




МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**


«СОГЛАСОВАНО»

Руководитель ОП Разработка программно-информационных систем по направлению 09.04.04 Программная инженерия

  
(подпись) Артемяева И.Л.  
(Ф.И.О. рук. ОП) «21» 07 2018 г.



«УТВЕРЖДАЮ»  
Заведующая кафедрой прикладной математики,  
механики, управления и программного обеспечения

  
(подпись) Артемяева И.Л.  
(Ф.И.О. зав. каф.) «21» 07 2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Параллельная обработка данных

Направление подготовки – 09.04.04 Программная инженерия

Магистерская программа «Разработка программно-информационных систем»

Форма подготовки (очная)

курс 2 семестр 3

лекции 18 час.

практические занятия 0 час.

лабораторные работы 18 час.

в том числе с использованием МАО лек. \_\_\_\_ /пр. \_\_\_\_ /лаб. \_\_\_\_ час.

в том числе в электронной форме лек. \_\_\_\_ /пр. \_\_\_\_ /лаб. \_\_\_\_ час.

всего часов аудиторной нагрузки 36 час.

в том числе с использованием МАО \_\_\_\_ час.

в том числе контролируемая самостоятельная работа 0 час.

в том числе в электронной форме \_\_\_\_ час.

самостоятельная работа 72 час.

в том числе на подготовку к экзамену 0 час.

курсовая работа / курсовой проект \_\_\_\_\_ семестр

зачет 3 семестр

экзамен не предусмотрен

Рабочая программа составлена в соответствии с требованиями образовательного стандарта, самостоятельно устанавливаемого ДВФУ, утвержденного приказом ректора от 07.07.2015 № 12-13-1282

Рабочая программа обсуждена на заседании кафедры прикладной математики, механики, управления и программного обеспечения, протокол № 7.2 от 21.07.2018 г.

Заведующая кафедрой прикладной математики, механики, управления и программного обеспечения д.т.н., профессор Артемяева И.Л.

Составитель (ли): профессор каф. Прикладной математики, механики, управления и программного обеспечения ДВФУ Гриняк В. М., д.т.н. доцент

**Оборотная сторона титульного листа РПУД**

**I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ И.Л. Артемьева  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ И.Л. Артемьева  
(подпись) (И.О. Фамилия)

## ABSTRACT

**Master's degree in** 09.04.04 – Software engineering

**Master's Program** “Development of software and information systems”

**Course title:** parallel data processing

**Variable part of Block 1, \_3\_credits**

**Instructor:** Grinyak V.

**At the beginning of the course a student should be able to:** analyze the problems and the development trends of the technology of programming; apply main methods and tools of the automation of the design, the production and the evaluation tests of software in a professional activity; know and apply the main conceptual regulations of functional, logic, object-oriented and visual kinds of programming and also the methods and the tools of the software design in these kinds; know and apply methods of the design and the production of software; know about the principles of the structure and the operation of the tools used for the creation of software

**Learning outcomes:** an ability to lead free scientific and profession communication in a foreign community; sufficient knowledge of at least one foreign language for social and professional communication, an ability to use a professional lexicon; an ability to design distributed information systems, its components and communication protocols; an ability to design network services; possession of skills in the program realization of distributed information systems

**Course description:** structures, client specifics, efficiency and design patterns of distributed systems

### **Main course literature:**

1. DiMarzio J.F. Quick Start Guide to JavaFX. – New York: Oracle Press. – 2014.  
[https://www.logobook.ru/prod\\_show.php?object\\_uid=12502540#](https://www.logobook.ru/prod_show.php?object_uid=12502540#)
2. Ebbers H. Mastering JavaFX 8 Controls. – New York: Oracle Press. – 2014.  
<http://www.allitebooks.com/mastering-javafx-8-controls/>
3. Schildt H. Introducing JavaFX 8 Programming. – New York: Oracle Press. – 2015.  
<http://java-buddy.blogspot.com/2015/09/introducing-javafx-8-programming-oracle.html>

**Form of final knowledge control:** Examination

## **Аннотация рабочей программы учебной дисциплины «Параллельная обработка данных»**

Рабочая программа дисциплины «Параллельная обработка данных» разработана для студентов 2 курса, обучающихся по направлению 09.04.04 Программная инженерия, магистерская программа «Разработка программно-информационных систем». Дисциплина является обязательной дисциплиной вариативной части учебного плана Б1.В.01.02.

Трудоемкость дисциплины 3 зачетных единицы (108 часов). Дисциплина реализуется в 3 семестре. Учебным планом предусмотрено: 18 часов лекций, 18 часов лабораторных работ, 72 часа самостоятельной работы.

Дисциплина «Параллельная обработка данных» базируется на дисциплинах: «Методология научных исследований в программной инженерии», «Современные языки и системы программирования». Знания, полученные при ее изучении, будут использованы при подготовке выпускных квалификационных работ.

**Цель дисциплины** – на основе актуальной научной литературы в области высокопроизводительных и распределенных вычислительных технологий изучить и научиться использовать алгоритмы и методы параллельной обработки данных для прикладных задач; изучить соответствующее программное обеспечение, библиотеки, пакеты программ и инструментальные средства; научиться применять современные суперкомпьютеры в проводимых исследованиях.

### **Задачи дисциплины:**

1. Изучение концептуальных и теоретических моделей, используемых при создании параллельных программных систем
2. Изучение методов создания параллельных систем для решения прикладных задач
3. Изучение существующих инструментальных средств, используемых при создании параллельных систем.

Для успешного изучения дисциплины «Параллельная обработка данных» у обучающихся должны быть сформированы следующие предварительные компетенции:

способность к самоорганизации и самообразованию;  
способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности, способностью к

определению общих форм и закономерностей отдельной предметной области.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций (общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций)):

Код и формулировка компетенции	Этапы формирования компетенции	
ПК-3 знанием методов оптимизации и умением применять их при решении задач профессиональной деятельности	Знает	основы математического аппарата описания процессов
	Умеет	Применять математический аппарат для описания параллельных процессов
	Владеет	навыками разработки параллельных программных систем
ПК-9 способностью проектировать системы с параллельной обработкой данных и высокопроизводительные системы, и их компоненты	Знает	основные методы разработки параллельных систем, их достоинства и недостатки
	Умеет	применять методы разработки параллельных систем
	Владеет	методами разработки параллельных систем и подходами к их верификации

Для формирования вышеуказанных компетенций в рамках дисциплины «Параллельная обработка данных» применяются следующие методы активного/ интерактивного обучения: метод активного диалога и метод проектов.

## I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

### **Lesson 1. Introduce Rich Client Applications (Тема 1. Сущность клиентских приложений) (1 час).**

Describe an overview of the BrokerTool application. Explain the problem statement of the BrokerTool project.

Пример клиентского приложения. Основные проблемы построения клиентских приложений.

### **Lesson 2. JavaFX. (Тема 2. Платформа JavaFX) (1 час).**

Describe the features of JavaFX. Identify the features of the JavaFX scene graph. Describe the JavaFX development tools. Describe how JavaFX is integrated into a Java application. Use FXML and determine when to use it in an application. Download and install JavaFX and related samples

Функциональность платформы JavaFX. Средства разработчика платформы JavaFX. Особенности интеграции средств JavaFX в приложения Java. Использование средства FXML. Установка платформы JavaFX.

### **Lesson 3. Generics and JavaFX Collections (Тема 3. Обобщённые типы и коллекции платформы JavaFX) (1 час).**

Generics and JavaFX Collections: Java Collections Framework, javafx.collections package. ObservableList: A list that enables listeners to track changes when they occur. ObservableMap: A map that enables observers to track changes when they occur. ListChangeListener: An interface that handles list-related events generated by an ObservableList.

Обобщённые типы и специфические коллекции значений платформы JavaFX: пакет javafx.collections, обобщённые типы ObservableList, ObservableMap, ListChangeListener.

