

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«Луганский государственный университет имени Владимира Даля»
(ФГБОУ ВО «ЛГУ им. В. Даля»)**

**Северодонецкий технологический институт
Кафедра информационных технологий, приборостроения и электротехники**

УТВЕРЖДАЮ:
Врио. директора СТИ (филиал)
ФГБОУ ВО «ЛГУ им. В. Даля»
Ю.В. Бородач
(подпись)
« 26 » 2024 года



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Построение распределенных систем мониторинга»

По направлению подготовки: 09.03.02 «Информационные системы и технологии»

Профиль «Информационные ситемы и технологии»

Лист согласования РПУД

Рабочая программа учебной дисциплины «Построение распределенных систем мониторинга» по направлению подготовки: 09.03.02 «Информационные системы и технологии» (программа бакалавриата «Информационные ситемы и технологии») – 26 с.

Рабочая программа учебной дисциплины «Построение распределенных систем мониторинга» разработана в соответствии Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии» утвержденный приказом Министерства науки и высшего образования Российской Федерации от 19 сентября 2017 г. № 926 (с изменениями и дополнениями в соответствии с приказами Министерства образования и науки Российской Федерации №1456 от 26.11.2020 г., №83 от 08.02.2021 г., №662 от 19.07.2022 г., №208 от 27.02.2023 г.).

СОСТАВИТЕЛЬ:

старший преподаватель Кузнецова Е.В.

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий, приборостроения и электротехники « 05 » сентября 2024 г., протокол № 1.

Заведующий кафедрой ИТПЭ  В.Г. Чебан

Переутверждена: « » 20 г., протокол № .

Рекомендована на заседании учебно-методической комиссии Северодонецкого технологического института (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Луганский государственный университет имени Владимира Даля» « 16 » сентября 2024 г., протокол № 1.

Председатель учебно-методической комиссии
СТИ (филиал) ФГБОУ ВО «ЛГУ им. В.Даля»

 Ю.В. Бородач

Структура и содержание дисциплины

1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины - формирование представлений о современных распределенных вычислительных архитектурах систем мониторинга, моделях, методах и технологиях их программирования, привить навыки работы с современными распределенными вычислительными системами;

Задачи: изучение принципов функционирования и особенностей построения распределенных информационных систем мониторинга, методов организации распределенного доступа к информации; приобретение студентами базового набора знаний из области параллельных вычислений, а также первичных навыков работы с современными параллельными вычислительными системами

2. Место дисциплины в структуре ООП ВО.

Дисциплина «Технологии распределенных систем и мониторинга» входит в блок дисциплин обязательной части учебного плана.

Содержание дисциплины является логическим продолжением содержания дисциплин «Технологии обработки информации», «Базы данных», «Web-программирование и web-дизайн», «Кроссплатформенное программирование», «Архитектура информационных систем и облачных технологий» и служит основой для освоения дисциплины «Администрирование баз данных ORACLE»

3. Требования к результатам освоения содержания дисциплины

Студенты, завершившие изучение дисциплины «Технологии распределенных систем и параллельных вычислений», должны

знать : классификацию информационных систем, методы распределенной обработки информации и управления, технологии построения распределенных информационных систем, разновидности архитектур распределенных информационных систем, среды передачи данных; локальные вычислительные сети; методы коммутации и маршрутизации; протоколы стека TCP/IP, принципы организации многопоточного и параллельного выполнения программ, способы взаимодействия параллельно выполняемых программ, основные стандарты в области инфокоммуникационных систем и технологий, современные стандартные фреймворки параллельного и многопоточного программирования, теоретические основы архитектурной и системотехнической организации вычислительных сетей, основные тенденции развития параллельных архитектур, факторы, влияющие на производительность, критерии выбора программно-аппаратной платформы для решения вычислительно-сложных задач заданного класса

уметь: использовать системные и прикладные программные средства для проектирования формирования распределенных информационных систем из разнородных компонентов, адаптация распределенных информационных систем и их компонентов к изменяющимся условиям функционирования, устанавливать, тестировать, испытывать и использовать программно-аппаратные средства вычислительных и информационных систем, ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, работать с современными системами программирования, включая объектно-ориентированные, решать задачи многопоточного программирования с использованием пакета `java.util.concurrent`, пользоваться средствами удаленного доступа к вычислительным ресурсам коллективного пользования и запуска параллельных программ на вычислительных кластерах;

владеть: навыками программирования, проектирования АС и ИС, подключения компьютера к локальной сети, настройки и диагностики сетей и сетевого оборудования; языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ,, методами описания схем баз данных, навыками конфигурирования локальных сетей, реализации сетевых протоколов с помощью программных средств, методами и средствами разработки и оформления технической документации.

Перечисленные результаты образования являются основой для формирования следующих компетенций (в соответствии с ФГОС ВО и требованиями к результатам освоения основной профессиональной образовательной программы (ОПОП ВО):

общефессиональными

ОПК-7 Знать: основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем.

ОПК-7.1 Знать: основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем.

