**Урок 10: Работа с файлами**

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

“Что мы изучали на прошлом уроке? Кто может напомнить, что такое исключения и как их обрабатывать в Java? Какие ключевые слова используются для обработки исключений?”

**Цели и задачи урока:**

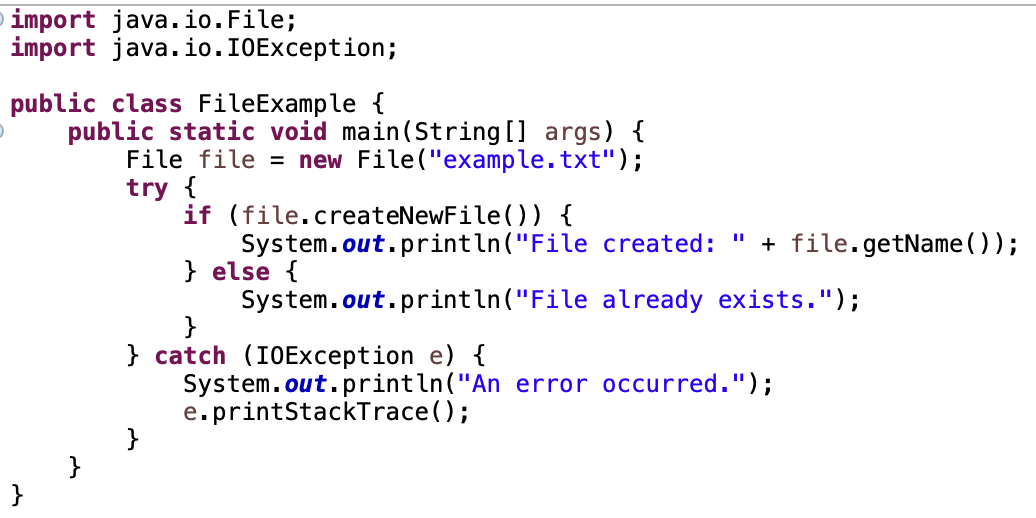
“Сегодня мы разберем как подключать и как работать со сторонними файлами в Java”

**Познание нового (25 минут):**

1.Файлы и потоки:

“Файл — это именованное место на диске, используемое для хранения информации. Поток — это абстракция, представляющая источник или приемник данных.”

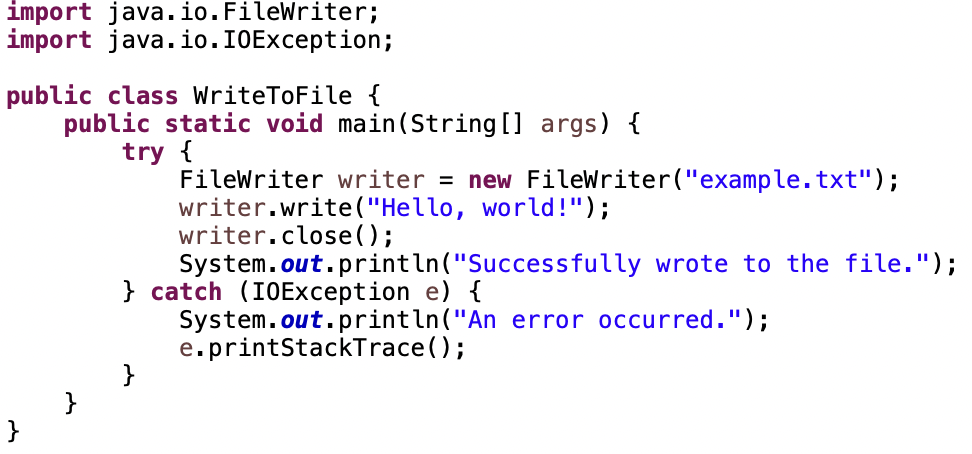
Пример:



2. Запись в файл:

“Для записи данных в файл можно использовать классы FileWriter и BufferedWriter.”