### **Lesson 4. UI Controls, Layouts, Charts, and CSS. (Тема 4. Средства построения пользовательского интерфейса платформы JavaFX: управляющие элементы, панели, графики и их взаимное расположение) (1 час).**

Relating UI components to the scene graph. Describing and implementing JavaFX UI components such as controls, images, shapes, and layout containers. Using CSS . Adding events to JavaFX components. Creating charts.

Использование компонент пользовательского интерфейса для прорисовки интерфейса приложения. Взаимное расположение компонент. Примеры компонент пользовательского интерфейса: управляющие элементы, картинки, графики, контейнеры. Обработка событий элементов визуального интерфейса.

### **Lesson 5. Visual Effects, Animation, WebView, and Media. (Тема 5. Визуальные эффекты, анимация, просмотр веб-элементов, отражение медиа-файлов) (1 час).**

Using animation and effects in an application. Describing how to implement media in an application. Describing the benefits of using WebView.

Использование анимации и визуальных эффектов в клиентских приложениях. Реализация медиа-контента в клиентских приложениях. Преимущества использования средств WebView.

**Lesson 6. JavaFX Tables and Client GUI. (Тема 6. Таблицы платформы JavaFX и их реализация в графическом интерфейсе клиентских приложений) (1 час).**

Create a table and custom table cell. Apply CSS to a table. Recognize JavaFX development practices. Describe the BrokerTool application interface. Identify the JavaFX components and charts to use in the BrokerTool interface.

Создание таблиц и настройка особенностей их отображения. Использование средств настройки стиля для описания таблиц. Пример интерфейса приложения. Использование компонент платформы JavaFX и графиков при разработке пользовательского интерфейса.

**Lesson 7. JavaFX Concurrency and Binding. (Тема 7. Параллельные вычисления и связывание в платформе JavaFX) (1 час).**

Describe and implement JavaFX concurrency. Describe binding and properties in JavaFX, including simple binding and bi-directional binding.

Описание и реализация параллельных вычислений в платформе JavaFX. Описание и особенности связи параллельных процессов: простая и двусторонняя связь.

**Lesson 8. Java Persistence API (JPA). (Тема 8. Программный интерфейс Java Persistence) (1 час).**

Describe the Java Persistence API (JPA). Define Object-Relational Mapping (ORM) and how JPA provides a framework to support ORM. Use JPA to create, read, update, and delete database entities. Create and use queries in JPA.

Описание программного интерфейса Java Persistence API. Фреймворки Java Persistence API, определение связей между объектами. Использование Java Persistence API для создания, чтения, обновления и удаления объектов баз данных. Создание и реализация запросов в Java Persistence API.

**Lesson 9. Applying the JPA. (Тема 9. Использование Java Persistence API) (1 час).**

Identify relationships in an application. Build and deploy a JPA application in a Java SE environment. Apply a two-tier design in the HenleyApp application.

Определение связей в приложении. Построение и развёртывание приложений на основе JPA в платформе Java SE. Пример приложения с двухслойной архитектурой.

**Lesson 10. Implementing a Multi-tier Design with RESTful Web Services. (Тема 10. Реализация многослойных приложений с помощью инструментария RESTful Web Services) (1 час).**

Compare the HenleyApp two-tier design and the HenleyApp three-tier design. Describe a RESTful web service. List the web services used in the HenleyApp application. Describe how the RESTful web services were developed in the HenleyServer application.

Особенности реализации приложений с двух- и трёхслойной архитектурой. Инструментарий RESTful web service. Перечень веб-сервисов, используемых при разработке приложений с многослойной архитектурой. Пример использования инструментов RESTful web service.

**Lesson 11. Connecting to a RESTful Web Service. (Тема 11. Интеграция приложений на основе инструментария RESTful Web Service) (1 час).**

Describe how to test a RESTful Web Service. Identify how to develop a Jersey RESTful Client. Review the implementation of web service clients in the HenleyApp application.

Особенности тестирования функциональности RESTful Web Service. Пример разработки клиентского приложения. Обзор особенностей реализации клиентских приложений на основе RESTful Web Service. Пример клиентского приложения на основе RESTful Web Service.

**Lesson 12. Packaging and Deploying Applications. (Тема 12. Использование пакетов и развертывание приложений) (1 час).**

Describe a JAR file and the steps for creating one. Create a manifest file that uses headers. Create a JAR by using development tools. Deploy a stand-alone JAR. Deploy a JAR as an applet. Deploy a JAR by using Java Web Start.

JAR файлы и их создание. Создание файлов, использующих заголовки. Создание JAR файлов вручную и в различных средах разработки. Развертывание stand-alone JAR файлов. Развертывание JAR файлов как апплетов. Развертывание JAR файлов посредством Java Web Start.

**Lesson 13. Developing Secure Applications. (Тема 13. Развертывание приложений: вопросы безопасности) (1 час).**

Describe aspects of security. Describe fundamental software security concepts. Avoid common injection and inclusion attacks. Avoid common confidential data errors. Limit class accessibility, extensibility, and mutability.



Define immutability. Create classes that are immutable. List security resources available on the Internet.

Различные аспекты безопасности приложений. Фундаментальные основы безопасности программного обеспечения. Противодействие проникновению и атакам. Предотвращение доступа к закрытым данным. Ограничение доступности классов, их расширяемости и изменчивости. Определение неизменяемости классов и данных. Создание неизменяемых классов.

#### **Lesson 14. Signing an Application and Authentication. (Тема 14. Подписывание приложений и их аутентификация) (1 час).**

Describe cryptography, encryption, and cipher. Differentiate a symmetric key from an asymmetric key. Describe public/private key encryption. Describe digital certificates. Describe hash algorithms. Sign an application by using Java tools. Describe how SSL works. Explain the security concepts that are applied in BrokerTool.

Криптография, шифрование и цифровая подпись. Симметричные и асимметричные ключи. Открытый и закрытый ключ. Цифровой сертификат. Алгоритмы хеширования. Подписывание приложений с использованием средств Java. Особенности работы SSL. Пример реализации различных аспектов защиты приложения.

#### **Lesson 15. Logging. (Тема 15. Логирование) (1 час).**

Add a logging mechanism to your Java application. Log messages to a file at an appropriate level. Configure your logger. Configure logging formatters and handlers.

Включение различных механизмов логирования в приложения Java. Сообщения-логи и их запись в файл. Конфигурирование средств логирования.

#### **Lesson 16. Implementing Unit Testing and Using Version Control. (Тема 16. Модульное тестирование и контроль версий) (1 час).**

Set up a unit test system. Write test cases. Apply JUnit test framework. Run unit tests against source code. Create a test suite. Use a version control system.

Установка системы модульного тестирования. Написание тестовых сценариев. Использование инструментария JUnit test framework. Запуск модульных тестов. Использование систем контроля версий.

**Lesson 17. Development Methodologies and Design Patterns. (Тема 17. Методы разработки и шаблоны проектирования) (1 час).**

Agile Development. Design Patterns.

Технология разработки программного обеспечения Agile Development. Шаблоны проектирования программного обеспечения.

**Lesson 18. JavaFX History and Architecture. (Тема 18. История развития платформы JavaFX и её архитектура) (1 час).**

JavaFX History. JavaFX Architecture.

История развития платформы JavaFX. Архитектура платформы JavaFX и её место среди платформ технологии Java.

**II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ  
КУРСА**

**Лабораторные работы (18 час.)**

**Practices for Lesson 1. Introduce Rich Client Applications (Лабораторная работа 1. Сущность клиентских приложений) (1 час).**

Setting Up Your Development Environment. Exploring the BrokerTool Application. Running the Ensemble Sample Application. Running the Henley Application.

Установка среды разработчика. Обзор приложения BrokerTool. Запуск приложения Ensemble Sample. Запуск приложения Henley.

**Practices for Lesson 2. JavaFX. (Лабораторная работа 2. Платформа JavaFX) (1 час).**

Creating a Login Window by Using JavaFX. Creating an FXML Login Window.