ОПК-7.2 Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем, применять современные технологии реализации информационных систем.

ОПК-7.3 Иметь навыки: владения технологиями и инструментальными программно-аппаратными средствами для реализации

информационных систем.

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (з.е.)		
	Очная форма	Очно-заочная форма	Заочная форма
Объем учебной дисциплины (всего)	72 (2 з.е.)	-	72 (2 з.е.)
Обязательная аудиторная учебная нагрузка дисциплины (всего) в том числе:	42	-	-
Лекции	14	-	6
Семинарские занятия	-	-	-
Практические занятия	-	-	-
Лабораторные работы	28	-	6
Курсовая работа (курсовой проект)	-	-	-
Индивидуальное задание	-	-	-
Самостоятельная работа студента (всего)	30	-	60
Форма аттестации	зачет	--	зачет

4.2. Содержание разделов дисциплины

Тема 1. Введение в архитектуру распределенных информационных систем.

Типовые архитектуры распределенных ИС; многозвенные системы; системы с «тонким» и «толстым» клиентом. Централизованная, файл-серверная, клиент-серверная архитектуры ИС. Архитектуры «хост-терминал», Интранет, одноранговая (P2P), Grid-систем, мультиагентных систем. Разработка компонентов прикладного и представления данных учебной ИС с клиент-серверной архитектурой.

Тема 2. Введение в параллельные вычисления

Цели и задачи параллельной обработки данных. Необходимость и актуальность параллельных вычислений.

Различия между многозадачным, параллельным и распределенным режимами выполнения программ. Закон Амдаля. Закон Мура. Гипотеза Минского. Способы построения многопроцессорных вычислительных систем. Краткая история развития высокопроизводительных вычислений. Примеры параллельных вычислительных систем. Рейтинги ведущих суперкомпьютеров: мировой TOP-500, TOP-50 СНГ.

Тема 3. Обзор параллельных вычислительных систем и их классификация. Способы организации параллельной обработки данных

Систематика Флинна. Детализация систематики Флинна.

Понятия мультипроцессора, мультикомпьютера, вычислительного кластера. Особенности организации параллельных вычислений в системах с общей памятью (обеспечение однозначности кэш-памяти разных процессоров, синхронизация вычислений). Особенности организации параллельных вычислений в системах с распределенной памятью посредством передачи сообщений. Топологии сетей передачи данных в мультикомпьютерах. Типовые схемы коммуникации.

Тема 4. Вычислительные кластеры: основные понятия, архитектура, типовый набор кластерного программного обеспечения, средства доступа и управления, тестирование производительности

Понятие кластера и кластерной архитектуры. Классификация кластерных вычислительных систем. Состав сетевой инфраструктуры кластера. Типы топологий и критерии эффективности коммуникационной сети кластера. Сетевые решения для кластерных систем. Основные критерии оценки кластерных систем. Типичный набор программно-аппаратного обеспечения кластеров. Особенности запуска задач на кластерах. Системы управления заданиями. Интегрированные наборы кластерного программного обеспечения.

Тема 5. Моделирование параллельных программ. Реализация параллелизма различного вида. Общая схема и методика разработки параллельных алгоритмов

Показатели эффективности параллельного алгоритма и оценка максимально достижимого параллелизма. Параллелизм на примере модельных задач нахождения частных сумм последовательности числовых значений и умножения матриц. Общая схема и методика разработки параллельных алгоритмов. Пример использования методики разработки параллельных алгоритмов для параллельного решения гравитационной задачи N тел.

Тема 6. Базовые средства параллельного программирования вычислительных кластеров. Методы передачи данных. Стандарт MPI

Общая характеристика методов передачи данных, оценка времени выполнения коммуникационных операций. Оценка трудоемкости операций передачи данных для кластерных систем. Модель Хокни. MPI: основные понятия и определения. Базовый (минимальный) набор функций MPI, достаточный для разработки параллельных программ. Пример: программа вычисления числа π . Операции передачи данных между двумя процессами Коллективные операции передачи данных. Упаковка и распаковка разнотипных данных в MPI. Управление группами процессов и коммутаторами. Виртуальные топологии Модельный пример: умножение матрицы на вектор.

Тема 7. Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP

Общие сведения. Структура стандарта OpenMP. Достоинства технологии OpenMP. Модель параллелизма OpenMP. Модель памяти OpenMP. Директивы OpenMP. Типы директив. Формат записи директив. Определение параллельной области. Распределение вычислений между потоками. Директивы синхронизации. Директивы управления областью видимости данных. Совместимость директив и их параметров. Библиотека функций OpenMP. Функции для контроля/запроса параметров среды исполнения. Функции синхронизации. Переменные среды исполнения. Пример программы произведения матриц. Сравнение технологий MPI и OpenMP для SMP-систем. Гибридный (MPI+OpenMP) подход для SMP-кластеров. Компиляторы Intel с поддержкой OpenMP. Инструментальные средства разработки и отладки многопоточных приложений.

Тема 8. Высокоуровневые средства программирования многопроцессорных систем. DVM-система.

