

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ «ОБЪЕКТНАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА»

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве методических указаний для обучающихся по основной образовательной программе высшего образования по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Составитель О.А. Гордеева

САМАРА
Издательство Самарского университета
2023

УДК 004.652.5(075)

ББК 32.973я7

Г681

Составитель *Гордеева Ольга Александровна*

Рецензент канд. техн. наук, доцент И. В. Лёзина

Г681 **Методические указания к курсовой работе по дисциплине «Объектная распределенная обработка»:** методические указания / *О.А. Гордеева.* – Самара: Издательство Самарского университета, 2023. – 20 с.

Изложены требования к выполнению и оформлению курсовой работы по дисциплине «Объектная распределенная обработка», которая является завершающим этапом изучения технологий разработки объектно-ориентированных распределенных приложений на языке программирования Java. Указания также содержат требования к оформлению пояснительной записки и предназначены для обучающихся по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Подготовлено на кафедре программных систем.

УДК 004.652.5(075)

ББК 32.973я7

СОДЕРЖАНИЕ

Введение.....	4
Основные этапы выполнения курсовой работы	5
Основные контрольные точки выполнения курсовой работы.....	6
Номинальная архитектура программной системы	7
Варианты заданий на курсовую работу.....	9
Общие замечания и рекомендации к работе	16
Содержание пояснительной записки к курсовой работе	17
Библиографический список.....	18
<i>Приложение 1</i>	19

ВВЕДЕНИЕ

Целью выполнения курсовой работы является закрепление студентом навыков создания объектных распределенных приложений на языке java [1], как web-ориентированных, так и не относящихся к web-приложениям, обоснование и применение изученных в теоретическом и лабораторно-практическом цикле технологий для решения поставленной задачи распределенной обработки данных, определение и формализация требований и допущений к системе, накладываемых реализуемой технологией обработки, что в целом готовит студента к выполнению итоговой выпускной квалификационной работы.

В ходе выполнения курсовой работы студент разрабатывает программную систему распределенной обработки данных, реализующую предложенное задание с помощью одной из предложенных технологий распределенной обработки. Теоретические сведения, необходимые для выполнения задания, содержатся в теоретическом лекционном курсе. Кроме того, основные этапы и особенности реализации той или иной технологии распределенной обработки данных рассматриваются и реализуются на практических занятиях, а также на лабораторных работах.

Основой для создания программной системы курсовой работы может служить программная система, разработанная в ходе выполнения одной из работ лабораторного практикума.

Задание на курсовую работу выдает преподаватель, любые изменения в задании уточняются и согласовываются с преподавателем. Отчетная документация разрабатывается в соответствии с существующими в организации стандартами.

В ходе выполнения курсовой работы преподавателем назначаются контрольные точки по основным этапам в соответствии с графиком выполнения работ. По завершении создания программной системы студент проводит ее отладку и тестирование, демонстрирует резуль-

таты работы системы преподавателю, при необходимости осуществляет ее доработку, оформляет пояснительную записку по курсовой работе. Итоговая оценка ставится по итогам защиты курсовой работы с обязательным учетом промежуточных оценок за этапы выполнения.

ОСНОВНЫЕ ЭТАПЫ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1. Анализ предметной области, где формализуются описания участвующих в процессе обработки объектов, определяется их структура, обозначаются ограничения и допущения (2 недели).

2. Постановка задачи, где формулируются функции, выполняемые системой, а также требования к системе (1 неделя).

3. Рассмотрение и определение используемых технологий и методик, применяемых для решения поставленной задачи, выявление их особенностей и возможностей для реализации программной системы с учетом требований к ней и особенностей предметной области (2 недели).

4. Построение структурной схемы, где производится подробная декомпозиция системы на подсистемы, прописываются связи между ними, формализуются потоки данных для реализации используемой технологии обработки данных, а также построение проекта системы на языке UML [2] (как минимум диаграмма вариантов использования, диаграмма последовательностей для основного варианта использования, диаграмма деятельности всей системы и диаграмма компонентов, отображающая файлы с исходными кодами и необходимыми дополнительными файлами) (2 недели).