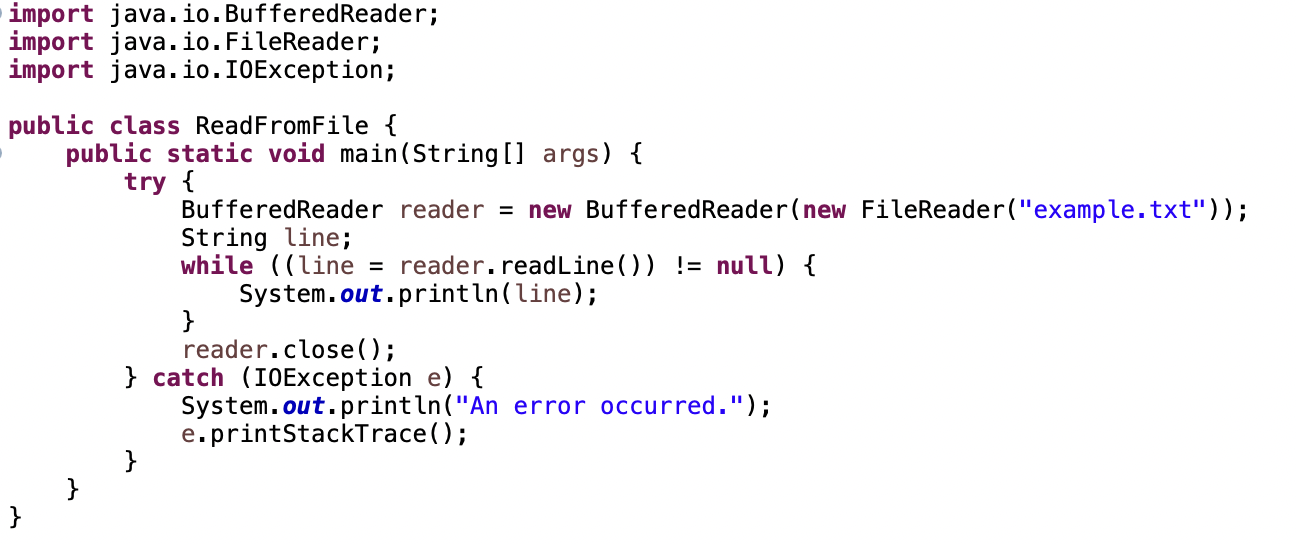
Пример:



3. Чтение из файла:

“Для чтения данных из файла можно использовать классы FileReader и BufferedReader.”

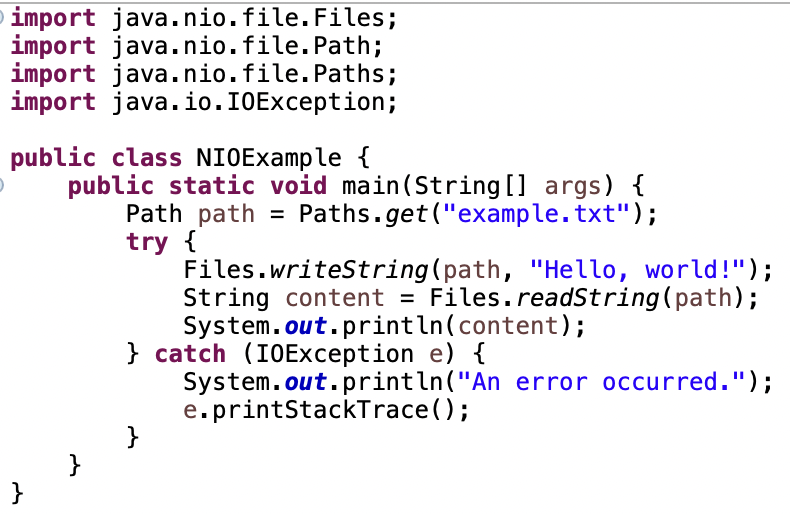
Пример:



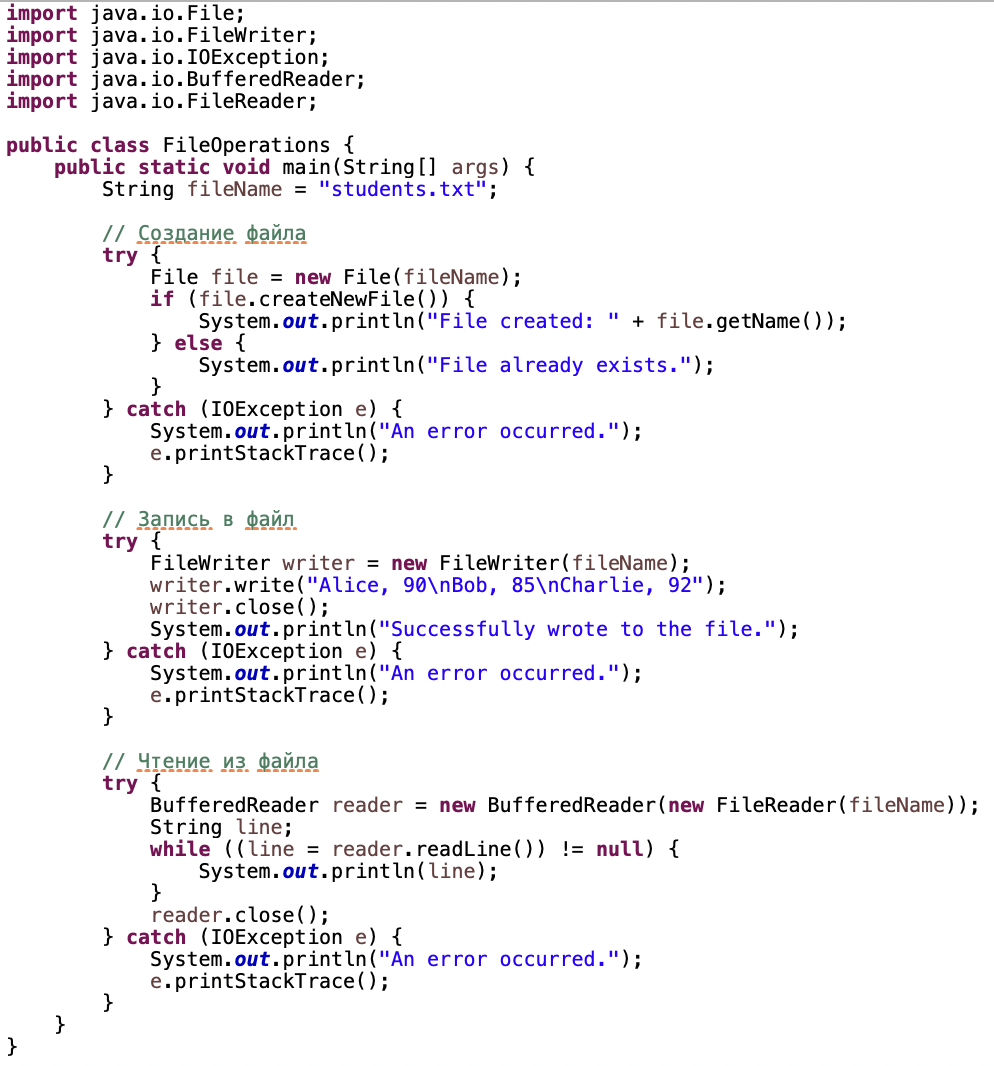
4. Файловый ввод/вывод с помощью NIO:

“Java NIO (New I/O) предоставляет более мощные возможности для работы с файлами и потоками. Основные классы — это Path, Paths и Files.”

Пример:



Пример программы:



“В этой программе мы создали файл, записали в него данные о студентах и прочитали их обратно, используя классы File, FileWriter и BufferedReader.”

**Некомпьютерная активность (10 минут):**

1. Обсуждение:

“Почему работа с файлами важна в программировании?”

“Какие типы файлов вам приходилось использовать в повседневной жизни?”