DVM-система. Общие сведения, цели создания, принципы построения. Модель параллелизма, модель выполнения и модель программирования DVM. Языки программирования DVM. Директивы DVM (на примере языка C-DVM). Сравнение размеров и эффективности MPI- и DVM-программ. Переносимость и повторное использование DVM-программ. Средства функциональной отладки, анализа и прогноза производительности DVM-программ. Особенности компиляции и запуска DVM-программ. Примеры программ на языке C-DVM.

Тема 9. Высокопроизводительные вычисления с применением графических процессоров (GPU). Технология NVidia CUDA.

Введение в вычисления общего назначения с использованием GPU. Основные архитектурные отличия GPU от CPU. Архитектура современных GPU. Технология CUDA. Модели и шаблоны программирования с использованием технологии CUDA. Модель памяти CUDA. Типы памяти. Оптимизация CUDA-приложений. Модель исполнения CUDA. Компиляция CUDA-приложений. CUDA-расширение языка C (спецификаторы функций, спецификаторы переменных, встроенные переменные, директивы запуска ядра). Некоторые функции API CUDA Runtime. Пример программы на CUDA. Произведение матриц. Ускорение Matlab-расчетов на GPU.

4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1	Введение в технологии распределенных систем и параллельных вычислений.	1		2
2	Распределенные файловые системы	1		2
3	Распределенные базы данных.	1		2

4	Модель распределенных вычислений.	1		
5	Операционные системы мультимикропроцессорных ЭВМ..	1		
6	Принципы организации интерфейса MPI.	1		
7	Технология программирования OpenMP.	1		
8	Система параллельного программирования Linda.	1		
9	Коммуникации в распределенных системах.	1		
10	Программирование сетевых соединений в Java.	1		
11	Потоки выполнения в Java.	2		
12	Пакет java.util.concurrent	2		
Итого:		14		6

4.4. Практические (семинарские) занятия

Практические занятия по дисциплине не предусмотрены.

4.5. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	Очно- заочная форма	Заочная форма
1	Методы создания потоков в Java.	2		2
2	Получение и установка параметров потоков.	2		2
3	Исследование методов прерывания потоков.	2		2
4	Работа с исключениями в потоках	2		2
5	Приостановка работы потока	2		
6	Ожидание завершения работы потока	2		
7	Создание и запуск потоков-демонов	2		
8	Обработка исключительных ситуаций в потоках	2		
9	Использование локальных переменных потока	2		
10	Создание групп потоков	2		
11	Обработка непроверяемых исключений в группе потоков	2		
12	Создание потоков при помощи фабрики	2		
13	Синхронизация методов и участков кода	2		
14	Обработка независимых атрибутов в синхронизированных классах	2		
Итого:		28		6

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	Очно- заочная форма	Заочная форма
1	Введение в технологии распределенных систем и параллельных вычислений.	подготовка к лабораторным работам и оформление отчетов	2,5		5
2	Распределенные файловые системы	подготовка к лабораторным работам и оформление отчетов Написание реферата	2,5		5
3	Распределенные базы данных.	подготовка к лабораторным работам и оформление отчетов Написание контрольной работы	2,5		5
4	Модель распределенных вычислений.	подготовка к лабораторным работам и оформление отчетов	2,5		5
5	Операционные системы мультипроцессорных ЭВМ..	подготовка к лабораторным работам и оформление отчетов	2,5		5
6	Принципы организации интерфейса MPI.	подготовка к лабораторным работам и оформление отчетов	2,5		5
7	Технология программирования OpenMP.	подготовка к лабораторным работам и оформление отчетов	2,5		5
8	Система параллельного программирования Linda.	подготовка к лабораторным работам и оформление отчетов	2,5		5
9	Коммуникации в распределенных системах.	подготовка к лабораторным работам и оформление отчетов	2,5		5
10	Программирование сетевых соединений в Java.	подготовка к лабораторным работам и оформление отчетов	2,5		5
11	Потоки выполнения в Java.	подготовка к лабораторным работам и оформление отчетов	2,5		5
12	Пакет java.util.concurrent	подготовка к лабораторным работам и оформление отчетов	2,5		5
Итого:			30		60

4.7. Курсовые работы/проекты.

Курсовые работы по дисциплине не предусмотрены.

