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

УТВЕРЖДАЮ
Проректор
по учебно-методической работе
_____ А.А.Панфилов
«10» 02 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Профиль/программа подготовки: Вычислительные машины, комплексы, системы и сети

Уровень высшего образования: магистратура

Форма обучения: очная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
2	3/108	18	36	-	18	Экзамен (36)
3	2/72	18	36	-	18	Зачет, курсовой проект
Итого	5/180	36	72	-	36	Экзамен (36), курсовой проект, зачет

Владимир 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Общая цель освоения дисциплины «Технология разработки программного обеспечения» состоит в стремлении снабдить учащихся по возможности универсальным инструментом для разработки программных систем и решения на этой основе проблем внедрения результатов работы магистрантов практически по всем направлениям их деятельности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина входит в базовую часть ОПОП. Для изучения дисциплины «Технология разработки программного обеспечения» необходимо знать такие дисциплины как «Иностранный язык», «Дискретная математика», «Математическая логика и теория алгоритмов», «Языки программирования», «Программирование», «Системное программное обеспечение», «Операционные системы», «Технология программирования», «Сети и телекоммуникации». В ходе первого семестра обучение по данной дисциплине корреспондируется с изучением дисциплины «Вычислительные системы». Необходимо хорошо знать языки программирование C, C++, Java и иметь практические навыки составления программ средней сложности на этих языках. Иметь хорошие навыки работы на уровне прикладного интерфейса программирования для операционных систем Windows и Linux.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения дисциплины «Технология разработки программного обеспечения» обучающийся должен обладать следующими компетенциями: способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень (ОК-1), использование на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом (ОК-5), способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности (ОК-6), способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности (ОК-7), способность воспринимать математические, естественнонаучные, социально-экономические и профессиональные знания, умением самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе, в новой или незнакомой среде и в междисциплинарном контексте (ОПК-1), культура мышления, способностью выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных их разных

областей науки и техники, выносить суждения на основании неполных данных (ОПК-2), владение методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях (ОПК-2), способность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями (ОПК-6), знание методов оптимизации и умение применять их при решении задач профессиональной деятельности (ПК-3), владение существующими методами и алгоритмами решения задач распознавания и обработки данных (ПК-4), понимание существующих подходов к верификации моделей программного обеспечения (ПК-6), применение перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий (ПК-7).

В результате изучения дисциплины обучающийся должен:

- **ЗНАТЬ**: основы функционального программирования, основные сведения из спецификации Scala, способы программирования типовых задач в Scala, базовые средства Scala предназначенные для разработки DSL.
- **УМЕТЬ**: пользоваться спецификацией Scala, разрабатывать на Scala программы средней сложности , разрабатывать и реализовывать собственные DSL.
- **ВЛАДЕТЬ**: навыками работы с технической и справочной литературой, навыками самостоятельного освоения современных языков программирования и инструментальных средств разработки программных систем.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технология разработки программного обеспечения» составляет 5 зачетных единиц, 180 часа.

№ 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	Семинары	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Введение	2	1-2	2		4			2		3/50	

2	Основные понятия Scala		3-4	2		4			2		3/50	
3	Базовые типы и операции		5-6	2		4			2		3/50	Рейтинг-контроль №1
4	Функциональные объекты		7-8	2		4			2		3/50	
5	Встроенные конструкции управления		9-10	2		4			2		3/50	
6	Функции и замыкания		11-12	2		4			2		3/50	Рейтинг-контроль №2
7	Абстракция управления		13-14	2		4			2		3/50	
8	Композиция и наследование		15-16	2		4			2		3/50	
9	Trait-конструкции		17-18	2		4			2		3/50	Рейтинг-контроль №3
ИТОГО за семестр 2				**18**		**36**			**18**		**24/50%**	**Экзамен (36)**
10	Case-классы и сопоставление с образцом	3	1-2	2		4			2		3/50	
11	Работа со списками		3-4	2		4			2		3/50	
12	Реализация списков		5-6	2		4			2		3/50	Рейтинг-контроль №1
13	Коллекции		7-8	2		4			2		3/50	
14	For-выражения		9-10	2		4			2		3/50	
15	Параллелизм		11-12	2		4			2		3/50	Рейтинг-контроль №2
16	Грамматический разбор на основе комбинаторов		13-14	2		4			2		3/50	
17	Работа с XML		15-16	2		4			2		3/50	
18	Заключение		17-18	2		4			2		3/50	Рейтинг-контроль №3
ИТОГО за семестр 3				**18**		**36**			**18**	**КП**	**24/50%**	**Зачет, курсовой проект**
ВСЕГО				**36**		**72**			**36**		**48/50%**	**Экзамен, курсовой проект, зачет**