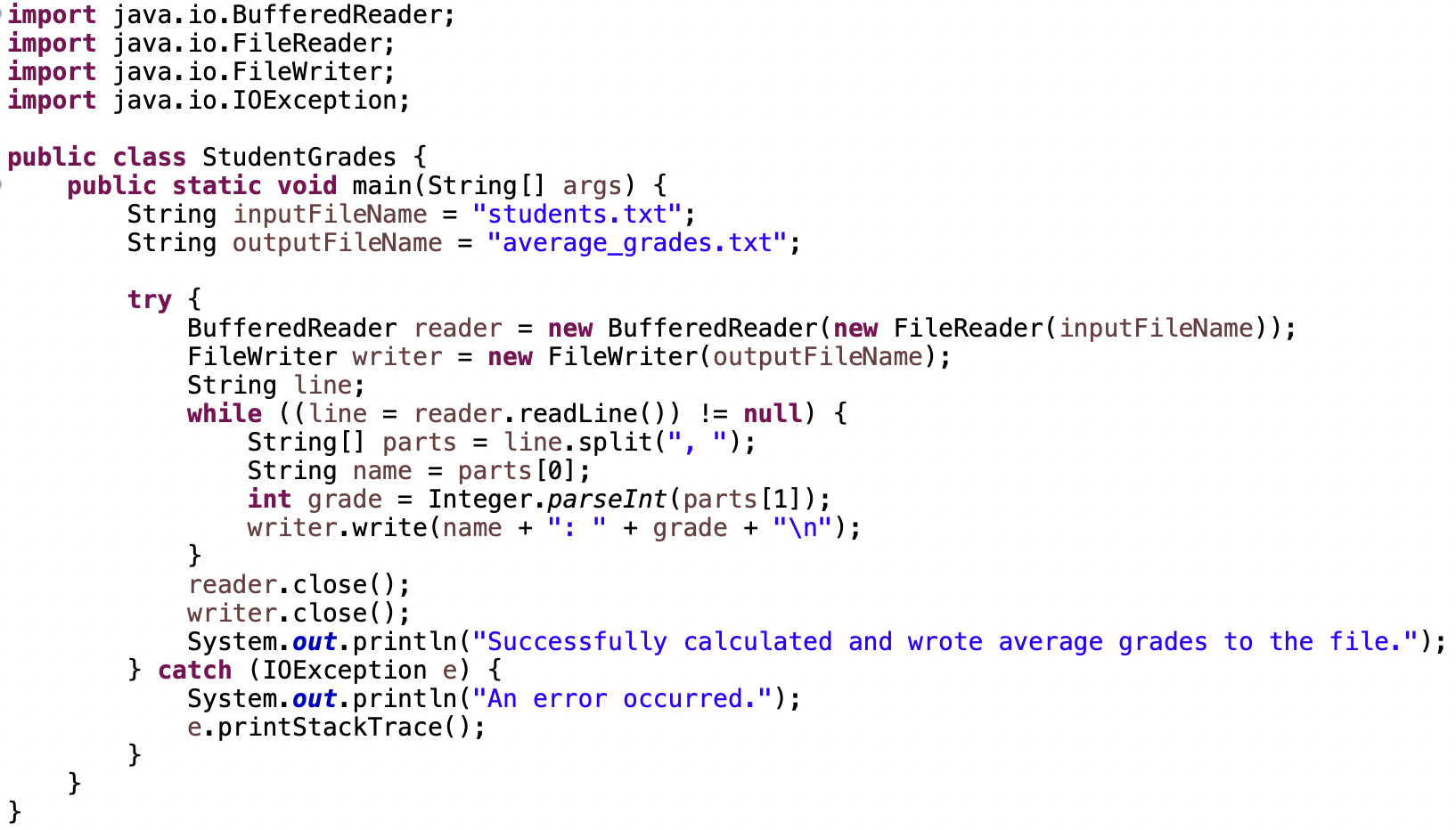
2. Групповая работа:

“Разделитесь на группы по 3-4 человека и обсудите, как можно использовать файлы для хранения данных в ваших проектах.”

**Работа над проектом (25 минут):**

“Создайте новый проект и напишите программу, которая включает методы для вычисления площади и периметра прямоугольника.”

Пример кода:



“Мы создали класс StudentGrades, который считывает данные о студентах из файла students.txt, обрабатывает их и записывает результаты в файл average\_grades.txt.”

**Дополнительное задание:**

“Добавьте метод для нахождения и записи максимальной и минимальной оценки.”

Пример кода:

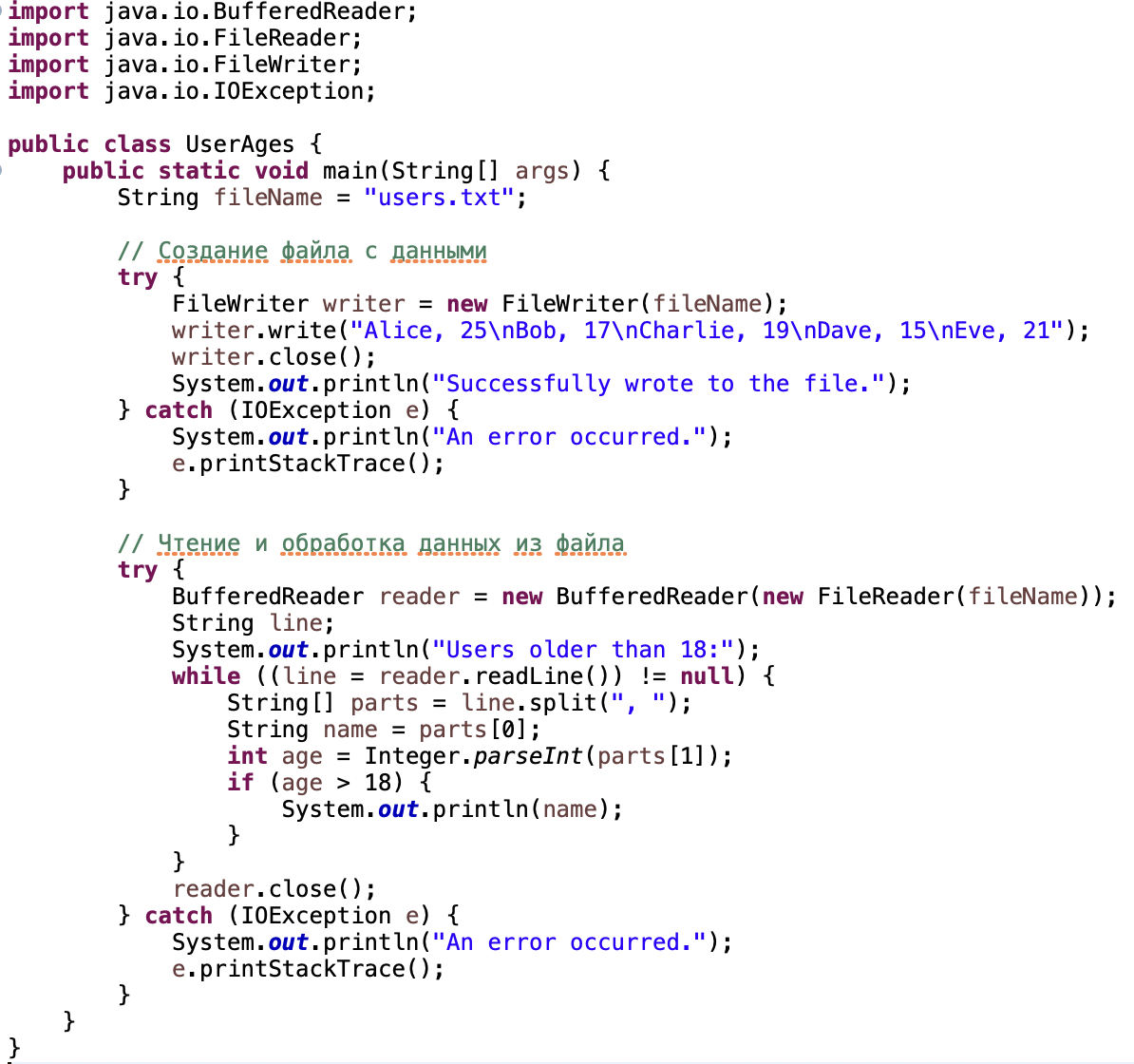


“Мы добавили код для нахождения максимальной и минимальной оценки и записи их в файл grades\_summary.txt.”

**Проблемная задача (10 минут):**

“Напишите программу, которая создает файл с именами и возрастами пользователей, затем считывает эти данные и выводит имена пользователей старше 18 лет.”

Пример кода:



“Мы создали класс UserAges, который записывает данные о пользователях в файл users.txt, а затем считывает эти данные и выводит имена пользователей старше 18 лет.”

Решение:

“Используйте классы FileWriter и BufferedReader для записи и чтения данных из файла соответственно. Примените метод split для разделения строки на части и преобразуйте возраст в целое число с помощью Integer.parseInt.”

**Рефлексия (10 минут):**

“Что нового вы узнали сегодня?”

“Как можно применять работу с файлами в реальных проектах?”

“Какие трудности у вас возникли и как вы их преодолели?”

Обсуждение выполненного задания:

“Кто хочет показать свой код и результат работы?”

Подведение итогов:

“Сегодня мы изучили, как работать с файлами в Java, научились создавать, записывать и считывать файлы, а также писать программы с их использованием. Молодцы, вы отлично справились!”