5. Реализация системы, где проверяется полнота и качество реализации сформулированных требований к системе, формируется список исключительных ситуаций, подготавливается пакет тестовых примеров. На данном этапе программная система демонстрируется преподавателю. При необходимости, производится доработка и корректировка элементов системы, формализованных в предыдущих эта-

пах, с повторной демонстрацией результатов работы преподавателю (4 недели).

6. Оформление документации по курсовой работе (2 неделя).
7. Защита курсовой работы (1 неделя).

ОСНОВНЫЕ КОНТРОЛЬНЫЕ ТОЧКИ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

1. Утверждение задания – условия игры, входные и выходные параметры, ограничения, особенности, технология распределенной обработки. (2-3 неделя семестра).

2. Утверждение структуры системы (4-5 неделя семестра).

3. Утверждение проекта системы и прототипа пользовательского интерфейса (7-8 неделя семестра).

4. Демонстрация работы системы (8-11 неделя семестра).

5. При наличии замечаний корректировка системы и повторная демонстрация ее работы (9-12 неделя семестра).

6. Представление пояснительной записки по курсовой работе (12 неделя семестра).

7. Внесение корректировок в пояснительную записку (13 неделя семестра).

8. Защита курсовой работы (14 неделя семестра).

Примерный график выполнения курсовой работы:

1-я учебная неделя – определиться с вариантом задания, понять задание, согласовать задание с преподавателем.

3-я учебная неделя – формализовать предметную область, конкретизировать правила игры, определить допущения и ограничения.

5-я учебная неделя – изучить выбранную технологию реализации обработки данных, определить особенности, требования и ограничения, накладываемые выбранной технологией на формализацию предметной области и проект разрабатываемого программного приложения.

7-я учебная неделя – разработать проект системы на языке UML, проект содержит необходимое количество диаграмм в соответствии с настоящими методическими указаниями.

11-я учебная неделя – разработка программного обеспечения, его тестирование и отладка.

13-я учебная неделя – оформление пояснительной записки к курсовой работе.

14-я неделя – защита курсовой работы.

НОМИНАЛЬНАЯ АРХИТЕКТУРА ПРОГРАММНОЙ СИСТЕМЫ

1. Система представляет собой клиент-серверное приложение. Серверное приложение определяет правила игры, данные игры, порядок и форму обработки запросов от клиентской части, порядок и форму отклика в клиентскую часть приложения. В парных играх серверное приложение также имитирует игрока-оппонента.

2. Клиентское приложение формирует запрос в виде, требуемом серверным приложением, а также интерпретирует полученный ответ с учетом эргономичности и комфорта для пользователя.

3. Серверная и клиентская части приложения, а также структура и количество необходимых вспомогательных серверных и клиентских объектов определяются обучающимся исходя из правил игры и реализуемой технологии.

КРАТКОЕ ОПИСАНИЕ РЕАЛИЗАЦИЙ ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННОЙ ОБЪЕКТНОЙ ОБРАБОТКИ

Сокеты

Сокет (англ. socket) – программный интерфейс, предназначенный для обмена данными между двумя приложениями или процессами, функционирующими на одном или разных узлах компьютерной сети

[3]. Сокет является конечной точкой двустороннего соединения и привязан к номеру порта.

Каждая из сторон клиент-серверного взаимодействия имеет свой сокет. Различают два вида сокетов. Клиентский сокет предназначен для установки соединения с сервером и отправки запросов к серверному приложению. Серверный сокет прослушивает обращения клиентского приложения и устанавливает с ним связь.

Для реализации программного приложения в курсовой работе используются сокеты, поддерживающие обмен данными в соответствии с сетевым протоколом TCP.

