**Урок 13: Работа с сетью**

**Повторение прошлого материала (10 минут):**

“На прошлом уроке мы познакомились с созданием графических интерфейсов в Java с использованием библиотеки Swing. Кто может рассказать, как создаются окна и добавляются компоненты, такие как кнопки и текстовые поля?”

**Цели и задачи урока:**

“Сегодня мы рассмотрим, как можно взаимодействовать с сетью с помощью Java.”

**Познание нового (25 минут):**

Основные понятия:

1. Что такое сетевое программирование:

“Сетевое программирование — это процесс написания программ, которые могут обмениваться данными по сети, например, через интернет.”

2. Основные классы для работы с сетью:

“Java предоставляет классы для работы с сетью в пакете java.net. Основные классы включают Socket, ServerSocket, URL и URLConnection.”

3.Пример создания простого клиента и сервера:

Пример кода для сервера:



Пример кода для клиента:



“В примере кода сервера мы создаем объект ServerSocket, который слушает подключения на порту 12345. Когда клиент подключается, сервер создает Socket для общения с клиентом. Мы используем BufferedReader для чтения сообщений от клиента и PrintWriter для отправки сообщений.”

“В примере кода клиента мы создаем объект Socket, который подключается к серверу на localhost и порту 12345. Мы отправляем сообщение серверу с помощью PrintWriter и читаем ответ с помощью BufferedReader.”

**Некомпьютерная активность (10 минут):**

1. Обсуждение:

“Какие примеры сетевых приложений вы знаете? Как вы думаете, какие задачи можно решить с помощью сетевого программирования?”

2. Групповая работа:

“Разделитесь на группы и обсудите, как можно использовать сетевое программирование в ваших проектах. Придумайте несколько идей для сетевых приложений.”

**Работа над проектом (25 минут):**

“Создайте проект, который включает в себя простой сервер и клиент, обменивающиеся сообщениями.”

Пример кода для сервера:



Пример кода для клиента:



“Мы создаем проект, в котором сервер слушает подключения на порту 12345 и обрабатывает сообщения от клиентов. Клиент подключается к серверу и отправляет сообщение, получая ответ от сервера.”

**Дополнительное задание:**

“Добавьте к проекту возможность отправки нескольких сообщений от клиента к серверу. Сервер должен отправлять ответ на каждое сообщение.”

Пример кода для сервера:



Пример кода для клиента:



“Мы добавили возможность отправки нескольких сообщений от клиента к серверу, используя цикл while. Клиент считывает пользовательский ввод и отправляет его серверу, который отвечает на каждое сообщение.”

**Проблемная задача (10 минут):**

“Напишите программу, которая создает сервер, принимающий сообщения от нескольких клиентов и отправляющий обратно количество символов в каждом сообщении.”

Пример кода для сервера:



“В этом примере мы создали сервер, который принимает подключения от нескольких клиентов с помощью ServerSocket. Для обработки каждого клиента создается новый поток, чтобы сервер мог одновременно обслуживать несколько клиентов. В классе ClientHandler мы обрабатываем сообщения от клиента и отправляем обратно количество символов в каждом сообщении.”

**Рефлексия (10 минут):**

“Что нового вы узнали сегодня о сетевом программировании?”

“Какие примеры использования сетевых приложений вы можете привести?”

“Какие трудности у вас возникли при создании сетевых приложений?”

Обсуждение выполненного задания:

“Как создание сетевых приложений может улучшить взаимодействие между пользователями? Какие методы и классы оказались наиболее полезными?”