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий:

- традиционные объяснительно-иллюстративные технологии, которые обеспечивают доступность учебного материала для большинства студентов, системность, отработанность организационных форм и привычных методов, относительно малые затраты времени;
- технологии проблемного обучения, направленные на развитие познавательной активности, творческой самостоятельности студентов и предполагающие последовательное и целенаправленное выдвижение перед студентом познавательных задач, разрешение которых позволяет студентам активно усваивать знания (используются поисковые методы; постановка познавательных задач);
- технологии развивающего обучения, позволяющие ориентировать учебный процесс на потенциальные возможности студентов, их реализацию и развитие;
- технологии концентрированного обучения, суть которых состоит в создании максимально близкой к естественным психологическим особенностям человеческого восприятия структуры учебного процесса и которые дают возможность глубокого и системного изучения содержания учебных дисциплин за счет объединения занятий в тематические блоки;
- технологии модульного обучения, дающие возможность обеспечения гибкости процесса обучения, адаптации его к индивидуальным потребностям и особенностям обучающихся (применяются, как правило, при самостоятельном обучении студентов по индивидуальному учебному плану);
- технологии дифференцированного обучения, обеспечивающие возможность создания оптимальных условий для развития интересов и способностей студентов, в том числе и студентов с особыми образовательными потребностями, что позволяет реализовать в культурно-образовательном пространстве университета идею создания равных возможностей для получения образования
- технологии активного (контекстного) обучения, с помощью которых осуществляется моделирование предметного, проблемного и социального содержания будущей профессиональной деятельности студентов (используются активные и интерактивные методы обучения) и т.д.

Максимальная эффективность педагогического процесса достигается путем конструирования оптимального комплекса педагогических технологий и (или) их элементов на личностно-ориентированной, деятельностной, диалогической основе и использования необходимых современных средств обучения.

6. Формы контроля освоения дисциплины

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем(ями), ведущими практические занятия по дисциплине в следующих формах:

- лабораторные работы;
- защита лабораторных работ

Фонды оценочных средств, включающие вопросы к защите лабораторных работ, позволяющие оценить результаты текущей и промежуточной аттестации обучающихся по данной дисциплине, помещаются в приложении к рабочей программе в соответствии с «Положением о фонде оценочных средств».

Форма аттестации по результатам освоения дисциплины проходит в форме зачета (включает в себя защиту всех предусмотренных программой лабораторных работ). Студенты, выполнившие 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачетную книжку выставляются оценки по национальной шкале, приведенной в таблице.

Характеристика знания предмета и ответов	Зачеты
Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

7. Учебно-методическое и программно-информационное обеспечение дисциплины:

а) основная литература

1. Кирсанов Э.А., Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы / Кирсанов Э.А., Сирота А.А. - М. : ФИЗМАТЛИТ, 2012. - 344 с. - ISBN 978-5-9221-1420-2 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785922114202.html>. - Режим доступа : по подписке.
2. Крищенко В.А., Технологии создания кросс-платформенных распределенных приложений : Учеб. пособие / Крищенко В.А. - М. : Издательство МГТУ им. Н. Э. Баумана, 2009. - 40 с. - ISBN 978-5-7038-3316-2 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785703833162.html>. - Режим доступа : по подписке.
3. Миренков Н. Н. Параллельное программирование для многомодульных вычислительных систем [Текст] / Н. Н. Миренков. - М. : Радио и связь, 1989. - 320 с. - 681:3 - М633 (4)
4. Горбатов В. А. Логическое управление распределенными системами [Текст] / В. А. Горбатов, М. И. Смирнов, И. С. Хлытчиев ; под ред. В. А. Горбатова. - М. : Энергоатомиздат, 1991. - 287 с. - 681.3 - Г671

5.
б) дополнительная литература:

1. Малявко А.А., Параллельное программирование на основе технологий OpenMP, MPI, CUDA : учеб. пособие / Малявко А.А. - Новосибирск : Изд-во НГТУ, 2015. - 116 с. - ISBN 978-5-7782-2614-2 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785778226142.html>. - Режим доступа : по подписке.
2. Цегелик Г. Г. Системы распределенных баз данных [Текст] / Г. Г. Цегелик. - Львов : СВІТ, 1990. - 167 с. - 681.3 - Ц299
3. Уильямс Э., Параллельное программирование на C++ в действии. Практика разработки многопоточных программ / Энтони Уильямс ; Пер. с англ. Слинкин А.А. - М. : ДМК Пресс, 2012. - 672 с. - ISBN 978-5-94074-448-1 - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <http://www.studentlibrary.ru/book/ISBN9785940744481.html>. - Режим доступа : по подписке.

в) методические указания:

1. Методические указания к лабораторным занятиям по дисциплине «Построение распределенных систем мониторинга» для студентов, обучающихся по направлению подготовки 09.03.02 – «Информационные системы и технологии» [Электронный ресурс] / сост. С. С. Стоянченко. - Луганск : ЛНУ им. В. Даля, 2019. - 77 с

г) Интернет-ресурсы:

1. Министерство образования и науки Российской Федерации – <http://минобрнауки.рф/>
 2. Федеральная служба по надзору в сфере образования и науки – <http://obrnadzor.gov.ru/>
 3. Министерство образования и науки Луганской Народной Республики – <https://minobr.su>
 4. Народный совет Луганской Народной Республики – <https://nslnr.su>
 5. Портал Федеральных государственных образовательных стандартов высшего образования – <http://fgosvo.ru>
 6. Федеральный портал «Российское образование» – <http://www.edu.ru/>
 7. Информационная система «Единое окно доступа к образовательным ресурсам» – <http://window.edu.ru/>
 8. Федеральный центр информационно-образовательных ресурсов – <http://fcior.edu.ru/>
- Электронные библиотечные системы и ресурсы
1. Электронно-библиотечная система «Консультант студента» – <http://www.studentlibrary.ru/cgi-bin/mb4x>
 2. Электронно-библиотечная система «StudMed.ru» – <https://www.studmed.ru>
- Информационный ресурс библиотеки образовательной организации
1. Научная библиотека имени А. Н. Коняева – <http://biblio.dahluniver.ru/>

8. Материально-техническое обеспечение дисциплины

Освоение дисциплины «Построение распределенных систем мониторинга» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Лекционные занятия: комплект электронных презентаций/слайдов; аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук).