Remote Method Invocation

RMI – это программная модель, в соответствии с которой java-приложение может вызывать методы объекта другого java-приложения, функционирующего на другой виртуальной машине, на другом узле компьютерной сети [4]. Модель содержит набор объектов (классов) для организации удаленного (remote) взаимодействия.

RMI-приложение, как правило, содержит две основные части – сервер и клиент. Серверная часть приложения создает удаленные объекты, публикует ссылки на них. Клиентская часть получает ссылку на удаленные объекты и вызывает их методы, обращаясь к ним как к «родным», находящимся в том же адресном пространстве на той же виртуальной машине java.

Сервлеты&JSP

Сервлет (servlet) – это интерфейс на языке java, расширяющий возможности серверного приложения, поддерживающий обращения в рамках модели «запрос-отклик» [5]. Гипотетически сервлеты могут поддерживать любой формат запросов, однако актуальными являются HTTP-сервлеты, обрабатывающие методы GET и POST протокола HTTP, то есть сервлеты предназначены для реализации компонентов серверного web-приложения.

Сервлеты используются в распределенных приложениях стандарта JEE [6], реализация которого требует обязательного использования контейнера – сервера приложений (Application Server).

JSP (Java Server Pages) – это технология, предназначенная для упрощения создания web-страниц (страниц, содержащих тэги разметочного языка) с динамически изменяющимся во время генерации страницы содержимым.

Динамическая часть формируется на языке java на стороне сервера. JSP-файл транслируется в класс сервлета, который и реализует программный код. Скриплеты, генерирующие динамическое содержимое страницы, преобразуются в части методов сервлета. Таким образом, JSP является надстройкой над Servlets.

Подробно технологии распределенной обработки рассматриваются в теоретическом лекционном курсе. Там же приведены примеры исходных программных кодов для каждой из технологий, правила его структурирования, перечислено программное обеспечение, необходимое для реализации технологии.

ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ

Задание 1

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Змейка» со следующими правилами:

- в игре присутствует один игрок-пользователь;
- пользователь управляет движущимся объектом (змейкой);
- змейка выполняет задания серверного приложения (например, пересечь заданную точку на экране, выполнить заданную траекторию, и т.д.);
- при успешном выполнении задания игрок получает некоторый гешефт (например, змейка увеличивается, игрок получает баллы, и т.д.)

– игра прекращается при нарушении некоторых ограничений игры (например, выход за границу игрового поля, пересечение змейкой самой себя, и т.д.).

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 2

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Реверси» со следующими правилами:

- в игре присутствует игрок-пользователь и игрок-компьютер.
- игра имеет поле размером 8x8 и 64 двусторонних фишки, одна сторона которых черная, другая – белая;
- в начале игры в центре доски выставляются 4 фишки, две белых и две черных, по диагонали;
- пользователь играет черной стороной и начинает игру, компьютер играет белой стороной, ходы делаются по очереди;
- игрок ставит фишку своей стороной так, чтобы между ней и уже имеющейся фишкой того же цвета находился непрерывный ряд фишек соперника, при этом запертые фишки соперника переворачиваются;
- если хода нет, то игрок пропускает свой ход;
- игра продолжается до окончания фишек или ходов;
- выигравшим объявляется тот игрок, фишек чьего цвета на поле больше.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 3

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Поле чудес» со следующими правилами:

- игрок должен угадать загаданное компьютером слово из некоторой области/темы, причем игроку известно только количество букв в слове;
- предлагая некоторые буквы, пользователь угадывает слово;
- ход игрока осуществляется при выполнении некоторого условия (например, игрок «вращает барабан», игроку дается ограниченное число попыток, как в игре «Виселица», и т.д.);
- игра заканчивается, когда игрок угадывает слово или количество ходов заканчивается.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 4

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Тетрис» со следующими правилами:

- в игре присутствует один игрок, который заполняет игровое поле фигурами;
- в игре есть базовый набор фигур, размер поля также можно задавать;
- серверное приложение предлагает игроку последовательно фигуры из набора, игрок должен их расположить на поле снизу вверх;
- игрок может управлять фигурами до момента размещения их на поле (вращать, двигать);