Создание приложения «Окно для ввода логина и пароля» с использованием средств платформы JavaFX. Создание окна для ввода логина и пароля с помощью FXML.

**Practices for Lesson 3. Generics and JavaFX Collections (Лабораторная работа 3. Обобщённые типы и коллекции платформы JavaFX) (1 часа).**

Adding a Second Listener to Your List.

Добавление слушателей событий в список.

**Practices for Lesson 4. UI Controls, Layouts, Charts, and CSS. (Лабораторная работа 4. Средства построения пользовательского**

**интерфейса платформы JavaFX: управляющие элементы, панели, графики и их взаимное расположение) (1 час).**

Creating an Order Form by Using FXML. Adding Events to the Order Form. Creating an Interactive Pie Chart.

Создание упорядоченной формы с помощью FXML. Добавление событий на упорядоченную форму. Создание интерактивной диаграммы.

**Practices for Lesson 5. Visual Effects, Animation, WebView, and Media. (Лабораторная работа 5. Визуальные эффекты, анимация, просмотр веб-элементов, отражение медиа-файлов) (1 час).**

Adding a Fade Transition.

Добавление плавно разворачивающейся анимации.

**Practices for Lesson 6. JavaFX Tables and Client GUI. (Лабораторная работа 6. Таблицы платформы JavaFX и их реализация в графическом интерфейсе клиентских приложений) (1 час).**

Creating a Simple TableView. Styling a Smart Table. Creating a Complete BrokerTool Interface.

Создание таблиц и настройка особенностей их отображения. Создание завершённого интерфейса приложения BrokerTool.

**Practices for Lesson 7. JavaFX Concurrency and Binding. (Лабораторная работа 7. Параллельные вычисления и связывание в платформе JavaFX) (1 час).**

Displaying Service State Information. Adding a Second Service to your Application.

Отображение информации о состоянии сервисов. Добавление сервисов в приложение.

**Practices for Lesson 8. Java Persistence API (JPA). (Лабораторная работа 8. Программный интерфейс Java Persistence) (1 час).**

Creating Entity Classes by Using JPA. Implementing CRUD Operations by Using JPA.

Создание классов сущностей с использованием Java Persistence API.. Создание и реализация запросов в Java Persistence API.

**Practices for Lesson 9. Applying the JPA. (Лабораторная работа 9. Использование Java Persistence API) (1 час).**

Identifying Entity Relationships in the BrokerTool Application. Implementing Database Connectivity in the BrokerTool Application by Using the JPA.

Определение связей в приложении BrokerTool. Реализация базы данных и соединения с ней с помощью средств Java Persistence API.

**Practices for Lesson 10. Implementing a Multi-tier Design with RESTful Web Services. (Лабораторная работа 10. Реализация многослойных приложений с помощью инструментария RESTful Web Services) (1 час).**

Reviewing Basic Concepts of Java Web Services. Examining BrokerToolServer's Web Services.

Обзор основных положений Java Web Services. Проверка работы веб-сервисов в приложении BrokerToolServer.

**Practices for Lesson 11. Connecting to a RESTful Web Service. (Лабораторная работа 11. Интеграция приложений на основе инструментария RESTful Web Service) (1 час).**

Testing RESTful Web Services. Creating a RESTful Web Services Client.

Тестирование RESTful Web Service. Создание клиента для RESTful Web Service.

**Practices for Lesson 12. Packaging and Deploying Applications. (Лабораторная работа 12. Использование пакетов и развертывание приложений) (1 час).**

Deploying an Application in a Web App. Deploying an Embedded Application Only (Browser Only). Deploying Java Web Start Only.

Развертывание приложений как приложений веб. Развертывание приложений как встроенных. Развертывание приложений в режиме веб-старт.

**Practices for Lesson 13. Signing an Application and Authentication. (Лабораторная работа 13. Подписывание приложений и их аутентификация) (2 часа).**

Signing an Application by Using keytool and jarsigner. Signing and Deploying TextViewer by Using NetBeans.

Подписывание приложений с использованием инструментов keytool и jarsigner. Подписывание и развертывание приложений в среде NetBeans.

**Practices for Lesson 14. Logging. (Лабораторная работа 14. Логирование) (2 часа).**

Logging in Java Applications.

Логирование приложений Java.

**Practices for Lesson 15. Implementing Unit Testing and Using Version Control. (Лабораторная работа 15. Модульное тестирование и контроль версий) (2 часа).**

Creating and Executing Test Cases. Writing Test Cases Using Additional JUnit 4 Annotations. Creating a Test Suite. Performing Parameterized Testing.

Разработка и прогон тестовых сценариев. Разработка тестов с использованием JUnit 4. Создание тестовых наборов. Параметрическое тестирование.

**III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Объём самостоятельной работы по дисциплине составляет 72 часа.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Параллельная обработка данных» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

**IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА**

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Lesson 1. Introduce Rich Client Applications (Тема 1. Сущность клиентских приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (1)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (1)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (1)

2	Lesson 2. JavaFX. (Тема 2. Платформа JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (1)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (1)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (1)
3	Lesson 3. Generics and JavaFX Collections (Тема 3. Обобщённые типы и коллекции платформы JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (2)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (2)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (2)
4	Lesson 4. UI Controls, Layouts, Charts, and CSS. (Тема 4. Средства построения пользовательского интерфейса платформы JavaFX: управляющие элементы, панели, графики и их взаимное расположение)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (3)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (3)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (3)
5	Lesson 5. Visual Effects, Animation, WebView, and Media. (Тема 5. Визуальные эффекты, анимация, просмотр веб-элементов, отражение медиа-файлов)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (4)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (4)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (4)
6	Lesson 6. JavaFX Tables and Client GUI. (Тема 6. Таблицы платформы JavaFX и их реализация в графическом интерфейсе клиентских приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (5)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (5)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (5)
7	Lesson 7. JavaFX Concurrency and Binding. (Тема 7. Параллельные вычисления и связывание в платформе JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (6, 7)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (6, 7)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (6, 7)
8	Lesson 8. Java Persistence API (JPA). (Тема 8. Программный интерфейс Java Persistence)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (8)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (8)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (8)
9	Lesson 9. Applying the JPA. (Тема 9. Использование Java Persistence API)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (9)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (9)
			Владения	С/работа	Экзамен (ПР-1)

				(ПР-9)	(9)
10	Lesson 10. Implementing a Multi-tier Design with RESTful Web Services. (Тема 10. Реализация многослойных приложений с помощью инструментария RESTful Web Services)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (10)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (10)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (10)
11	Lesson 11. Connecting to a RESTful Web Service. (Тема 11. Интеграция приложений на основе инструментария RESTful Web Service)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (11, 12)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (11, 12)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (11, 12)
12	Lesson 12. Packaging and Deploying Applications. (Тема 12. Использование пакетов и развертывание приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (13)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (13)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (13)
13	Lesson 13. Developing Secure Applications. (Тема 13. Развертывание приложений: вопросы безопасности)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (14, 15)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (14, 15)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (14, 15)
14	Lesson 14. Signing an Application and Authentication. (Тема 14. Подписывание приложений и их аутентификация)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (16, 17)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (16, 17)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (16, 17)
15	Lesson 15. Logging. (Тема 15. Логирование)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (18)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (18)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (18)
16	Lesson 16. Implementing Unit Testing and Using Version Control. (Тема 16. Модульное тестирование и контроль версий)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (19)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (19)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (19)
17	Lesson 17. Development Methodologies and Design Patterns. (Тема 17. Методы разработки и шаблоны проектирования)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (20)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (20)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (20)

18	Lesson 18. JavaFX History and Architecture. (Тема 18. История развития платформы JavaFX и её архитектура)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (21, 22)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (21, 22)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (21, 22)