Лабораторные работы: компьютерная аудитория, оснащенная компьютерами с установленным специализированным программным обеспечением.

Прочее: рабочее место преподавателя, оснащенное компьютером с доступом в Интернет, проектор, экран, рабочие места студентов, оснащенные компьютерами с доступом в Интернет, предназначенные для работы в электронной образовательной среде.

Программное обеспечение

Функциональное назначение	Бесплатное программное	Ссылки

	обеспече ние	
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционн ая система	UBUNT U 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Набор инструмент ов для разработки программ на Java	jdk1.7	https://www.oracle.com/technetwork/java/javase/downloads/index.html
Интегриров анная среда разработки, отладки и сборки программно го обеспечения	Netbeans	http://netbeans.apache.org/
Архиватор	7Zip	http://www.7-zip.org/
Редактор PDF	PDFCreat or	http://www.pdfforge.org/pdfcreator
Аудиоплеер	VLC	http://www.videolan.org/vlc/

1. Паспорт
фонда оценочных средств по учебной дисциплине
«Построение распределенных систем мониторинга»

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые разделы (темы) учебной дисциплины	Этапы формирования (семестр изучения)
1	2	3	4	5
1	ОПК-7	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем	Тема 1. Введение в архитектуру распределенных информационных систем. Тема 2. Введение в параллельные вычисления. Тема 3. Обзор параллельных вычислительных систем и их классификация. Способы организации параллельной обработки данных. Тема 4. Вычислительные кластеры: основные понятия, архитектура, типовый набор кластерного программного обеспечения, средства доступа и управления, тестирование производительности. Тема 5. Моделирование параллельных программ. Реализация параллелизма различного вида. Общая схема и методика разработки параллельных алгоритмов. Тема 6. Базовые средства параллельного программирования вычислительных кластеров. Методы	7

			передачи данных. Стандарт MPI. Тема 7. Базовые средства параллельного программирования систем с общей памятью. Стандарт OpenMP. Тема 8 8. Высокоуровневые средства программирования многопроцессорных систем. DVM-система. Тема 9. Высокопроизводительные вычисления с применением графических процессоров (GPU). Технология NVidia CUDA.	
--	--	--	--	--

Примечание:

* Наименование темы (раздела) или тем (разделов) берется из рабочей программы учебной дисциплины (программы практики) в соответствии с компетенциями, которые формируются у студента при ее изучении.

2. Показатели и критерии оценивания компетенций, описание шкал оценивания

Таблица 2.1 – Показатели оценивания компетенций

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые разделы (темы) учебной дисциплины (модуля), практики ¹	Наименование оценочного средства ²
1	2	3	4	5
1	ОПК-7	Знать: основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем Уметь: осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации	Тема 1. Введение в архитектуру распределенных информационных систем. Тема 2. Введение в параллельные Тема 3. Обзор параллельных вычислительных систем и их классификация. Способы организации параллельной обработки данных. вычисления.	Лабораторная работа

		информационных систем, применять современные технологии реализации информационных систем. Иметь навыки: владения технологиями и инструментальными программно-аппаратными средствами для реализации информационных систем		
--	--	---	--	--

Примечание:

¹ - Контролируемые разделы (темы) учебной дисциплины, перечисленные в рабочей программе, показывающие этапность формирования компетенции в ходе изучения учебной дисциплины.

² - Наименование оценочного средства в соответствии с их примерным перечнем.

Вопросы для защиты лабораторных работ

Лабораторная работа №1. Методы создания потоков в Java.

Цель работы: приобрести навыки разработки многопоточных программ в Java.

Контрольные вопросы:

1. Какие методы создания потоков в Java Вы знаете? Опишите каждый метод.
2. С какой целью используется метод run() класса Thread? Укажите его сигнатуру.
3. Как запустить поток на выполнение? Какой метод для этого используется?
4. Укажите условия завершения многопоточной Java программы

Лабораторная **работа №2. Получение и установка параметров потоков.**

Цель работы: изучить состав параметров потоков и методы работы с ними.

Контрольные вопросы:

1. Перечислите информационные поля потока.
2. Укажите диапазон допустимых значений приоритета потока (класса Thread).
3. Какое значение соответствует самому низкому приоритету?
4. Какое значение соответствует самому высокому приоритету?
5. В каких состояниях может находиться поток?

Лабораторная работа №3. Исследование методов прерывания потоков.

