Java IO Lab

SANJANA KUMBHAR

1. Create an application that take 3 string input values from user one from command line ,second

from bufferedReader and third through scanner class and display all.

**package** io\_assignment;

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** java.util.Scanner;

**public** **class** Que1 {

**public** **static** **void** main(String[] args) **throws** Exception {

// Input 1: Command-line argument

String input1 = args.length > 0 ? args[0] : "No command-line argument provided";

// Input 2: BufferedReader

BufferedReader r = **new** BufferedReader(**new** InputStreamReader(System.***in***));

System.***out***.print("Enter a string (BufferedReader): ");

String input2 = r.readLine();

// Input 3: Scanner

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter a string (Scanner): ");

String input3 = scanner.nextLine();

// Display all inputs

System.***out***.println("\nInputs Received:");

System.***out***.println("Command-line argument: " + input1);

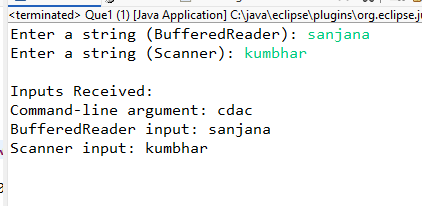
System.***out***.println("BufferedReader input: " + input2);

System.***out***.println("Scanner input: " + input3);

scanner.close();

}

}



2. Write a program to ask 7 int arrays from user through command line argument and display

them in ascending order .Create a user defined exception class that check where argument

entered by user is less than 7 and should be more than zero and handle all built in exception

raising In program .

**package** io\_assignment;

//Write a program to ask 7 int arrays from user through command line argument and display

//them in ascending order .Create a user defined exception class that check where argument

//entered by user is less than 7 and should be more than zero and handle all built in exception

//raising In program .

**import** java.util.Arrays;

**class** InvalidArgumentException **extends** Exception {

**public** InvalidArgumentException(String message) {

**super**(message);

}

}

**public** **class** Que2 {

**public** **static** **void** main(String[] args) {

**try** {

**if** (args.length != 7) {

**throw** **new** InvalidArgumentException("Exactly 7 arguments are required.");

}

// Parse arguments to integers

**int**[] numbers = **new** **int**[7];

**for** (**int** i = 0; i < args.length; i++) {

**int** value = Integer.*parseInt*(args[i]);

**if** (value <= 0) {

**throw** **new** InvalidArgumentException("All numbers must be greater than zero.");

}

numbers[i] = value;

}

Arrays.*sort*(numbers);

System.***out***.println("Sorted Array: " + Arrays.*toString*(numbers));

} **catch** (InvalidArgumentException e) {

System.***err***.println("Error: " + e.getMessage());

} **catch** (NumberFormatException e) {

System.***err***.println("Error: All arguments must be valid integers.");

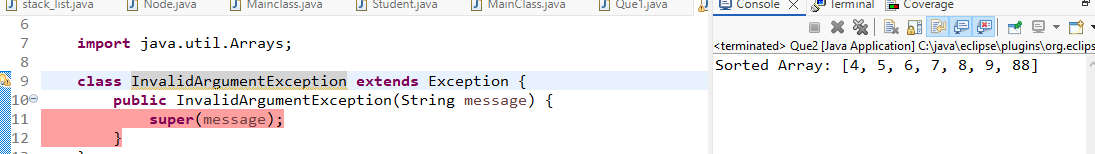
} **catch** (Exception e) {

System.***err***.println("Unexpected error: " + e.getMessage());

}

}

}



3. Write a program to ask a string from user and display it in uppercase ,exit loop when quit is not

encountered.(using BufferedReader)

**package** io\_assignment;

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**import** java.io.IOException;

**public** **class** Que3 {

**public** **static** **void** main(String[] args) {

**try** (BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(System.***in***))) {

String input;

**while** (**true**) {

System.***out***.print("Enter a string (or type 'quit' to exit): ");

input = reader.readLine(); // Read input from user

**if** ("quit".equalsIgnoreCase(input)) { // Check if input is "quit" (case-insensitive)

System.***out***.println("Exiting program. Goodbye!");

**break**; // Exit the loop

}

System.***out***.println("Uppercase: " + input.toUpperCase()); // Display the string in uppercase

}

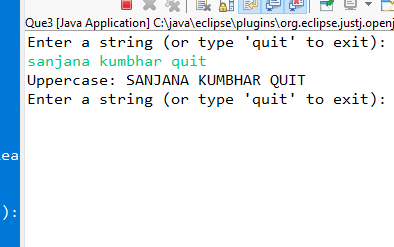
} **catch** (IOException e) {

System.***err***.println("An error occurred while reading input: " + e.getMessage());