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

## V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Основная литература

*(электронные и печатные издания)*

1. Курняван Б. Программирование web приложений на языке Java. М.: Лори, 2009. 880 с. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:382737&theme=FEFU>
2. Мухамедзянов Р.Р. Java. Серверные приложения / Солон-Пресс. 2007. 336с. <http://www.iprbookshop.ru/65089.html?replacement=1>
3. Хеффельфингер Д. Разработка приложений Java EE 6 в NetBeans 7 / ДМК Пресс. 2013. 330с. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785940749141.html>
4. Монахов В.В. Язык программирования Java и среда NetBeans / Интернет-университет информационных технологий. 2009. 635с. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=234170&razdel=276>
5. Гонсалвес, Энтони. Изучаем Java EE 7 / Энтони Гонсалвес; [пер. с англ. Е. Зазноба и др.]. М: Питер, 2016 - 640 с. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:821123&theme=FEFU>
6. Свистунов, А. Н. Построение распределенных программных систем на Java: учебное пособие / А. Н. Свистунов. М: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний. –2014. 279 с. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:797993&theme=FEFU>



**Дополнительная литература**  
(печатные и электронные издания)

1. Молчанова Л.А., Прудникова Л.И. Java в примерах и задачах: учеб.-метод. пособие [для вузов]. Владивосток: Изд-во Тихоокеанского экономического университета. – 2011. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:359168&theme=FEFU>
2. Баженова И.Ю. Языки программирования: учебник для высшего профессионального образования. Под редакцией В.А. Сухомлина. М.: Академия. – 2012. 358 С. Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:668317&theme=FEFU>
3. Горнаков С.Г. Программирование мобильных телефонов на Java 2 Micro Edition: учебное пособие / ДМК Пресс. 2008. 511с. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=232028&razdel=276>

**Перечень ресурсов информационно-телекоммуникационной сети  
«Интернет»**

1. [http://books.ifmo.ru/book/1746/programmirovanie\\_na\\_yazyke\\_Java\\_konsp\\_ekt\\_lekciy.htm](http://books.ifmo.ru/book/1746/programmirovanie_na_yazyke_Java_konsp_ekt_lekciy.htm) Гаврилов А.В., Клименков С.В., Цопа Е.А. Программирование на Java. Конспект лекций / НИУ ИТМО. 2010. 130с. Режим доступа:
2. <http://docs.oracle.com/javase/tutorial/index.html> The Java Tutorials - [Электронный ресурс]. Режим доступа: -
3. [https://www.logobook.ru/prod\\_show.php?object\\_uid=12502540#](https://www.logobook.ru/prod_show.php?object_uid=12502540#) DiMarzio J.F. Quick Start Guide to JavaFX. – New York: Oracle Press. – 2014.
4. Ebbers H. Mastering JavaFX 8 Controls. – New York: Oracle Press. – 2014. <http://www.allitebooks.com/mastering-javafx-8-controls/>
5. 3. <http://java-buddy.blogspot.com/2015/09/introducing-javafx-8-programming-oracle.html> Schildt H. Introducing JavaFX 8 Programming. – New York: Oracle Press. – 2015.
6. <http://www.ibm.com/developerworks/ru/java/> IBM developerWorks Россия: Технология Java - [Электронный ресурс].
7. <http://docs.oracle.com/javase/7/docs/api/index.html> Java™ Platform, Standard Edition 7 API Specification - [Электронный ресурс].

## **Перечень информационных технологий и программного обеспечения**

Лекции проводятся с использованием проектора и мультимедийного комплекса для проведения лекций внутренней системы портала ДВФУ. При выполнении лабораторных работ используются интегрированные среды разработчика Eclipse, NetBeans, Visual Studio. Система контроля версий Git. Офисное ПО.

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **Особенности проведения лабораторных работ по дисциплине (активные и интерактивные формы)**

Лабораторные работы по дисциплине проводятся в форме семинара.

Студенты выполняют лабораторные работы, после чего по очереди представляют получившийся у них результат. Проводится обсуждение результата работы каждого студента. Голосованием выбирается лучшая работа по теме. Её автор получает бонус в виде дополнительных баллов.

### **Рекомендации по планированию и организации времени, отведенного на изучение дисциплины**

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит перечень основных изучаемых тем, основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании каждой темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

- 1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине.

### **Рекомендации по работе с литературой**

1. Всю учебную литературу желательно изучать «под конспект». Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

2. При работе над литературой обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

3. При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте.

### **Требования к представлению и оформлению результатов самостоятельной работы**

Самостоятельная работа включает в себя разработку распределённого приложения шаблонного типа. В рамках самостоятельной работы студенты должны разбиться на группы по 2-3 человека и на основе технологии JavaFX разработать распределённое приложение - базу данных и приложение к ней (на языке Java), реализующие простую систему торгового учета.

Результаты самостоятельной работы представляются на итоговом семинаре в виде доклада (презентации) и оформляются в виде документации к разработанному программному средству.

### **Критерии оценки выполнения самостоятельной работы**

Критерии оценки выполнения самостоятельной работы – достижение правильного результата при разработке информационной системы, качество оформления документации и представления работы на семинаре.

### **Подготовка к промежуточной аттестации по дисциплине: экзамену**

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали уверенные знания в ходе выполнении лабораторных работ.

Непосредственная подготовка к аттестации осуществляется по темам, представленным в рабочей учебной программе. Тщательно изучите содержание каждой темы.

Экзамен проводится в тестовой форме. Тест составлен на базе материалов профессионального сертификационного экзамена Oracle Certified Associate, Java SE 7 Programmer. Примеры тестовых вопросов см. в приложении 2.

### **Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины**

**Текущая аттестация студентов.** Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме представления результатов выполнения лабораторных работ и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме собеседования;
- уровень овладения практическими умениями и навыками – оценивается в форме оценки качества и правильности выполнения лабораторных работ и самостоятельной работы.

**Промежуточная аттестация студентов.** Промежуточная аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрен экзамен, который проводится в тестовой форме.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Лекции проводятся с использованием проектора и внутренней системы портала ДВФУ. Лабораторные занятия проходят в аудиториях, оборудованных компьютерами типа Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 и аудиовизуальными средствами проектор Panasonic DLPPjectorPT-D2110XE, плазма LG FLATRON M4716CCBAM4716CJ. Для выполнения самостоятельной работы студенты в жилых корпусах ДВФУ обеспечены Wi-Fi.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ  
ОБУЧАЮЩИХСЯ**  
по дисциплине «Параллельная обработка данных»  
**Направление подготовки – 09.04.04 Программная инженерия**  
Магистерская программа «Разработка программно-информационных систем»  
**Форма подготовки (очная)**

**Владивосток  
2018**

## План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-2 неделя	Спроектировать структуру базы данных в части функциональности «Поступление товаров»	9 часа	Проект
2	3-4 неделя	Реализовать базу данных и интерфейс для таблиц-справочников	9 часа	Проект
3	5-6 неделя	Реализовать интерфейс для таблиц-документов	9 часа	Проект
4	7-8 неделя	Реализовать процедуру проведения документов и форму отчёта «остатки товаров»	9 часа	Проект
5	9-10 неделя	Спроектировать структуру базы данных в части учета продаж	9 часа	Проект
6	11-12 неделя	Реализовать интерфейс для таблицы-документа «Продажи»	9 часов	Проект
7	13-14 неделя	Реализовать процедуру проведения документа «Продажи» и форму отчета «Продажи»	9 часов	Проект
8	14-18 неделя	Подготовка к зачету	9 часов	зачет
		Всего	72 часа	

### Методические указания по выполнению самостоятельной работы по дисциплине «Параллельная обработка данных»

В рамках самостоятельной работы студенты должны разбиться на группы по 2-3 человека и на основе технологии JavaFX разработать распределённое приложение - базу данных и приложение к ней (на языке Java), реализующие простую систему торгового учета.