4.1 Дидактический минимум разделов дисциплины

1. **Введение**. О разработке ПО. Процедурная и функциональная метафоры в языках программирования. Роль и значение АЯ Scala в создании современного ПО. Мастабируемость ПО. Первые шаги в Scala. Работа в режиме интерпрета-

тора. Определение переменных. Определение функций. Составление Scala-скриптов. Организация циклов типа while. Оператор if. операторы foreach и for.

2. **Основные понятия 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.

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).

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. Преобраования по умолчанию (implicit conversions). О краткости выражения и простоте понимания.

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.

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

7**. Абстракция управления**. Сокращение повторяемости кода. Упрощение клиентского кода. Currying. Создание новых конструкций управления. Передача параметров по имени.

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.

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.

10. **Case-классы и сопоставление с образцом** (pattern matching). Простой пример. Выды образцов. Использование предохранителей в образцах. Наложение образцов. Тип Option. Более сложный пример.

11. **Работа со списками**. Списковые литералы. Тип List. Создание списков. Базовые операции над списками. Высокоуровневые методы класса List. Методы объекта List. Представление об алгоритме вывода типа в Scala.

12. **Реализация списков**. Принципы организации типа List. Класс ListBuffer.

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

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

15. **Параллелизм**. Проблемы организации параллельного выполнения кода. Концепия Actor. Actor-объекты и обмен сообщениями. Actor-объекты на основе базовых потоков. Повышение производительности Actor-объектов на основе повторного использования потоков. Учебый пример "Моделирование дискретных параллельных систем".

16. **Грамматический разбор на основе комбинаторов**. 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. Реализация комбинаторов. Строковые литералы и регулярные выражения. Лексический анализ. Сообщения об ошибках.

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

18. **Заключение**. Примеры создания Scala-систем. Akka. Примеры языков программирования, базируемых на JVM и CLR. Clojure.

4.2. Практические занятия

Практические занятия являются формой индивидуально-группового и практико-ориентированного обучения на основе реальных или модельных ситуаций применительно к виду и профилю профессиональной деятельности.

Темы практических занятий 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.

Темы практических занятий в 3-м семестре:

1. Case-классы и сопоставление с образцом.
2. КП: Выдача задания на проектирование. Анализ задания.
3. Работа со списками.
4. КП: Работа над внешним проектом системы.
5. Реализация списков.
6. КП: Защита внешнего проекта.
7. Коллекции.
8. КП: Работа над проектом архитектуры системы.
9. For-выражения.
10. КП: Работа над проектом архитектуры системы. Продолжение.
11. Параллелизм.
12. КП: Реализация.
13. Грамматический разбор на основе комбинаторов.
14. КП: Реализация. Продолжение.
15. Работа с XML.
16. КП: Защита.
17. Примеры создания Scala-систем.
18. КП: Защита.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При освоении дисциплины используются следующие сочетания видов учебной работы с методами и формами активизации познавательной деятельности бакалавров для достижения запланированных результатов обучения и формирования компетенций:

- изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;

- самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;

- закрепление теоретического материала при проведении лабораторных работ с использованием современной вычислительной техники и пакетов математического моделирования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

- самостоятельная работа студента, направленная на углубление и закрепление знаний, а также развитие практических умений заключается в работе бакалавров с лекционным материалом, поиск и анализ литературы и электронных источников информации по заданной проблеме, выполнении домашних заданий, переводе материалов из тематических информационных ресурсов с иностранных языков, изучении тем, вынесенных на самостоятельную проработку, изучении теоретического материала к лабораторным занятиям, подготовке к экзамену.