Цель работы: приобрести навыки разработки программ прерывания потоков.

Контрольные вопросы:

1. Назовите условия завершения работы многопоточной Java программы
2. Что такое поток – демон?

3. Каким образом поток может отреагировать на прерывание с требованием завершения его работы?
4. Какие методы потока используются для проверки факта его прерывания?
5. Что общего и в чем различие методов **isInterrupted()** и **interrupted()**?
6. Может ли поток проигнорировать требование своего завершения? Какая реакция потока на запрос завершения работы является рекомендуемой?

Лабораторная работа №4. Работа с исключениями в потоках

Цель работы: Приобрести навыки обработки прерываний в потоках в Java приложениях.

Контрольные вопросы:

1. Какое исключение предусматривается в Java для использования в потоках?
2. Для какой цели используется исключение **InterruptedException**?
3. В каком методе потока рекомендуется обрабатывать исключение **InterruptedException**?
4. Назовите методы библиотеки Java, которые могут бросать исключение **InterruptedException**.
5. Объясните механизм работы рекурсии, который использован в лабораторной работе.

Лабораторная работа №5. Приостановка работы потока

Цель работы: работы: приобрести навыки использования в Java программного механизма приостановки работы потока.

Контрольные вопросы:

1. Для чего используется класс **TimeUnit**?
2. Какие у него есть атрибуты для указания отрезков времени?
3. Для чего используется метод **sleep()**?
4. Что является рекомендуемой практикой программирования при обработке прерываний остановки потока?
5. Как связан вызов метода **sleep()** с использованием потоком процессорного времени?
6. Какое исключение может бросить метод **sleep()**? Назовите условия бросания этого исключения. Исключение бросается сразу при возникновении необходимых условий или поток ожидает своего выхода из «состояния сна»?
7. Для каких целей используется метод **yield()**?

Лабораторная работа №6. Ожидание завершения работы потока

Цель работы: приобрести навыки разработки многопоточного программного обеспечения с использованием аппарата ожидания завершения работы потока.

Контрольные вопросы:

1. Какому классу принадлежит метод **join()**?
2. Опишите назначение метода **join()**.
3. Какие варианты метода **join()** Вы знаете?
4. Опишите структуру и алгоритм работы программы, разработанной в рамках данной лабораторной работы

Лабораторная работа №7. Создание и запуск потоков-демонов

Цель работы: приобрести навыки разработки многопоточного программного обеспечения с использованием потоков-демонов.

Контрольные вопросы:

1. Для решения каких задач используются потоки- демоны?
2. Назовите характеристики (особенности) потока-демона
3. Когда завершаются потоки демоны? (Назовите условия завершения).

4. Как создается поток-демон?
5. Для чего можно использовать метод **isDaemon()**?
6. Какой приоритет имеют потоки-демоны?

Лабораторная работа №8. Обработка исключительных ситуаций в потоках

Цель работы: приобрести навыки разработки многопоточных приложений с использованием обработчиков исключительных ситуаций.

Контрольные вопросы:

1. Дайте характеристику классификации исключений в Java.
2. Укажите способы обработки **Checked** исключений в программе?
3. Укажите способы обработки **Unchecked** исключений в программе?
4. Дайте описание способов обработки прерываний в потоках Java.
5. Укажите алгоритм обработки непроверяемых исключений в потоках Java.

Лабораторная работа №9. Использование локальных переменных потока

Цель работы: приобрести навыки разработки многопоточных приложений с использованием локальных переменных потока.

Контрольные вопросы:

1. Опишите проблему разделения полей в потоках. Для каких потоков эта проблема актуальна?
2. Укажите способ преодоления проблемы разделения переменных в потоках?
3. Для какой цели используется механизм **thread-local**? Приведите примеры его использования.

Лабораторная работа №10. Создание групп потоков

Цель работы: приобрести навыки разработки многопоточных приложений с использованием групп потоков.

Контрольные вопросы:

1. Для какой цели используется класс **ThreadGroup** ?
2. Как указать, что поток относится к данной группе потоков?
3. Перечислите методы класса **ThreadGroup**. Для этого проконсультируйтесь в документации на классы Java.
4. Как в рамках лабораторной работы решена задача принудительного завершения работы потоков, входящих в группу?
5. Для какой цели в программе используется объект класса **Random**?
6. Каким образом в программе обнаруживается факт завершения работы одного (первого) потока?

Лабораторная работа №11. Обработка непроверяемых исключений в группе потоков

Цель работы: приобрести навыки разработки многопоточных приложений с поддержкой возможности обработки непроверяемых исключений в группе потоков.

Контрольные вопросы:

1. Какие виды исключений предусмотрены в Java. Укажите отличия **Checked** и **Unchecked** исключений.
2. Опишите особенности обработки исключений в группах потоков **ThreadGroup**.
3. Охарактеризуйте особенности работы программы, созданной в рамках данной лабораторной работы.

Лабораторная работа №12. Создание потоков при помощи фабрики

Цель работы: приобрести навыки разработки многопоточных приложений, в которых потоки создаются на основе паттерна «фабрика».

