	2	2		4	2/50	Рейтинг-контроль №2
16	Грамматический разбор	3	13-14	2	2		4	2/50	
17	Библиотека комбинаторов	3	15-16	2	2		4	2/50	
18	Работа с XML	3	17-18	2	2		4	2/50	Рейтинг-контроль №3
Всего за 3семестр:							72	18/50	Зачет
Наличие в дисциплине КП/КР									Курсовой проект
Итого по дисциплине							162	45/50	Зачет

Содержание лекционных занятий по дисциплине

Тема 1 Основные определения и термины

О разработке ПО. Процедурная и функциональная метафоры в языках программирования. Роль и значение АЯ Scala в создании современного ПО. Масштабируемость ПО. Первые шаги в Scala. Работа в режиме интерпретатора. Определение переменных. Определение функций. Составление Scala-скриптов. Организация циклов типа while. Оператор if. операторы foreach и for.

Тема 2. Основные конструкции Scala

Параметризация массивов с целью указания типа их элементов. Использование списков. Использование кортежей (tuples). Использование множеств и отображений. Процедурный и функциональный стили программирования. Чтение строк из файла. Классы и объекты. Классы, поля и методы. Одиночные объекты (вида singleton). Scala-приложения. Конструкция trait. Trait Application.

Тема 3. Базовые типы и операции

Литералы. Использование методов в качестве операторов. Арифметические операции. Отношения и логические операции. Побитовые операции. Эквивалентность объектов. Предшествование и ассоциативность операторов. Развитые обертки (rich wrappers).

Тема 4. Функциональные объекты

Пример организации класса для работы с рациональными числами: спецификация, конструктор, метод toString, проверка предусловий, добавление полей, ссылки this, дополнительные конструкторы, приватные поля и методы. Определение операторов. Идентификаторы в Scala. Перегрузка методов. Преобразования по умолчанию (implicit conversions). О краткости выражения и простоте понимания.

Тема 5. Встроенные конструкции управления

If-выражения. While-циклы. For-выражения. Обработка исключений. Match-выражения. Об отсутствии операторов break и continue. Область видимости переменных. Преобразование процедурного кода в функциональный и наоборот.

Тема 6. Функции и замыкания

Методы. Локальные функции. Функции в качестве аргументов функций (first-class functions). Функции-литералы. Частично-определенные функции. Замыкания. Специальные формы вызова функций. Хвостовая рекурсия.

Тема 7. Абстракция управления

Сокращение повторяемости кода. Упрощение клиентского кода. Создание новых конструкций управления. Передача параметров по имени.

Тема 8. Композиция и наследование

Учебный пример "Библиотека двумерных схем". Абстрактные классы. Определение методов без параметров. Расширение классов. Перегрузка методов и полей. Определение параметрических полей. Вызов конструкторов суперкласса. Модификатор override. Полиморфизм и динамическое связывание. Определения финитных методов. Использование композиции и наследования. Развитие программы примера. Иерархия в Scala. Реализация примитивов. Основные типы.

Тема 9. Trait-конструкции

Принципы работы trait. Тонкая версия развитых интерфейсов. Пример "Объекты-прямоугольники". Trait Ordered. Trait как средство масштабируемости. О нереализованности множественного наследования. О целесообразности использования trait. Пакеты и импорт. Пакеты. Импорт. Импорт по умолчанию. Модификаторы доступа.

Тема 10. Списки

Списковые литералы. Тип List. Создание списков. Базовые операции над списками. Высокоуровневые методы класса List. Методы объекта List. Представление об алгоритме вывода типа в Scala. Принципы организации типа List. Класс ListBuffer.

Тема 11. Case-классы и сопоставление с образцом

Простой пример. Выды образцов. Использование предохранителей в образцах. Наложение образцов. Тип Option. Более сложный пример.

Тема 12. Коллекции

Последовательности. Множества и отображения. Изменяемые (mutable) и неизменяемые (immutable) коллекции. Инициализация коллекций. Кортежи (tuples). Trait Traversable. Trait Iterable. Массивы. Строки. Характеристики производительности.

Тема 13. For-выражения

Учебный пример "Проблема восьми ферзей". Использование метафоры запросов в for-выражениях. Трансляция for-выражений. Обобщение for.

Тема 14. Функции высшего порядка

Функции как значения. Анонимные функции. Функции с функциональными параметрами. Выводы типов. Карринг.