- творческая проблемно-ориентированная самостоятельная работа направлена на развитие интеллектуальных умений, комплекса универсальных (общекультурных) и профессиональных компетенций, повышение творческого потенциала бакалавров и заключается в поиске, анализе, структурировании и презентации информации, анализе научных публикаций по определенной теме, анализе статистических и фактических материалов по заданной теме, проведении расчетов, составлении схем и моделей, выполнении расчетно-графических работ, исследовательской работе и участии в научных студенческих конференциях и семинарах.

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

6.1. 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, подготовке к рейтинг-контролям, к зачету и экзамену.

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.

6.2. Оценочные средства для текущего контроля успеваемости

Вопросы к рейтинг-контролю № 1 во 2-м семестре

1) Основные понятия Scala
2) Базовые типы и операции

Вопросы к рейтинг-контролю № 2 во 2-м семестре

3) Функциональные объекты
4) Функции и замыкания

Вопросы к рейтинг-контролю № 3 во 2-м семестре

1) Композиция и наследование
2) Trait-конструкции

Вопросы к рейтинг-контролю № 1 в 3-м семестре

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

Вопросы к рейтинг-контролю № 2 в 3-м семестре

1) For-выражения
2) Параллелизм

Вопросы к рейтинг-контролю № 3 в 3-м семестре

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

6.3. Оценочные средства для промежуточной аттестации студентов по итогам освоения дисциплины

6.3.1. Вопросы к экзамену 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.

6.3.2. Вопросы к зачету 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. Концепия 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.

6.3.3. Задания к курсовому проекту

Порядок выполнения курсового проекта следующий:

1. Изучить заданную программу.
2. Воспроизвести программу задания и отладить ее. При необходимости разрешается внести в программу небольшие изменения.
3. При необходимости модифицировать программу так, чтобы процесс 0 читал входные данные из файла input.txt и выводил выходные данные в файл output.txt.
4. Составить программу генерации входа input.txt с применением генератора псевдослучайных чисел. Необходимо предусмотреть возможность варьирования объема данных в файле input.txt в зависимости от параметров на входе генератора.
5. Экспериментально исследовать производительность составленной программы и определить ее коэффициент ускорения (по отношению к однопоточному варианту).
   Объем входа должен быть таким, чтобы время выполнения программы было не менее 5 минут, или, чтобы достигался верхний предел ресурсов, выделяемых ОС для приложения.
6. Составить отчет о выполненной работе.

Варианты индивидуальных заданий:

1. Program 6.2 Cannon's Matrix-Matrix Multiplication with MPI's Topologies
2. Program 6.3 Non-Blocking Cannon's Matrix-Matrix Multiplication
3. Program 6.4 Row-wise Matrix-Vector Multiplication
4. Program 6.5 Column-wise Matrix-Vector Multiplication
5. Program 6.6 Dijkstra's Single-Source Shortest-Path
6. Program 6.7 Samplesort
7. Program 6.8 Two-Dimensional Matrix-Vector Multiplication
8. Program 6.1 Odd-Even Sorting

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Основная литература

1. Java EE 7 и сервер приложений GlassFish 4 [Электронный ресурс] / Дэвид Хеффельфингер - М. : ДМК Пресс, 2015. - http://www.studentlibrary.ru/book/ISBN9785970603321.html
2. C++ AMP: построение массивно параллельных программ с помощью Microsoft Visual C++ [Электронный ресурс] / Кэйт Грегори, Эйд Миллер - М. : ДМК Пресс, 2013. - http://www.studentlibrary.ru/book/ISBN9785940748960.html
3. Язык C в XXI веке [Электронный ресурс] / Клеменс Бен - М. : ДМК Пресс, 2015. - http://www.studentlibrary.ru/book/ISBN9785970601013.html