– игра заканчивается, когда поле полностью заполнено фигурами снизу доверху, при этом баллы начисляются с учетом возможных пустых ячеек на поле.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 5

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Морской бой» со следующими правилами:

- в игре присутствует один игрок, который задает размеры поля и количество целей на поле;
- серверная часть размещает цели на поле случайным образом;
- игрок совершает «выстрелы» по ячейкам поля, получая информацию о пораженных целях и промахах;
- количество «выстрелов» игрока ограничено;
- игра заканчивается, когда игрок совершил все выстрелы или поразил все цели.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 6

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Пазл» со следующими правилами:

- в игре присутствует один игрок, который попарно открывает ячейки игрового поля, если ячейки имеют одинаковое изображе-

ние/цвет, то они освобождаются, если ячейки разные, то они закрываются обратно;

- размер поля можно задавать, количество ячеек на поле должно быть четным;

- игра заканчивается, когда поле полностью освобождено.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 7

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Крестики-нолики» со следующими правилами:

- в игре присутствуют один игрок-пользователь и один игрок-компьютер, пользователь начинает первым и играет крестиками, компьютер играет ноликами;

- размер игрового поля можно задавать, поле должно быть квадратным;

- игроки по очереди занимают ячейку поля своим знаком с целью выстроить длинную последовательность (горизонтальную, вертикальную или диагональную) и помешать сделать то же оппоненту;

- игра заканчивается победой того игрока, чья цепочка на поле будет длиннее, а поле будет полностью занято.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 8

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Филворд» со следующими правилами:

- в игре присутствует один игрок;
- игровое поле квадратное, заполнено буквами, цепочки которых составляют слова;
- цепочка букв каждого слова может иметь изломы, но не может себя пересекать;
- игрок должен распознать слово на поле и последовательно обойти цепочку его букв, при этом соответствующие ячейки считаются освобожденными;
- игра заканчивается, когда все слова распознаны, все цепочки маркированы, а поле полностью освобождено.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 9

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Быки и коровы» со следующими правилами:

- игрок должен угадать загаданную компьютером комбинацию из 4 неповторяющихся цифр;
- на каждом ходе игрок делает предположение о загаданной комбинации, получая от компьютера сведения о совпадении искомой комбинации с предположением в виде «N быков, M коров», где «N быков» – количество цифр в комбинации, совпавших и по значению, и по их позиции в загаданной комбинации, а «M коров» – количество цифр в комбинации, совпавших только по значению;

– игра заканчивается, когда игрок-пользователь угадает загаданную комбинацию, то есть ответом компьютера будет «4 быка».

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

Задание 10

Написать распределенное клиент-серверное приложение, используя указанную технологию распределенной объектной обработки. Приложение реализует игру «Мозаика» со следующими правилами:

– в игре присутствует один игрок-пользователь, который должен за определенное время выложить из разноцветных элементов предложенную компьютером картинку

– в игре возможен выбор уровня сложности картинки;

– игра заканчивается, когда картинка полностью собрана, либо отведенное время истекло.

Вариант 1. Сокетные соединения.

Вариант 2. Технология RMI.

Вариант 3. Технология servlets (JSP, JSF).

ОБЩИЕ ЗАМЕЧАНИЯ И РЕКОМЕНДАЦИИ К РАБОТЕ

1. Правила игры можно модифицировать, упрощать или усложнять.
2. При согласовании с преподавателем можно изменить задание (выбрать игру самостоятельно).
3. Пользовательский интерфейс разработанного программного приложения должен адекватно и полно отражать возможности игры для пользователя.
4. Серверное приложение должно реализовывать параллельную обработку запросов пользователя (многопользовательское приложение).
5. Можно реализовать игру для двух пользователей-игроков-оппонентов.
6. В приложении обязательны следующие функции:
 - идентификация пользователя;
 - ведение статистики игры (количество сделанных ходов, количество успехов/неуспехов, количество набранных очков и т.д.);
 - оповещение пользователя об окончании игры с предоставлением ему текущей статистики и сообщением о выигрыше/проигрыше;
 - ведение рейтинга лучших игроков (если в игре есть уровни, то для каждого уровня).
7. Упрощения задания ведут к снижению итоговой оценки, реализация дополнительных функций может повысить итоговую оценку.

СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

Записка по курсовой работе оформляется в соответствии со стандартом организации [7] и должна содержать:

- Титульный лист. Образец титульного листа приведен в Приложении 1.

- Задание.

- Реферат. Содержит структуру пояснительной записки (количество страниц, рисунков, таблиц, приложений), ключевые слова, краткую аннотацию работы.

- Оглавление с указанием страниц.

- Введение.

- Основная часть:

- описание и формализация предметной области;

- формулировка постановки задачи (правила игры, ограничения, допущения, формат входных и выходных данных, функции приложения);

- описание технологий, используемых в ходе выполнения курсовой работы, дополнительных библиотек, платформ и технологий;

- описание проекта разрабатываемой системы на языке UML;

- описание экранных форм с приведением тестовых примеров и реакций системы на исключительные ситуации;

- результаты работы системы.

- Список использованных источников.

- Исходные коды программы в качестве приложения.

Рекомендуемый объем пояснительной записки – 15-20 листов без учета приложений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Эккель, Б. Философия Java. 4-е полное издание / Б. Эккель. – Санкт-Петербург: Питер, 2022. – 1168 с.
2. Рамбо, Дж. Введение в UML от создателей языка / Дж. Рамбо, Г. Буч, И. Якобсон. – Москва: ДМК-Пресс, 2015. – 496 с.
3. All About Sockets // The Java Tutorials: [сайт]. – 2022. – URL: <https://docs.oracle.com/javase/tutorial/networking/sockets/index.html> (дата обращения 23.03.2023).
4. An Overview of RMI Applications // The Java Tutorials: [сайт]. – 2022. – URL: <https://docs.oracle.com/javase/tutorial/rmi/overview.html> (дата обращения 23.03.2023).
5. Курняван, Б. Создание WEB-приложений на языке Java с помощью сервлетов, JSP и EJB / Б. Курняван. – Москва: Лори, 2021. – 880 с.
6. Дейтел, Х.М. Технологии программирования на Java 2. Распределенные приложения / Х.М. Дейтел, П.Дж. Дейтел, С.И. Сантри. – Москва: Бином-Пресс, 2009. – 420 с.
7. СТО СГАУ 02068410-004-2018. Общие требования к учебным текстовым документам. Стандарт организации // Самарский университет: [сайт]. – 2018. – URL: https://ssau.ru/docs/sveden/localdocs/STO_SGAU_02068410-004-2018.pdf (дата обращения 23.03.2023).

Приложение 1

Образец титульного листа пояснительной записки

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский
университет имени академика С.П. Королева»
(Самарский университет)

Институт _____ информатики и кибернетики _____
Кафедра _____ программных систем _____

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ
по дисциплине «Объектная распределенная обработка»

по теме «*Название темы*»

Выполнил:
обучающийся группы № 6414-020302D _____ И.И. Иванов

Проверил:
руководитель работы,
должность, степень, звание _____ П.П. Петров

Дата защиты _____

Оценка _____

Самара 2023

Учебное издание

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ
«ОБЪЕКТНАЯ РАСПРЕДЕЛЕННАЯ ОБРАБОТКА»**

Методические указания

Составитель Гордеева Ольга Александровна

Редакционно-издательская обработка
издательства Самарского университета

Подписано в печать 15.11.2023. Формат 60×84 1/8.

Бумага офсетная. Печ. л. 1,25.

Тираж 27 экз. Заказ . Арт. – 3(Р2/МУ)/2023

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Издательство Самарского университета.

443086, Самара, Московское шоссе, 34.