}

}

}



4. Write a program to ask a record from user and display them .(using BufferedReader).

package io\_assignment;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.IOException;

public class UserRecord {

public static void main(String[] args) {

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

try {

System.out.println("Enter your name: ");

String name = reader.readLine();

System.out.println("Enter your age: ");

int age = Integer.parseInt(reader.readLine());

System.out.println("Enter your email: ");

String email = reader.readLine();

System.out.println("\n--- User Record ---");

System.out.println("Name: " + name);

System.out.println("Age: " + age);

System.out.println("Email: " + email);

} catch (IOException e) {

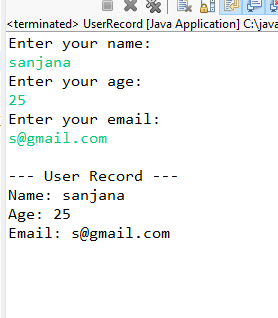
System.out.println("An error occurred while reading input.");

} catch (NumberFormatException e) {

System.out.println("Invalid age entered. Please enter a valid number.");

}

}

}  


5. Create a class file FileParse.java under c:\myjava folder. Then save and compile by giving the

javac FileParse.java command on command prompt. Write suitable methods in the class to read

an input file called input.txt under myjava and output its contents to console

Handle any Exceptions. Make sure no errors and fix if any. Run using command java FileParse

**package** io\_assignment.myjava;

**import** java.io.\*;

**public** **class** FileParse {

**public** **static** **void** main(String[] args) {

// Define the file path

String filePath = "D:\\cdac\\java\_core\\eclipse\\cdac\\src\\io\_assignment\\myjava\\input.txt";

// Read the file and display its contents

**try** (BufferedReader reader = **new** BufferedReader(**new** FileReader(filePath))) {

String line;

System.***out***.println("Contents of input.txt:");

**while** ((line = reader.readLine()) != **null**) {

System.***out***.println(line);

}

} **catch** (FileNotFoundException e) {

System.***err***.println("Error: File not found at " + filePath);

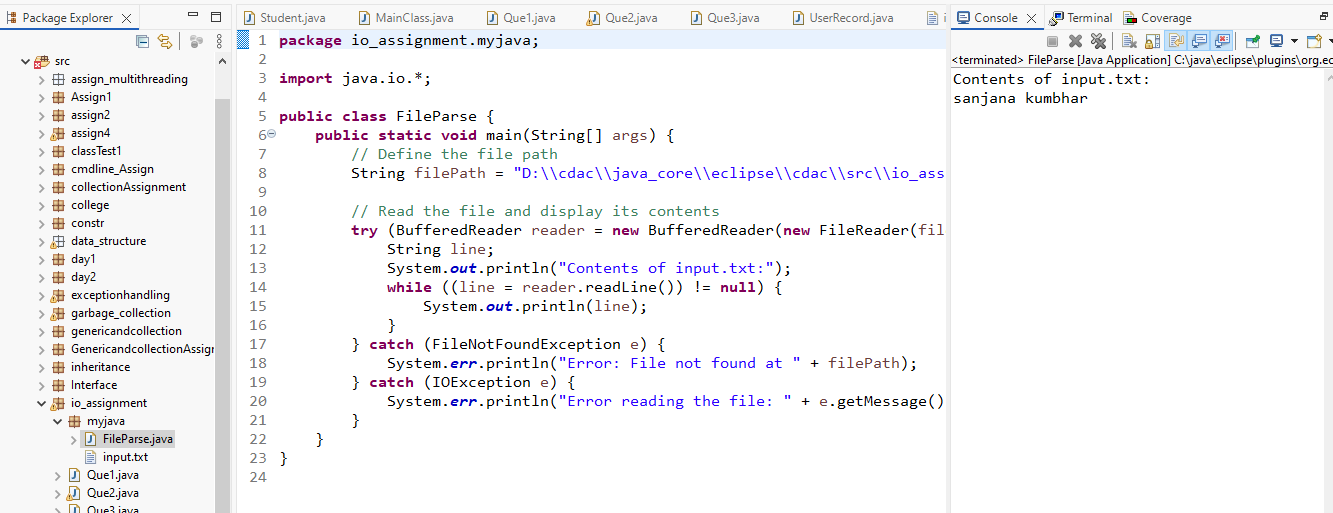
} **catch** (IOException e) {

System.***err***.println("Error reading the file: " + e.getMessage());

}

}

}



6. Write a program to copy content of one file to another .use command line argument to take file

name .