7.2. Дополнительная литература

1. Модели параллельного программирования [Электронный ресурс] / Федотов И.Е. - М. : СОЛОН-ПРЕСС, 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. http://www.studentlibrary.ru/book/ISBN9785940744481.html

3. Java EE 6 и сервер приложений GlassFish 3 [Электронный ресурс] / Дэвид Хеффельфингер ; Пер. с англ.: Карышев Е.Н. - М. : ДМК Пресс, 2013. - http://www.studentlibrary.ru/book/ISBN9785940749028.html

4. Языки программирования. Концепции и принципы [Электронный ресурс] / Кауфман В.Ш. - М. : ДМК Пресс, 2010. - http://www.studentlibrary.ru/book/ISBN9785940746225.html

5. Параллельное программирование на C++ в действии. Практика разработки многопоточных программ [Электронный ресурс] / Энтони Уильяме ; Пер. с англ. Слинкин А.А. - М. : ДМК Пресс, 2012. - http://www.studentlibrary.ru/book/ISBN9785940744481.html

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

1. Двуядерные ПК для выполнения информационной и информационно-поисковой работы.
2. Вычислительный кластер кафедры ВТ, который состоит из 4-х двуядерных ПК, соединенных между собой с помощью ЛВС отдельной от ЛВС университета.
3. Проектор с демонстрационным ПК.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 09.04.01 Информатика и вычислительная техника.

Рабочую программу составил: профессор кафедры ВТ Дубов И.Р.

Рецензент: ООО «Фирма Инрэко ЛАН», заместитель генерального директора, к.т.н. Отрызков В.А. (представитель работодателя)

Программа рассмотрена и одобрена на заседании кафедры вычислительной техники

Протокол № 6 от 10 февраля 2015 года

Заведующий кафедрой _____ В.И. Ланцов

(ФИО, подпись)

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

Протокол № 1 от 10 февраля 2015 года

Председатель комиссии _____ В.И. Ланцов

(ФИО, подпись)

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
Высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

УТВЕРЖДАЮ
Заведующий кафедрой

_____ _В.Н. Ланцов_____
подпись инициалы, фамилия

«_10_» _февраля___ 2015__

Основание:
решение кафедры
ОТ « 10 » ФЕВРАЛЯ 2015

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПРИ ИЗУЧЕНИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
наименование дисциплины

09.04.01 – Информатика и вычислительная техника
код и наименование направления подготовки

_____ магистратура_____
Уровень высшего образования

Владимир 2015

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств (ФОС) для текущего контроля успеваемости и промежуточной аттестации по дисциплине разработан в соответствии с рабочей программой, входящей в ООП направления подготовки 09.04.01 «Информатика и вычислительная техника».

Семестр 2

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Введение	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
2	Основные понятия Scala	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
3	Базовые типы и операции	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
4	Функциональные объекты	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
5	Встроенные конструкции управления	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
6	Функции и замыкания	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
7	Абстракция управления	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
8	Композиция и наследование	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
9	Trait-конструкции	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы

Семестр 3

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
10	Case-классы и сопоставление с образцом	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
11	Работа со списками	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы

12	Реализация списков	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
13	Коллекции	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
14	For-выражения	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
15	Параллелизм	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
16	Грамматический разбор на основе комбинаторов	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
17	Работа с XML	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы
18	Заключение	ОК-1, ОК-5, ОК-6, ОК-7, ОПК-1, ОПК-2, ОПК-6, ПК-4, ПК-6, ПК-7	Тестовые вопросы

Комплект оценочных средств предназначен для аттестации обучающихся на соответствие их персональных достижений поэтапным требованиям образовательной программы, в том числе рабочей программы дисциплины, для оценивания результатов обучения: знаний, умений, владений и уровня приобретенных компетенций. Комплект оценочных средств включает:

1. Тестовые вопросы как систему стандартизированных знаний, позволяющую провести процедуру измерения уровня знаний и умений обучающихся на практических занятиях и при проведении рейтинг-контроля по лекционному материалу.