Тема 15. Параллелизм

Проблемы организации параллельного выполнения кода. Концепция Actor. Actor-объекты и обмен сообщениями. Actor-объекты на основе базовых потоков. Повышение производительности Actor-объектов на основе повторного использования потоков. Учебный пример "Моделирование дискретных параллельных систем".

Тема 16. Грамматический разбор

Понятие о DSL (Domain Specific Language - проблемно ориентированный язык). Учебный пример "Арифметические выражения". Выполнение грамматического разбора. Базовые анализаторы регулярных выражений. Пример "JSON". Выход программы грамматического разбора.

Тема 17. Библиотека комбинаторов

Реализация комбинаторов. Строковые литералы и регулярные выражения. Лексический анализ. Сообщения об ошибках.

Тема 18. Работа с XML

Организация данных. XML-литералы. Сериализация и десериализация. Извлечение подэлементов и атрибутов. Загрузка и сохранение. Сравнение с образцом.

Содержание практических занятий по дисциплине

Практические занятия во 2-м семестре:

1. Работа в режиме интерпретатора.
2. Основные понятия Scala.
3. Базовые типы и операции.
4. Функциональные объекты.
5. Встроенные конструкции управления.
6. Функции и замыкания.
7. Абстракция управления.
8. Композиция и наследование.
9. Trait-конструкции.

Практические занятия в 3-м семестре:

1. Case-классы и сопоставление с образцом.
2. Работа со списками.
3. Реализация списков.
4. Коллекции.
5. For-выражения.
6. Параллелизм.
7. Грамматический разбор на основе комбинаторов.
8. Работа с XML.
9. Примеры создания Scala-систем.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины «Технология разработки программного обеспечения» используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- Групповая дискуссия (темы №1,2,4,9);
- Применение объектных программных моделей (темы № 8,9,16,18);

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

Текущий контроль успеваемости (семестр 2)

Вопросы рейтинг-контроля №1

- 1) Основные понятия Scala
- 2) Базовые типы и операции

Вопросы рейтинг-контроля №2

- 1) Функциональные объекты
- 2) Функции и замыкания

Вопросы рейтинг-контроля №3

- 1) Композиция и наследование
- 2) Trait-конструкции

Текущий контроль успеваемости (семестр 3)

Вопросы рейтинг-контроля №1

- 1) Case-классы и сопоставление с образцом
- 2) Реализация списков

Вопросы рейтинг-контроля №2

- 1) For-выражения
- 2) Акторы

Вопросы рейтинг-контроля №3

- 1) Грамматический разбор на основе комбинаторов
- 2) Работа с XML

Вопросы к экзамену (семестр 2):

1. Процедурная и функциональная метафоры в языках программирования.
2. Значение АЯ Scala в создании современного ПО. Масштабируемость ПО.
3. Основные понятия Scala. Определение переменных. Определение функций. Составление Scala-скриптов. Организация циклов типа while. Оператор if. операторы foreach и for.
4. Параметризация массивов.

5. Использование списков и кортежей (tuples).
6. Использование множеств и отображений.
7. Процедурный и функциональный стили программирования.
8. Чтение строк из файла.
9. Классы, поля и методы. Одиночные объекты (вида singleton).
10. Scala-приложения.
11. Trait. Trait Application.
12. Использование методов в качестве операторов.
13. Эквивалентность объектов.
14. Предшествование и ассоциативность операторов.
15. Развитые обертки (rich wrappers).
16. Функциональные объекты. Пример.
17. Встроенные конструкции управления. If-выражения и While-циклы.
18. Встроенные конструкции управления. For- и Match-выражения.
19. Обработка исключений.
20. Область видимости переменных.
21. Преобразование процедурного кода в функциональный и наоборот.
22. Функции в качестве аргументов функций (first-class functions).
23. Функции-литералы.
24. Частично-определенные функции.
25. Замыкания.
26. Специальные формы вызова функций.
27. Хвостовая рекурсия.
28. Currying.
29. Создание новых конструкций управления.
30. Передача параметров по имени.
31. Абстрактные классы.
32. Определение методов без параметров.
33. Расширение классов. Перегрузка методов и полей.
34. Полиморфизм и динамическое связывание.
35. Использование композиции и наследования.
36. Основные типы Scala.
37. Принципы работы trait.
38. Trait как средство масштабируемости.
39. О целесообразности использования trait.
40. Пакеты и импорт. Импорт по умолчанию. Модификаторы доступа.