Контрольные вопросы:

1. Опишите особенности паттерна «фабрика».
2. Как паттерн «фабрика» используется для создания объектов- потоков?
3. Какие параметры получает метод `newThread`? Какой результат он возвращает?
4. Перечислите преимущества использования паттерна «фабрика» для создания объектов – потоков.

Лабораторная работа №13. Синхронизация методов и участков кода

Цель работы: приобрести навыки разработки многопоточных приложений с использованием синхронизированных методов.

Контрольные вопросы:

1. Опишите проблемы доступа к общим данным в многопоточном приложении.
2. Что такое «критический участок» и как этот механизм реализован в Java?
3. Что такое синхронизированный метод? Как такой метод описывается в Java? Опишите порядок работы конкурирующих потоков в синхронизированном методе?
4. Укажите особенности статических синхронизированных методов.
5. Если один поток захватил один синхронизированный метод объекта, может ли другой поток захватить другой синхронизированный метод этого же объекта?
6. Объясните работу программы, разработанной в рамках данной лабораторной работы, с использованием синхронизированных методов и без использования синхронизированных методов.
7. Что такое синхронизированный участок кода? Какой параметр нужно передавать синхронизированному участку кода?

Лабораторная работа №14. Обработка независимых атрибутов в синхронизированных классах

Цель работы: приобрести навыки разработки многопоточных приложений с реализацией безопасного одновременного доступа к независимым атрибутам объектов.

Контрольные вопросы:

1. Опишите метод ограничения доступа в многопоточном приложении к участку кода программы.
2. Опишите алгоритм работы программы, созданной в рамках данной лабораторной работы.
3. Каким образом осуществляется доступ к двум независимым полям объекта в программе?
4. С какой целью в синхронизированный блок кода передается параметр – объект?

Лабораторная работа №15. Использование условий в синхронизированных критических участках многопоточного приложения.

Цель работы: приобрести навыки разработки многопоточных приложений с использованием условий в синхронизированных критических участках.

Контрольные вопросы:

1. Опишите назначение методов **`wait()`**, **`notify()`**, **`notifyAll()`**.
2. Опишите особенности использования методов **`wait()`**, **`notify()`**, **`notifyAll()`**. Как эти методы соотносятся с ключевым словом `synchronize`.
3. Опишите задачу производитель - потребитель
4. Дайте характеристику алгоритма работы программы, созданной в рамках лабораторной работы.
5. Опишите алгоритм работы методов **`get()`** и **`set()`**.

Лабораторная работа №16. Синхронизация выполнения блока программы с помощью механизма защепок (Lock).

Цель работы: приобрести навыки разработки многопоточных приложений с использованием синхронизированных критических участков, построенных на основе защепок.

Контрольные вопросы:

1. Опишите преимущества использования защепок для поддержания синхронизированного доступа к критическим участкам кода.
2. Укажите разницу в работе методов **tryLock()** и **lock()**.
3. Дайте характеристику алгоритма работы программы, созданной в рамках лабораторной работы.
4. Как производится освобождение защелки потоком?
5. Что такое тупиковая ситуация? Приведите пример возникновения тупиковой ситуации.

Лабораторная работа №17. Синхронизация доступа к данным в многопоточных приложениях с помощью механизма защепок чтения / записи.

Цель работы: приобрести навыки разработки многопоточных приложений с использованием синхронизированных критических участков, построенных на основе защепок чтения / записи.

Контрольные вопросы:

1. Опишите особенности защепок для выполнения операций чтения и записи.
2. Укажите разницу в интерфейсах Lock и ReadWriteLock.
3. Дайте характеристику алгоритма работы программы, созданной в рамках лабораторной работы.
4. Какой класс имплементирует интерфейс ReadWriteLock? Какие методы реализованы в этом классе?

Таблица 2.2 – Критерии и шкала оценивания по оценочному средству «Лабораторная работа»

Шкала оценивания (интервал баллов) ²	Критерий оценивания
5	Лабораторная работа выполнена самостоятельно на высоком уровне и в полном объеме, отчет оформлен в соответствии с требованиями, сделаны правильные выводы по проведенным экспериментам.
4	Лабораторная работа выполнена самостоятельно на среднем уровне и в полном объеме, отчет оформлен с незначительными отклонениями от требований, допущены незначительные неточности в выводах по проведенным экспериментам
3	Лабораторная работа выполнена на низком уровне и не полностью, отчет оформлен с отклонениями от требований, выводы по экспериментам сделаны не в полном объеме.
2	Лабораторная работа не выполнена, отчет не оформлен, или представленный отчет не соответствует варианту задания.

Пример тем лабораторных работ.

1. Введение в параллельное программирование .Visual Studio 2010
2. Параллельное программирование в .NET Framework 4.0
3. Параллелизм доступа к данным.

4. Потокобезопасные коллекции
5. Создание Windows приложения с использованием элемента BackgroundWorker
6. Язык интегрированных запросов Parallel LINQ (PLINQ)
7. Параллельные вычисления на языке Си

Оценочные средства для итоговой аттестации (зачет)

Перечень вопросов и заданий для зачета.