2. Оценочные средства для проведения промежуточной аттестации в форме контрольных вопросов для проведения экзамена.

Перечень компетенций, формируемых в процессе изучения дисциплины

ОК-1: способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень,

ОК-5: использование на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом),

ОК-6 способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности,

ОК-7: способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности,

ОПК-1: способность воспринимать математические, естественнонаучные, социально-

экономические и профессиональные знания, умением самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе, в новой или незнакомой среде и в междисциплинарном контексте),

ОПК-2: культура мышления, способностью выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных их разных областей науки и техники, выносить суждения на основании неполных данных, владение методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе, в глобальных компьютерных сетях,

ОПК-6: способность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями,

ПК-3: знание методов оптимизации и умение применять их при решении задач профессиональной деятельности,

ПК-4: владение существующими методами и алгоритмами решения задач распознавания и обработки данных,

ПК-6: понимание существующих подходов к верификации моделей программного обеспечения,

ПК-7: применение перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий.

Знать	***Уметь***	***Владеть***
основы функционального программирования, основные сведения из спецификации Scala, способы программирования типовых задач в Scala, базовые средства Scala предназначенные для разработки DSL.	пользоваться спецификацией Scala, разрабатывать на Scala программы средней сложности, разрабатывать и реализовывать собственные DSL.	навыками работы с технической и справочной литературой, навыками самостоятельного освоения современных языков программирования и инструментальных средств разработки программных систем.

ПОКАЗАТЕЛИ, КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ

Текущий контроль знаний, согласно «Положению о рейтинговой системе комплексной оценки знаний студентов в ВлГУ» (далее Положение) в рамках изучения данной дисциплины предполагает тестовые вопросы как систему стандартизированных знаний, позволяющую провести процедуру измерения уровня знаний и умений обучающихся на практических занятиях и при проведении рейтинг-контроля по лекционному материалу.

Общее распределение баллов текущего контроля по видам
учебных работ для студентов (в соответствии с Положением) в 2-м семестре

Рейтинг-контроль 1	До 10 баллов
Рейтинг-контроль 2	До 10 баллов
Рейтинг контроль 3	До 8 баллов
За выполнение практических занятий	До 32 балла

Экзамен	До 40 баллов

Общее распределение баллов текущего контроля по видам
учебных работ для студентов (в соответствии с Положением) в 3-м семестре

Рейтинг-контроль 1	До 20 баллов
Рейтинг-контроль 2	До 20 баллов
Рейтинг контроль 3	До 20 баллов
За выполнение практических занятий	До 40 балла

Регламент проведения мероприятия и критерии
оценивания решения задач на практических занятиях

В целях закрепления практического материала и углубления теоретических знаний по разделам данной дисциплины в учебном плане предусмотрены лабораторные работы, что позволяет углубить процесс познания, раскрыть понимание прикладной значимости осваиваемой дисциплины.

Регламент решения задач на практических занятиях

№	Вид работы	Продолжительность
1.	Предел длительности решения задачи	5-7 мин.
2.	Внесение исправлений в представленное решение	до 2 мин.
3.	Комментарии преподавателя	до 1 мин.
	Итого (в расчете на одну задачу)	до 10 мин.

Критерии оценки решения задач на практических занятиях

Оценка	Критерии оценивания
5 баллов	задачи решены полностью, в представленном решении обоснованно получен правильный ответ.
4 балла	задачи решены полностью, но нет достаточного обоснования или при верном решении допущена вычислительная ошибка, не влияющая на правильную последовательность рассуждений, и, возможно, приведшая к неверному ответу.
2 балла	задачи решены частично.
0 баллов	решение неверно или отсутствует.