Система должна содержать следующие таблицы-справочники (как минимум):

- Номенклатура
- Контрагенты

Система должна содержать следующие таблицы-документы (как минимум):

- Поступление товаров (с табличной частью «Товары»)
- Реализация товаров (с табличной частью «Товары»)

Состав таблиц-регистров системы должен позволять вести (как минимум):

- Складской учет в разрезе номенклатуры, количественного и суммового остатка товаров.
- Учет продаж в разрезе номенклатуры, контрагентов, количества, суммы выручки, себестоимости и прибыли. При подсчете себестоимости необходимо реализовать партионный учет товаров в рамках учетной политики ФИФО или ЛИФО.

Система должна позволять строить следующие отчеты (как минимум):

- Остатки товара (на конкретную дату). В отчете отображается номенклатура, партия, количество, сумма.
- Продажи (за конкретный период от ... до ...). В отчете отображаются номенклатура, партия, контрагент, количество, выручка, себестоимость, прибыль. Выводится общий итог по выручке, себестоимости, прибыли.

Под таблицей-справочником понимается таблица, хранящая информацию о некоторых объектах учета. Как правило, уникальным идентификатором такой таблицы является код элемента, а основным полем – наименование (см. рисунок с примером справочника «Номенклатура»). Простые справочники имеют только форму списка и не имеют иерархии, сложные – кроме формы списка имеют форму элемента, для них характерна иерархия.

Краткое наименование	↓	Код
Товары		00-00000183
Бытовая техника		00-00000040
Кондитерские изделия		00-00000076
Сигареты		00-00000164
Доска 3000x200x20		00-00000055
Доска 4000x300x40		00-00000057
Мазут светлый		00-00000099
Минеральное масло		00-00000110
Минеральные удобрения		00-00000006
Пшеница твердая, 1 сорт		00-00000149
Сульфат аммония		00-00000177
Телефон Meizu MX		00-00000182

Рис. 1. Пример формы списка справочника «Номенклатура» с иерархией.

Минеральное масло (Номенклатура)

[Главное](#)
[Цены](#)
[Правила определения счетов учета](#)
[Еще...](#)

Краткое наименование:

Полное наименование:

Группа:

Услуга

Единица:    Тонна

% НДС:

Номенклатурная группа:

Производитель:    Страна происхождения:

Импортёр:    Номер ГТД:

[Производство](#)

Статья затрат:    Спецификация:

[Коды](#)

Рис. 2. Пример формы элемента справочника «Номенклатура»

Под таблицей-документом понимается таблица, хранящая информацию о каких-либо событиях, произведённых операциях. Документы имеют номер (уникальный идентификатор), дату-время, несколько реквизитов, отражающих их функциональность (например, ссылку на контрагента). Для документов характерен также особый реквизит булевого типа «Проведение документа». Если документ проведён, то считается, что его информация отражена в учете, то есть сформированы соответствующие записи в таблицах-регистрах. Если документ не проведён, то он существует в базе только сам по себе, его информация в учёте не отражается. Переключение реквизита «Проведение документа» всегда связано с процедурой формирования записей в таблицах-регистрах базы данных.

Поступление товаров и услуг

Контрагент:

Дата	Номер	Контрагент	Сумма	Валюта
22.01.2013 12:00:02	ШИ00-000001	Торговый дом "Ко...	123 605,00	руб.
23.01.2013 0:00:00	ТД00-000003	База "Поставка ко...	42 480,00	руб.
25.01.2013 11:02:37	МН00-000001	Торговый дом "Ко...	109 150,00	руб.
31.01.2013 23:59:59	КС00-000004	МГТС	1 416,00	руб.
31.01.2013 23:59:59	КС00-000005	База "Инвентарь"	70 800,00	руб.
31.01.2013 23:59:59	0000-000002	База "Инвентарь"	82 600,00	руб.
01.02.2013 11:52:57	КП00-000006	База "Продукты"	194 100,00	руб.
02.02.2013 12:00:01	МН00-000004	Торговый дом "Ко...	196 000,00	руб.
05.02.2013 12:00:22	МН00-000002	База "Продукты"	6 726,00	руб.
07.02.2013 12:00:01	КС00-000008	База "Поставка ко...	413 472,00	руб.
10.02.2013 12:30:01	МН00-000003	База "Продукты"	115 463,00	руб.
11.02.2013 16:53:04	ТД00-000004	Конфетпром	525 000,00	руб.
14.02.2013 12:00:00	ТД00-000005	База "Продукты"	56 750,00	руб.



Рис. 3. Пример формы списка документа

У многих документов имеется так называемая табличная часть. Табличная часть документа – это отдельная таблица, хранящая перечень объектов учета, обрабатываемых документом. С точки зрения базы данных строки таблицы – табличной части всегда хранят ссылку на тот документ, к которому они привязаны.

Поступление товаров и услуг КПО0-000006 от 01.02.2013 11:52:57 (Товары)

Провести и закрыть    Записать    Провести    Печать    Создать на основании

Накладная №:    от:    Организация:

Номер: КПО0-000006    от: 01.02.2013 11:52:57    Склад:

Контрагент: База "Продукты"    Расчеты:

Договор: 15 от 15.03.2012

N	Номенклатура	Количество	Цена	Сумма	% НДС	НДС	Всего
1	Сахарный песок	2 000,000	25,00	50 000,00	10%	4 545,45	50 000,00
2	Молоко в бидоне	1 050,000	26,00	27 300,00	10%	2 481,82	27 300,00
3	Мука	800,000	19,00	15 200,00	10%	1 381,82	15 200,00
4	Масло в коробке	400,000	254,00	101 600,00	18%	15 498,31	101 600,00

Рис. 4. Пример формы элемента документа с табличной частью

Таблицы-регистры – это таблицы, хранящие информацию, предназначенную для последующего формирования отчетных форм. Как правило, эта информация формируется при проведении документов. Записи в таблицы-регистры добавляются при проведении документов и удаляются при отмене проведения документов. Структура таблиц-регистров определяется прежде всего теми отчетными формами, которые предполагается формировать системой.

Дата	Дебет	Кредит
22.01.2013	41.01    Кол.:    150,000	60.01
1	Ассорти (конфеты)	Торговый дом "Комплексный"
	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02	Розн 53
	Торговый зап	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02
22.01.2013	41.01    Кол.:    150,000	60.01
2	Барбарис (конфеты)	Торговый дом "Комплексный"
	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02	Розн 53
	Торговый зап	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02
22.01.2013	41.01    Кол.:    250,000	60.01
3	Белочка (конфеты)	Торговый дом "Комплексный"
	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02	Розн 53
	Торговый зап	Поступление товаров и услуг ШИ00-000001 от 22.01.2013 12:00:02

Рис. 5. Пример таблицы-регистра с движениями, сформированными по итогам проведения документа

Период: 01.01.2013 – 30.09.2015 Счет: 41.01 Торговый дом "Комплексный"

Сформировать Показать настройки Печать Регистр учета

ЗАО "Торговый дом Комплексный"							
Оборотно-сальдовая ведомость по счету 41.01 за Январь 2013 г. - Сентябрь 2015 г.							
Счет Номенклатура	Показатели	Сальдо на начало периода		Обороты за период		Сальдо на конец периода	
		Дебет	Кредит	Дебет	Кредит	Дебет	Кредит
41.01	БУ			5 081 102,46	4 212 836,07	868 266,39	
	Кол.			35 880,000	22 180,000	13 700,000	
Ассорти (конфеты)	БУ			57 838,99	48 940,69	8 898,30	
	Кол.			650,000	550,000	100,000	
Барбарис (конфеты)	БУ			79 322,03	42 966,10	36 355,93	
	Кол.			1 200,000	650,000	550,000	
Белочка (конфеты)	БУ			173 694,91	102 508,47	71 186,44	
	Кол.			1 830,000	1 380,000	450,000	
Конфеты "Вишня в коньяке" (Польша)	БУ			3 750 000,00	3 750 000,00		
	Кол.			15 000,000	15 000,000		
Масло в коробке	БУ			25 423,73	25 423,73		
	Кол.			150,000	150,000		
Молоко в бидоне	БУ			11 363,64	11 363,64		
	Кол.			500,000	500,000		
Мука	БУ			2 045,45	2 045,45		
	Кол.			150,000	150,000		
Паста шоколадная	БУ			970 504,62	218 678,90	751 825,72	
	Кол.			16 000,000	3 400,000	12 600,000	
Сахарный песок	БУ			10 909,09	10 909,09		
	Кол.			400,000	400,000		
<b>Итого</b>	<b>БУ</b>			<b>5 081 102,46</b>	<b>4 212 836,07</b>	<b>868 266,39</b>	
	<b>Кол.</b>			<b>35 880,000</b>	<b>22 180,000</b>	<b>13 700,000</b>	

Рис. 6. Пример отчета о движении товара по складу за период, сформированного по данным таблицы-регистра.