Вопросы на зачету (семестр 3):

1. Использование предохранителей в образцах.
2. Наложение образцов.
3. Тип Option.
4. Списковые литералы.
5. Базовые операции над списками.
6. Множества и отображения.
7. Изменяемые (mutable) и неизменяемые (immutable) коллекции.
8. Массивы.
9. Строки.
10. Использование метафоры запросов в for-выражениях.
11. Проблемы организации параллельного выполнения кода.
12. Концепция Actor.
13. Actor-объекты на основе базовых потоков.
14. DSL (Domain Specific Language - проблемно ориентированный язык).
15. Базовые анализаторы регулярных выражений.
16. Реализация комбинаторов.
17. Строковые литералы и регулярные выражения.

18. Лексический анализ.
19. Сериализация и десериализация XML.
20. Извлечение подэлементов и атрибутов.

Самостоятельная работа студентов

Целью самостоятельной работы являются формирование личности студента, развитие его способности к самообучению и повышению своего профессионального уровня.

Самостоятельная работа заключается в изучении содержания тем курса по конспектам, учебникам и дополнительной литературе, подготовке к лабораторным и практическим занятиям, выполнении заданий для самостоятельной работы,

Самостоятельная работа включает домашнюю работу с лекционными материалами с целью расширения и углубления теоретических знаний, выполнение заданий, предусмотренных контрольными работами, оформлении отчетов по лабораторным работам, выполнении курсового проекта, подготовке к промежуточной аттестации.

Вопросы для самостоятельной работы студентов во 2-м семестре:

1. Процедурная и функциональная метафоры в языках программирования.
2. Значение АЯ Scala в создании современного ПО. Масштабируемость ПО.
3. Основные понятия Scala. Определение переменных. Определение функций. Составление Scala-скриптов. Организация циклов типа while. Оператор if. операторы foreach и for.
4. Параметризация массивов.
5. Использование списков и кортежей (tuples).
6. Использование множеств и отображений.
7. Процедурный и функциональный стили программирования.
8. Чтение строк из файла.
9. Классы, поля и методы. Одиночные объекты (вида singleton).
10. Scala-приложения.
11. Trait. Trait Application.
12. Использование методов в качестве операторов.
13. Эквивалентность объектов.
14. Предшествование и ассоциативность операторов.
15. Развитые обертки (rich wrappers).
16. Функциональные объекты. Пример.
17. Встроенные конструкции управления. If-выражения и While-циклы.
18. Встроенные конструкции управления. For- и Match-выражения.
19. Обработка исключений.
20. Область видимости переменных.
21. Преобразование процедурного кода в функциональный и наоборот.
22. Функции в качестве аргументов функций (first-class functions).
23. Функции-литералы.
24. Частично-определенные функции.
25. Замыкания.
26. Специальные формы вызова функций.
27. Хвостовая рекурсия.
28. Currying.
29. Создание новых конструкций управления.
30. Передача параметров по имени.
31. Абстрактные классы.
32. Определение методов без параметров.
33. Расширение классов. Перегрузка методов и полей.
34. Полиморфизм и динамическое связывание.
35. Использование композиции и наследования.
36. Основные типы Scala.
37. Принципы работы trait.

38. Trait как средство масштабируемости.
39. О целесообразности использования trait.
40. Пакеты и импорт. Импорт по умолчанию. Модификаторы доступа.

Вопросы для самостоятельной работы студентов в 3-м семестре:

1. Использование предохранителей в образцах.
2. Наложение образцов.
3. Тип Option.
4. Списковые литералы.
5. Базовые операции над списками.
6. Множества и отображения.
7. Изменяемые (mutable) и неизменяемые (immutable) коллекции.
8. Массивы.
9. Строки.
10. Использование метафоры запросов в for-выражениях.
11. Проблемы организации параллельного выполнения кода.
12. Концепция Actor.
13. Actor-объекты на основе базовых потоков.
14. DSL (Domain Specific Language - проблемно ориентированный язык).
15. Базовые анализаторы регулярных выражений.
16. Реализация комбинаторов.
17. Строковые литералы и регулярные выражения.
18. Лексический анализ.
19. Сериализация и десериализация XML.
20. Извлечение подэлементов и атрибутов.