1. Теоретические вопросы.
 1. Понятия процесса и потока
 2. Класс Thread и интерфейс Runnable
 3. Взаимодействие потоков
 4. Особенности использования методов wait и notify
 5. Синхронизация потоков
 6. Поток-демон
 7. Приостановка потока
 8. Использование метода join
 9. Взаимная блокировка потоков
 10. Методы interrupt, interrupted, isInterrupted
 11. «Гонка» в многопоточном приложении
 12. «Голодание» в многопоточном приложении
 2. Пакет синхронизации java.util.concurrent
 13. Объекты синхронизации пакета java.util.concurrent
 14. Объект синхронизации Semaphore
 15. Объект синхронизации CountdownLatch
 16. Объект синхронизации CyclicBarrier
 17. Объект синхронизации Exchanger
 18. Объект синхронизации Phaser
 19. Потокобезопасные коллекции
 20. Атомарные классы пакета util.concurrent
 21. Блокирующие и неблокирующие очереди
 22. Блокировки пакета concurrent
 23. Какой метод переводит поток в режим ожидания?
 24. Какова функциональность методов notify и notifyAll?
 25. Что позволяет сделать метод join?
 26. Каковы условия вызова метода wait/notify?
 27. Дайте определение понятию “взаимная блокировка”.
 28. Чем отличаются методы interrupt, interrupted, isInterrupted?
 29. В каком случае будет выброшено исключение InterruptedException, какие методы могут его выбросить?
 30. Модификаторы volatile и метод yield().
 31. Пакет java.util.concurrent
 32. Есть некоторый метод, который выполняет операцию i++. Переменная i типа int. Предполагается, что код будет исполняться в многопоточной среде. Следует ли синхронизировать блок?
 33. Что используется в качестве mutex, если метод объявлен static synchronized? Можно ли создавать новые экземпляры класса, пока выполняется static synchronized метод?
 34. Предположим в методе run возник RuntimeException, который не был пойман. Что случится с потоком? Есть ли способ узнать о том, что Exception произошел (не заключая все тело run в блок try-catch)? Есть ли способ восстановить работу потока после того как это произошло?

35. Какие стандартные инструменты Java вы бы использовали для реализации пула потоков?
36. Что такое ThreadGroup и зачем он нужен?
37. Что такое ThreadPool и зачем он нужен?
38. Что такое ThreadPoolExecutor и зачем он нужен?
39. Что такое «атомарные типы» в Java?
40. Зачем нужен класс ThreadLocal?
41. Что такое Executor?
42. Что такое ExecutorService?
43. Зачем нужен ScheduledExecutorService?

2. Практические вопросы.

1. В методе main создать фиксированный пул из 5 трэдов используя класс Executors.
2. В методе main создать очередь LinkedBlockingQueue<Runnable>. В цикле добавлять в очередь 10 задач Runnable. У каждой задачи в методе run вызывать метод doExpensiveOperation с порядковым номером задачи начиная с 1.

Типовой экзаменационный билет

ГОУ ВПО «ЛУГАНСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Кафедра Информационные и управляющие системы

Семестр 7

Экзаменационный контроль

Дисциплина «Построение распределенных систем мониторинга»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Особенности использования методов wait и notify.
2. Что используется в качестве mutex, если метод объявлен static synchronized? Можно ли создавать новые экземпляры класса, пока выполняется static synchronized метод?
3. В методе main создать фиксированный пул из 5 трэдов используя класс Executors.

Утверждено на заседании кафедры _____

Протокол № _____

Зав. кафедрой _____ **Преподаватель** _____

Таблица 2.3 – Критерии и шкала оценивания по оценочному средству промежуточный контроль (зачет)

Характеристика знания предмета и ответов	Зачеты
Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено

Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков

Требования к выполнению контрольных заданий определены Методическими указаниями для лабораторных работ по дисциплине «Построение распределенных систем мониторинга», (электронное издание) (для студентов по специальности 09.03.02 Информационные системы и технологии).

Контрольные сроки защиты лабораторных работ определены графиком учебного процесса, в частности, лабораторные работы 1-5 должны быть защищены в первой половине семестра, 6-10 до начала экзаменационной сессии.

Требования к выполнению индивидуального задания определены методическими указаниями к выполнению индивидуального задания по дисциплине «Построение распределенных систем мониторинга», направление подготовки 09.03.02 Информационные системы и технологии (для студентов дневного и заочного обучения).

Контрольный срок сдачи индивидуального задания – до начала экзаменационной сессии.

Итоговая аттестация проводится в виде зачета по билетам, содержащим 2 теоретических вопроса и одно практическое задание.

4. Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр ¹), на котором были рассмотрены и одобренны изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Примечание:

¹ - для ФОС по государственной итоговой аттестации указываются реквизиты протоколов заседания кафедр и подписи заведующих кафедрами, деканов/директоров, совместно реализующих ОП