Регламент проведения рейтинг-контроля

№	Вид работы	Продолжительность
1	Предел длительности ответа на вопрос	15-25 мин.
2	Число вопросов в тесте	2

Критерии оценки ответов студентов на тестовые вопросы
рейтинг-контроля во 2-м семестре

Оценка выполнения тестов	Критерий оценки

5 баллов за правильный ответ для 1-го и 2-го рейтинг-контроля. 4 балла за правильный ответ 3-го рейтинг-контроля	Правильно записанный развернутый ответ на вопрос или правильно решенная задача.

Критерии оценки ответов студентов на тестовые вопросы
рейтинг-контроля в 3-м семестре

Оценка выполнения тестов	Критерий оценки
10 баллов за правильный ответ	Правильно записанный развернутый ответ на вопрос или правильно решенная задача.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ

Вопросы к рейтинг-контролю № 1 во 2-м семестре
5) Основные понятия Scala
6) Базовые типы и операции

Вопросы к рейтинг-контролю № 2 во 2-м семестре
1) Функциональные объекты
2) Функции и замыкания

Вопросы к рейтинг-контролю № 3 во 2-м семестре
3) Композиция и наследование
4) Trait-конструкции

Вопросы к рейтинг-контролю № 1 в 3-м семестре
3) Case-классы и сопоставление с образцом
4) Реализация списков

Вопросы к рейтинг-контролю № 2 в 3-м семестре
3) For-выражения
4) Параллелизм

Вопросы к рейтинг-контролю № 3 в 3-м семестре
3) Грамматический разбор на основе комбинаторов
4) Работа с XML

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Вопросы к экзамену 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 к зачету 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. Концепия 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

Порядок выполнения курсового проекта следующий:

1. Изучить заданную программу.
2. Воспроизвести программу задания и отладить ее. При необходимости разрешается внести в программу небольшие изменения.
3. При необходимости модифицировать программу так, чтобы процесс 0 читал входные данные из файла input.txt и выводил выходные данные в файл output.txt.
4. Составить программу генерации входа input.txt с применением генератора псевдослучайных чисел. Необходимо предусмотреть возможность варьирования объема данных в файле input.txt в зависимости от параметров на входе генератора.
5. Экспериментально исследовать производительность составленной программы и определить ее коэффициент ускорения (по отношению к однопоточному варианту).
   Объем входа должен быть таким, чтобы время выполнения программы было не менее 5 минут, или, чтобы достигался верхний предел ресурсов, выделяемых ОС для приложения.
6. Составить отчет о выполненной работе.

Варианты индивидуальных заданий:

1. Program 6.2 Cannon's Matrix-Matrix Multiplication with MPI's Topologies
2. Program 6.3 Non-Blocking Cannon's Matrix-Matrix Multiplication
3. Program 6.4 Row-wise Matrix-Vector Multiplication
4. Program 6.5 Column-wise Matrix-Vector Multiplication
5. Program 6.6 Dijkstra's Single-Source Shortest-Path
6. Program 6.7 Samplesort
7. Program 6.8 Two-Dimensional Matrix-Vector Multiplication
8. Program 6.1 Odd-Even Sorting

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на __2015/2016___ учебный год

Протокол заседания кафедры № __12__ от __2 июля 2015__ года

Заведующий кафедрой_____

Рабочая программа одобрена на __2016/2017___ учебный год

Протокол заседания кафедры № __12__ от __30 августа 2016__ года

Заведующий кафедрой_____

Рабочая программа одобрена на 2017/2018 учебный год

Протокол заседания кафедры № __1___ от 06.09.17 года

Заведующий кафедрой_____

Рабочая программа одобрена на 2018/2019 учебный год

Протокол заседания кафедры № __1___ от 14.09.18 года

Заведующий кафедрой_____

Рабочая программа одобрена на 2019/2020 учебный год

Протокол заседания кафедры № __1___ от 10.09.19 года

Заведующий кафедрой_____