Задания к курсовому проекту

1. DSL для описания графов
2. DSL для хранения описаний UML-диаграмм
3. DSL для трансформации текстовых данных
4. DSL для процессирования и печати текста
5. DSL для создания функциональных тестов библиотеки программ
6. DSL для создания функциональных тестов Web-приложения
7. DSL для сборки программного проекта
8. DSL для описания математических формул
9. DSL для программирования бизнес-процессов

Фонд оценочных средств для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ	
		Количество экземпляров изданий в библиотеке ВлГУ в соответствии с ФГОС ВО	Наличие в электронной библиотеке ВлГУ
1	2	3	4
Основная литература			
1. Хорстманн К., Scala для нетерпеливых [Электронный ресурс] / Хорстманн К. - М. : ДМК Пресс, 2017. - 414 с. - ISBN 978-5-97060-536-3	2017		http://www.studentlibrary.ru/book/ISBN9785970605363.html
2. Абрамян М.Э., Введение в	2017		http://www.studentlibrary.ru/

стандартную библиотеку шаблонов C++: учебник / Абрамян М. Э. - Ростов н/Д : Изд-во ЮФУ, 2017. - 178 с. - ISBN 978-5-9275-2374-0			book/ ISBN9785927523740.html
3. Ашарина И.В., Объектно-ориентированное программирование в C++: лекции и упражнения [Электронный ресурс]: Учебное пособие для вузов / Ашарина И.В. - М. : Горячая линия - Телеком, 2017. - 336 с. - ISBN 978-5-9912-0423-1	2017		http://www.studentlibrary.ru/ book/ ISBN9785991204231.html
4. Суханов М.В., Основы Microsoft .NET Framework и языка программирования C#: [Электронный ресурс] / Суханов М.В. - Архангельск : ИД САФУ, 2014. - 96 с. - ISBN 978-5-261-00934-4	2014		http://www.studentlibrary.ru/ book/ ISBN9785261009344.html
Дополнительная литература			
1. Модели параллельного программирования / Федотов И.Е. - М. : СОЛОН-ПРЕСС, 2012.	2012		http://www.studentlibrary.ru/ book/ ISBN9785913591029.html
2. Уильяме, Энтони. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. — М. : ДМК Пресс, 2012. — 672 с.	2012		http://www.studentlibrary.ru/ book/ ISBN9785940744481.html
3. Дубов, И. Р., Быков, В.И. Языки программирования. Объектно-ориентированное программирование : учеб. пособие; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2018. – 96 с.	2018		http://e.lib.vlsu.ru/bitstream/ 123456789/7245/1/01731.pdf
4. Языки программирования. Концепции и принципы / Кауфман В.Ш. - М. : ДМК Пресс, 2010.	2010		http://www.studentlibrary.ru/ book/ ISBN9785940746225.html
5. Уорбэртон Р., Лямбда-выражения в Java 8. Функциональное программирование - в массы / Уорбэртон Р. - М. : ДМК Пресс, 2014. - 192 с. - ISBN 978-5-94074-919-6	2014		http://www.studentlibrary.ru/ book/ ISBN9785940749196.html

7.2. Периодические издания

Журналы (<https://elibrary.ru/>):

1. Вычислительные технологии
2. Кибернетика и программирование
3. Journal of Object Technology

7.3. Интернет-ресурсы

1. Официальный сайт Scala: [<http://www.scala-lang.org/>].
2. JDK: [<http://www.oracle.com/technetwork/java/javase/downloads/index.html>].

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы.

Вычислительная техника должна отвечать следующим требованиям:

1. Персональные компьютеры для выполнения практических заданий, а также для информационно-поисковой работы.
2. Проектор с демонстрационным ПК.

Рабочую программу составил профессор кафедры ВТиСУ Дубов И.Р.

(ФИО, подпись)



Рецензент

(представитель работодателя)



Директор по производству ООО «Фирма «Инрэко ЛАН»»
Васильев А.В.

Программа рассмотрена и одобрена на заседании кафедры

ВТиСУ

Протокол № 6 от 26.06.2019 года

Заведующий кафедрой

(ФИО, подпись)



Ланцов В.Н.

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии
направления 09.04.01

Протокол № 2 от 26.06.2019 года

Председатель комиссии

(ФИО, подпись)



Ланцов В.Н.

**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

Рабочая программа одобрена на 2020/2021 учебный год

Протокол заседания кафедры № 1 от 31.08.20 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 _____

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 _____