#### Этапы работы

- Спроектировать структуру базы данных в части функциональности «Поступление товаров»
- Реализовать базу данных и интерфейс для таблиц-справочников
- Реализовать интерфейс для таблиц-документов
- Реализовать процедуру проведения документов и форму отчёта "остатки товаров"
- Спроектировать структуру базы данных в части учета продаж
- Реализовать интерфейс для таблицы-документа "Продажи"
- Реализовать процедуру проведения документа "Продажи" и форму отчета "Продажи"



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине «Параллельная обработка данных»

**Направление подготовки – 09.04.04 Программная инженерия**

Магистерская программа «Разработка программно-информационных систем»

**Форма подготовки (очная)**

Владивосток  
2018

**Паспорт**  
**фонда оценочных средств**  
**по дисциплине «Параллельная обработка данных»**

Код и формулировка компетенции	Этапы формирования компетенции	
	ПК-3 знанием методов оптимизации и умением применять их при решении задач профессиональной деятельности	Знает
Умеет		Планировать работу по проектированию распределённых информационных систем
Владеет		Инструментами проектирования распределённых информационных систем
ПК-9 способностью проектировать системы с параллельной обработкой данных и высокопроизводительные системы, и их компоненты	Знает	Методы создания распределённых информационных систем
	Умеет	Планировать работу по программной реализации распределённых информационных систем
	Владеет	Инструментами программной реализации распределённых информационных систем

№ п/п	Контролируемые разделы/темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
1	Lesson 1. Introduce Rich Client Applications (Тема 1. Сущность клиентских приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (1)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (1)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (1)
2	Lesson 2. JavaFX. (Тема 2. Платформа JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (1)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (1)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (1)
3	Lesson 3. Generics and JavaFX Collections (Тема 3. Обобщённые типы и коллекции платформы JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (2)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (2)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (2)
4	Lesson 4. UI Controls,	ПК 3	Знания	Опрос	Экзамен (ПР-1)

	Layouts, Charts, and CSS. (Тема 4. Средства построения пользовательского интерфейса платформы JavaFX: управляющие элементы, панели, графики и их взаимное расположение)	ПК 9		(УО-1)	(3)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (3)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (3)
5	Lesson 5. Visual Effects, Animation, WebView, and Media. (Тема 5. Визуальные эффекты, анимация, просмотр веб-элементов, отражение медиа-файлов)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (4)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (4)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (4)
6	Lesson 6. JavaFX Tables and Client GUI. (Тема 6. Таблицы платформы JavaFX и их реализация в графическом интерфейсе клиентских приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (5)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (5)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (5)
7	Lesson 7. JavaFX Concurrency and Binding. (Тема 7. Параллельные вычисления и связывание в платформе JavaFX)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (6, 7)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (6, 7)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (6, 7)
8	Lesson 8. Java Persistence API (JPA). (Тема 8. Программный интерфейс Java Persistence)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (8)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (8)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (8)
9	Lesson 9. Applying the JPA. (Тема 9. Использование Java Persistence API)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (9)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (9)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (9)
10	Lesson 10. Implementing a Multi-tier Design with RESTful Web Services. (Тема 10. Реализация многослойных приложений с помощью инструментария RESTful Web Services)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (10)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (10)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (10)
11	Lesson 11. Connecting to a RESTful Web Service. (Тема 11. Интеграция приложений на основе	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (11, 12)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (11, 12)

	инструментария RESTful Web Service)		Владения	С/работа (ПР-9)	Экзамен (ПР-1) (11, 12)
12	Lesson 12. Packaging and Deploying Applications. (Тема 12. Использование пакетов и развертывание приложений)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (13)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (13)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (13)
13	Lesson 13. Developing Secure Applications. (Тема 13. Развертывание приложений: вопросы безопасности)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (14, 15)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (14, 15)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (14, 15)
14	Lesson 14. Signing an Application and Authentication. (Тема 14. Подписывание приложений и их аутентификация)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (16, 17)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (16, 17)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (16, 17)
15	Lesson 15. Logging. (Тема 15. Логирование)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (18)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (18)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (18)
16	Lesson 16. Implementing Unit Testing and Using Version Control. (Тема 16. Модульное тестирование и контроль версий)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (19)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (19)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (19)
17	Lesson 17. Development Methodologies and Design Patterns. (Тема 17. Методы разработки и шаблоны проектирования)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (20)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (20)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (20)
18	Lesson 18. JavaFX History and Architecture. (Тема 18. История развития платформы JavaFX и её архитектура)	ПК 3 ПК 9	Знания	Опрос (УО-1)	Экзамен (ПР-1) (21, 22)
			Умения	Л/работа (ПР-6)	Экзамен (ПР-1) (21, 22)
			Владения	С/работа (ПР-9)	Экзамен (ПР-1) (21, 22)

## Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ПК-3 знанием методов оптимизации и умением применять их при решении задач профессиональной деятельности	знает (пороговый уровень)	Основные компоненты и протоколы распределённых информационных систем	Знание основных компонент и протоколов распределённых систем	Знание не менее 3 классов архитектур и 1 протокола обмена данными в распределённых ИС
	умеет (продвинутый)	Планировать работу по проектированию распределённых информационных систем	Умение выбирать классы архитектур распределённых ИС	Умение проектировать систему не менее чем в рамках 3 классов архитектур
	владеет (высокий)	Инструментами проектирования распределённых информационных систем	Владение средствами проектирования распределённых ИС	Владение не менее чем 2 средствами проектирования распределённых ИС
ПК-9 способностью проектировать системы с параллельной обработкой данных и высокопроизводительные системы, и их компоненты	знает (пороговый уровень)	Методы создания распределённых информационных систем	Знание методов создания распределённых ИС	Знание не менее чем 3 методов создания распределённых ИС
	умеет (продвинутый)	Планировать работу по программной реализации распределённых информационных систем	Умение реализовать архитектуру распределённых ИС	Умение реализовать систему не менее чем в рамках 3 классов архитектур
	владеет (высокий)	Инструментами программной реализации распределённых информационных систем	Владение средствами программной реализации распределённых ИС	Владение не менее чем 2 средствами программной реализации распределённых ИС

### Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

#### Промежуточный контроль

Промежуточный контроль осуществляется в конце семестра и завершает изучение дисциплины. Помогает оценить более крупные совокупности

знаний и умений, сформированность определенных профессиональных компетенций по дисциплине. Промежуточный контроль проводится в форме зачета, допуск к эзачету возможен для обучающихся, получивших оценку «зачтено» в результате выполнения самостоятельной работы и успешно выполнившие все лабораторные работы. зачет проводится в форме теста. Материал для теста прилагается в разделе «Вопросы к зачету».

### **Оценочные средства для промежуточной аттестации Вопросы к зачету по дисциплине**

Зачет проводится в тестовой форме. Тест составлен на базе материалов профессионального сертификационного экзамена Oracle Certified Associate, Java SE 7 Programmer и затрагивает следующие темы

- Типы данных в Java
- Операторы
- Управление выполнением программы
- Классы
- Пакеты и Интерфейсы
- Работа со строками
- Обработка исключений
- Потоки ввода/вывода

Примеры тестовых вопросов:

1. Given:

```
public class Sequence {
    Sequence() {
        System.out.print("x ");
    }

    public static void main(String[] args) {
        new Sequence().go();
    }
    void go() {
        System.out.print("y ");
    }
}
```

What is the result?