**package** io\_assignment.copyfile;

**import** java.io.\*;

**public** **class** FileCopy {

**public** **static** **void** main(String[] args) {

// Check if command-line arguments are provided

**if** (args.length < 2) {

System.***out***.println("Usage: java FileCopy <source\_file> <destination\_file>");

**return**;

}

// Get file names from command-line arguments

String sourceFile = args[0];

String destinationFile = args[1];

**try** (

FileInputStream inputStream = **new** FileInputStream(sourceFile);

FileOutputStream outputStream = **new** FileOutputStream(destinationFile)

) {

// Read from source and write to destination

**byte**[] buffer = **new** **byte**[1024]; // Buffer to store data temporarily

**int** bytesRead; // Number of bytes read

**while** ((bytesRead = inputStream.read(buffer)) != -1) {

// Correctly write only the read portion of the buffer

outputStream.write(buffer, 0, bytesRead);

}

System.***out***.println("File copied successfully from " + sourceFile + " to " + destinationFile);

} **catch** (FileNotFoundException e) {

System.***err***.println("Error: File not found - " + e.getMessage());

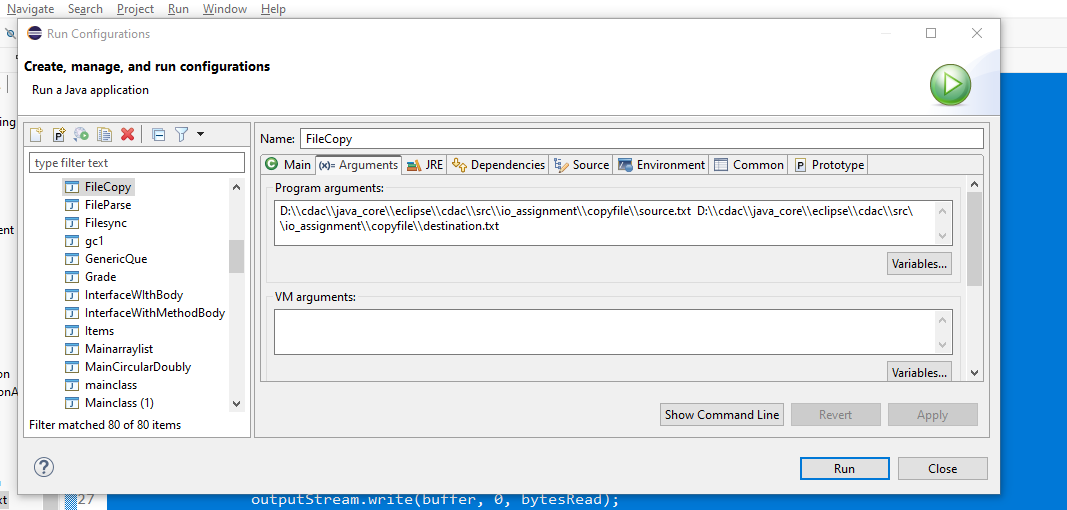
} **catch** (IOException e) {

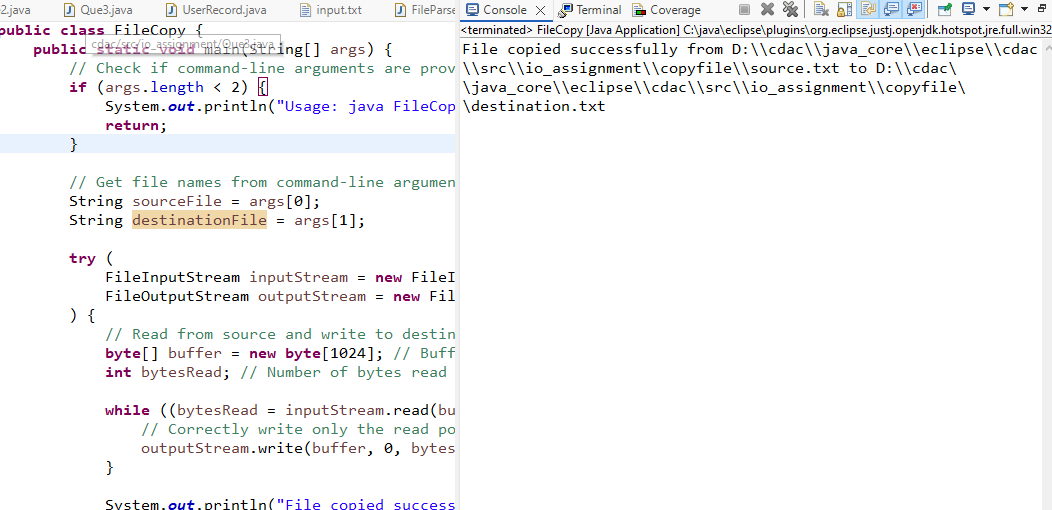
System.***err***.println("Error: I/O operation failed - " + e.getMessage());