A) x y



- B) y x
- C) x c
- D) y c

2. Given:

```
public class MyStuff {
    MyStuff(String n) {
        name = n;
    }
    String name;

    public static void main(String[] args) {
        MyStuff m1 = new MyStuff("guitar");
        MyStuff m2 = new MyStuff("tv");
        System.out.println(m2.equals(m1));
    }

    public boolean equals(Object o) {
        MyStuff m = (MyStuff) o;
        if(m.name != null)
            return true;
        return false;
    }
}
```

What is the result?

- A) The output is "true" and MyStuff fulfills the Object.equals() contract.
- B) The output is "false" and MyStuff fulfills the Object.equals() contract.
- C) The output is "true" and MyStuff does NOT fulfill the Object.equals() contract.
- D) The output is "false" and MyStuff does NOT fulfill the Object.equals() contract
- E) Compilation fails

3. Given:

```
public class MyLoop {
    public static void main(String[] args) {
        String[] sa = {"tom ", "jerry "};
        for(int x = 0; x < 3; x++) {
            for(String s: sa) {
                System.out.print(x + " " + s);
            }
        }
    }
}
```

```

        if( x == 1) break;
    }
}
}
}

```

What is the result?

- A) 0 tom 0 jerry 1 tom
- B) 0 tom 0 jerry 1 tom 1 jerry
- C) 0 tom 0 jerry 2 tom 2 jerry
- D) 0 tom 0 jerry 1 tom 2 tom 2 jerry
- E) 0 tom 0 jerry 1 tom 1 jerry 2 tom 2 jerry

4. Given:

```

class Feline {
    public String type = "f ";
    public Feline() {
        System.out.print("feline ");
    }
}

```

```

public class Cougar extends Feline {
    public Cougar() {
        System.out.print("cougar ");
    }
}

```

```

public static void main(String[] args) {
    new Cougar().go();
}

```

```

void go() {
    type = "c ";
    System.out.print(this.type + super.type);
}
}

```

What is the result?

- A) cougar c c
- B) cougar c f

- C) feline cougar c c
- D) feline cougar c f
- E) Compilation fails
- F) An exception is thrown at run time.

5. Given:

```
class Alpha {
    String getType() {
        return "alpha";
    }
}

class Beta extends Alpha {
    String getType() {
        return "beta";
    }
}

class Gamma extends Beta {
    String getType() {
        return "gamma";
    }
}

public static void main(String[] args) {
    Gamma g1 = new Alpha();
    Gamma g2 = new Beta();
    System.out.println(g1.getType() + " "
        + g2.getType());
}
}
```

What is the result?

- A) alpha beta
- B) beta beta
- C) gamma gamma
- D) alpha alpha
- E) Compilation fails.

6. Choose the False statement among the following?

Java support globally accessible variables  
Java is programming language from Sun  
JavaScript is a programming language from Netscape  
None of the Above

7. The 16 bit coding scheme employed in Java programming is  
EDCDIC  
UNICODE  
ASCII  
Hexadecimal

8. What is the output of the following program?

```
public class example {  
    public static void main(String args[]) {  
        int x=0, y=2;  
        do {  
            x=++x;  
            y--;  
        } while(y>0);  
        System.out.println(x);  
    }  
}
```

0

1

2

Compilation Error

9. Command to execute a compiled java programs is

run

execute

javac

java

10. Choose the correct statement from the given choice?

Arrays in Java are essentially objects

It is not possible to assign one array to another. Individual elements of array can however be assigned.

Array elements are indexed from 1 to size of array

If a method tries to access an array element beyond its range, a compile warning is generated.

11. Consider the following code segment.

```
List<String> list = new ArrayList<String>();  
list.add("P");  
list.add("Q");  
list.add("R");  
list.set(2, "s");  
list.add(2, "T");  
list.add("u");  
System.out.println(list);
```

What is printed as a result of executing the code segment?

- (a) [P, Q, R, s, T]
- (b) [P, Q, s, T, u]
- (c) [P, Q, T, s, u]
- (d) [P, T, Q, s, u]
- (e) [P, T, s, R, u]

11. Consider the following instance variable and method.

```
private List<Integer> nums;  
/** Precondition: nums.size > 0  
*/  
public void numQuest() {  
    int k = 0;  
    Integer zero = new Integer(0);  
    while (k < nums.size()) {  
        if (nums.get(k).equals(zero))  
            nums.remove(k);  
        k++;  
    }  
}
```

Assume that List nums initially contains the following Integer values.

[0, 0, 4, 2, 5, 0, 3, 0]

What will List nums contain as a result of executing numQuest ?

(a) [0, 0, 4, 2, 5, 0, 3, 0]

(b) [4, 2, 5, 3]

(c) [0, 0, 0, 0, 4, 2, 5, 3]

(d) [3, 5, 2, 4, 0, 0, 0, 0]

(e) [0, 4, 2, 5, 3]

12. At a certain high school students receive letter grades based on the following scale.

Numeric Score Letter Grade

93 or above A

From 84 to 92 inclusive B

From 75 to 83 inclusive C

Below 75 F

Which of the following code segments will assign the correct string to grade for a given integer score ?

I.

```
if (score => 93)
```

```
grade 5 "A";
```

```
if (score >= 84 && score <= 92)
```

```
grade 5 "B";
```

```
if (score >= 75 && score <= 83)
```

```
grade 5 "C";
```

```
if (score , 75)
```

```
grade < "F";
```

II.

```
if (score >= 93)
```

```
grade 5 "A";
```

```
if (84 <= score <= 92)
```

```
grade 5 "B";
```

```
if (75 <= score <= 83)
```

```
grade 5 "C";
```

```
if (score < 75)
```

grade 5 "F";

III.

```
if (score >= 93)
    grade 5 "A";
else if (score >= 84)
    grade 5 "B";
else if (score >= 75)
    grade 5 "C";
else
    grade 5 "F";
```

- (a) II only
- (b) III only
- (c) I and II only
- (d) I and III only
- (e) I, II, and III

13. Consider the following declarations.

```
public interface Comparable {
    int compareTo(Object other);
}
public class SomeClass implements Comparable {
    // ... other methods not shown
}
```

Which of the following method signatures of compareTo will satisfy the Comparable interface requirement?

- I. public int compareTo(Object other)
  - II. public int compareTo(SomeClass other)
  - III. public boolean compareTo(Object other)
- (a) I only
  - (b) II only
  - (c) III only
  - (d) I and II only
  - (e) I, II, and III

14. Consider the following instance variable and method.

```
private int[] myStuff;
```

```

/** Precondition: myStuff contains int values in no particular order.
 */
public int mystery(int num) {
    for (int k = myStuff.length - 1; k >= 0; k--) {
        if (myStuff[k] < num) {
            return k;
        }
    }
    return -1;
}

```

Which of the following best describes the contents of myStuff after the following statement has been executed?

```
int m = mystery(n);
```

- (a) All values in positions 0 through m are less than n.
- (b) All values in positions m+1 through myStuff.length-1 are less than n.
- (c) All values in positions m+1 through myStuff.length-1 are greater than or equal to n.
- (d) The smallest value is at position m.
- (e) The largest value that is smaller than n is at position m.

15. Consider the following method.

```

/** Precondition: x >= 0
 */
public void mystery(int x) {
    System.out.print(x % 10);
    if ((x / 10) != 0) {
        mystery(x / 10);
    }
    System.out.print(x % 10);
}

```

Which of the following is printed as a result of the call mystery(1234) ?

- (a) 1441
- (b) 3443
- (c) 12344321
- (d) 43211234
- (e) Many digits are printed due to infinite recursion.

16. Consider the following two classes.

```
public class Dog {
```



```
public void act() {
    System.out.print("run");
    eat();
}
public void eat() {
    System.out.print("eat");
}
}
```

```
public class UnderDog extends Dog {
    public void act() {
        super.act();
        System.out.print("sleep");
    }
    public void eat() {
        super.eat();
        System.out.print("bark");
    }
}
```

Assume that the following declaration appears in a client program.

```
Dog fido = new UnderDog();
```

What is printed as a result of the call `fido.act()` ?

- (a) run eat
- (b) run eat sleep
- (c) run eat sleep bark
- (d) run eat bark sleep
- (e) Nothing is printed due to infinite recursion.

17. Consider the following recursive method.

```
public static int mystery(int n) {
    if (n == 0)
        return 1;
    else
        return 3 * mystery(n - 1);
}
```

What value is returned as a result of the call `mystery(5)` ?

- (a) 0
- (b) 3

- (c) 81
- (d) 243
- (e) 6561

18. Consider the following instance variable and method.

```
private int[] arr;
/** Precondition: arr.length > 0
 */
public int checkArray() {
    int loc = arr.length / 2;
    for (int k = 0; k < arr.length; k++) {
        if (arr[k] > arr[loc])
            loc = k;
    }
    return loc;
}
```

Which of the following is the best postcondition for checkArray ?

- (a) Returns the index of the first element in array arr whose value is greater than arr[loc]
- (b) Returns the index of the last element in array arr whose value is greater than arr[loc]
- (c) Returns the largest value in array arr
- (d) Returns the index of the largest value in array arr
- (e) Returns the index of the largest value in the second half of array arr

19. Assume the following declarations have been made.

```
private String s;
private int n;
public void changer(String x, int y)
{
    x = x + "peace";
    y = y * 2;
}
```

Assume s has the value "world" and n is 6. What are the values of s and n after the call changer(s, n) ?

- (a) s is world, n is 6
- (b) s is worldpeace, n is 6
- (c) s is world, n is 12

(d) a is worldpeace, n is 12

(e) a is peace n is 12

20. Consider the following methods.

```
public List<Integer> process1(int n)
{
    List<Integer> someList = new ArrayList<Integer>();
    for (int k = 0; k < n; k++)
        someList.add(new Integer(k));
    return someList;
}

public List<Integer> process2(int n)
{
    List<Integer> someList = new ArrayList<Integer>();
    for (int k = 0; k < n; k++)
        someList.add(k, new Integer(k));
    return someList;
}
```

Which of the following best describes the behavior of process1 and process2 ?

- (a) Both methods produce the same result and take the same amount of time.
- (b) Both methods produce the same result, and process1 is faster than process2.
- (c) The two methods produce different results and take the same amount of time.
- (d) The two methods produce different results, and process1 is faster than process2.
- (e) The two methods produce different results, and process2 is faster than process1.

21. Consider the following instance variable and incomplete method, partialSum, which is intended to return an integer array sum such that for all i, sum[i] is equal to arr[0] + arr[1] + ... + arr[i]. For instance, if arr contains the values { 1, 4, 1, 3 }, the array sum will contain the values { 1, 5, 6, 9 }.

```
private int[] arr;
public int[] partialSum() {
    int[] sum = new int[arr.length];
    for (int j = 0; j < sum.length; j++)
        sum[j] = 0;
    /* missing code */
}
```

```
    return sum;
}
```

The following two implementations of `/* missing code */` are proposed so that `partialSum` will work as intended.

#### Implementation 1

```
for (int j = 0; j < arr.length; j++)
    sum[j] = sum[j - 1] + arr[j];
```

#### Implementation 2

```
for (int j = 0; j < arr.length; j++)
    for (int k = 0; k <= j; k++)
        sum[j] = sum[j] + arr[k];
```

Which of the following statements is true?

- (a) Both implementations work as intended, but implementation 1 is faster than implementation 2.
- (b) Both implementations work as intended, but implementation 2 is faster than implementation 1.
- (c) Both implementations work as intended and are equally fast.
- (d) Implementation 1 does not work as intended, because it will cause an `ArrayIndexOutOfBoundsException`.
- (e) Implementation 2 does not work as intended, because it will cause an `ArrayIndexOutOfBoundsException`.

22. Consider the following declaration for a class that will be used to represent points in the xy-coordinate plane.

```
public class Point {
    private int myX; // coordinates
    private int myY;
    public Point() {
        myX = 0;
        myY = 0;
    }
    public Point(int a, int b) {
        myX = a;
```

```
    myY = b;
}
// ... other methods not shown
}
```

The following incomplete class declaration is intended to extend the above class so that points can be named.

```
public class NamedPoint extends Point {
    private String myName;
    // constructors go here
    // ... other methods not shown
}
```

Consider the following proposed constructors for this class.

I.

```
public NamedPoint() {
    myName = "";
}
```

II.

```
public NamedPoint(int d1, int d2, String name) {
    myX = d1;
    myY = d2;
    myName = name;
}
```

III.

```
public NamedPoint(int d1, int d2, String name) {
    super(d1, d2);
    myName = name;
}
```

Which of these constructors would be legal for the NamedPoint class?

- (a) I only
- (b) II only
- (c) III only
- (d) I and III
- (e) II and III

## Критерии выставления оценки магистранту на зачете/экзамене

Баллы (рейтингово й оценки)	Оценка зачета/ экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

### Критерии оценки проектов

- 100-86 баллов выставляется, если магистрант/группа точно определили содержание и составляющие части задания, умеют аргументировано отвечать на вопросы, связанные с заданием. Продемонстрировано знание и владение навыками самостоятельной исследовательской работы по теме. Фактических ошибок, связанных с пониманием проблемы, нет.

- 85-76 - баллов - работа магистранта/группы характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания

проблемы. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет.

- 75-61 балл – проведен достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимание базовых основ и теоретического обоснования выбранной темы. Привлечены основные источники по рассматриваемой теме. Допущено не более 2 ошибок в смысле или содержании проблемы

- 60-50 баллов - если работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Допущено три или более трех ошибок смыслового содержания раскрываемой проблемы

### Шкала оценивания проектов

Менее 60 баллов	Не зачтено
От 61 до 75 баллов	зачтено
От 76 до 85 баллов	зачтено
От 86 до 100 баллов	зачтено

### Текущий контроль

Текущий контроль предполагает систематическую проверку усвоения учебного материала, сформированности компетенций или их элементов, регулярно осуществляемую на протяжении изучения дисциплины, в соответствии с ее рабочей программой.

Состоит в проверке правильности выполнения заданий по самостоятельной работе. Задание зачтено, если нет ошибок. По текущим ошибкам даются пояснения.

Тесты предназначены для проверки знаний по компетенциям. Проверка достижения умений и навыков по компетенциям проверяется выполнением практических работ и курсовой работы.

### Примерные тесты для проверки сформированности компетенций

<b>ПК-3</b> знанием методов оптимизации и умением применять их при решении задач профессиональной деятельности	Основные компоненты и протоколы распределённых информационных систем
1. Which of the following is an interface that receives notifications of changes to an	ответы a. ObservableMap? b. ObservableMap c. javafx.collections d. <b>MapChangeListener</b>
2. To create an entity from a POJO, you must:	ответы

	<ul style="list-style-type: none"> <li>a. Implement the javax.persistence.Entity interface</li> <li>b. Create a persistence.xml file</li> <li>c. <b>Annotate the class with @Entity</b></li> <li>d. Declare the class as final</li> </ul>
--	---

<b>ПК-9 способностью проектировать системы с параллельной обработкой данных и высокопроизводительные системы, и их компоненты</b>	Методы создания распределённых информационных систем
1. Возможность распределённой и параллельной обработки данных реализована в продукте	<p>ответы</p> <ul style="list-style-type: none"> <li>a. MS Excel</li> <li>б. MS Power Point</li> <li><b>в. Matlab</b></li> </ul>
2. Ситуация взаимной блокировки потоков/процессов называется	<p>ответы</p> <ul style="list-style-type: none"> <li>a. баг</li> <li>б. дедлок</li> <li>в. фикс</li> </ul>