}

}

}





7. Create an application that ask a record of person having name age, salary address using scanner

class object and store the complete record into file person.ser and display all records.

package io\_assignment;

import java.io.\*;

import java.util.\*;

public class Person implements Serializable {

private String name;

private int age;

private double salary;

private String address;

// Constructor

public Person(String name, int age, double salary, String address) {

this.name = name;

this.age = age;

this.salary = salary;

this.address = address;

}

// Getters and toString method for displaying information

public String getName() {

return name;

}

public int getAge() {

return age;

}

public double getSalary() {

return salary;

}

public String getAddress() {

return address;

}

@Override

public String toString() {

return "Name: " + name + ", Age: " + age + ", Salary: " + salary + ", Address: " + address;

}

}

class PersonApp {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// List to hold Person objects

List<Person> personList = new ArrayList<>();

// Loop to input multiple records

char choice;

do {

System.out.println("Enter person details:");

// Taking input for a new person

System.out.print("Enter Name: ");

String name = scanner.nextLine();

System.out.print("Enter Age: ");

int age = scanner.nextInt();

System.out.print("Enter Salary: ");

double salary = scanner.nextDouble();

scanner.nextLine(); // Consume the leftover newline

System.out.print("Enter Address: ");

String address = scanner.nextLine();

// Create a new Person object and add it to the list

Person person = new Person(name, age, salary, address);

personList.add(person);

System.out.print("Do you want to enter another record? (y/n): ");

choice = scanner.next().charAt(0);

scanner.nextLine(); // Consume the leftover newline

} while (choice == 'y' || choice == 'Y');

// Serialize the records to person.ser

try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("person.ser"))) {

out.writeObject(personList);

System.out.println("Records have been saved to person.ser.");

} catch (IOException e) {

System.err.println("Error saving records: " + e.getMessage());

}

// Deserialize and display the records

try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("person.ser"))) {

List<Person> deserializedList = (List<Person>) in.readObject();

System.out.println("\n--- Displaying all records ---");

for (Person p : deserializedList) {

System.out.println(p);

}

} catch (IOException | ClassNotFoundException e) {

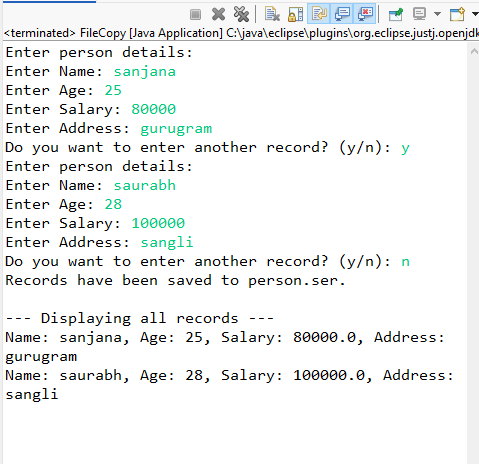
System.err.println("Error reading records: " + e.getMessage());

}

scanner.close();

}

}



8. Create a program that display information about file ,entered by user.

**package** io\_assignment;

**import** java.io.\*;

**import** java.util.Scanner;

**public** **class** FileInfo {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

// Ask the user for the file name

System.***out***.print("Enter the file name (with path if not in the current directory): ");

String fileName = scanner.nextLine();

// Create a File object

File file = **new** File(fileName);

// Check if the file exists

**if** (file.exists()) {

// Display file information

System.***out***.println("\nFile Information:");

System.***out***.println("File Name: " + file.getName());

System.***out***.println("Absolute Path: " + file.getAbsolutePath());

System.***out***.println("Parent Directory: " + file.getParent());

System.***out***.println("File Size: " + file.length() + " bytes");

System.***out***.println("Is File: " + file.isFile());

System.***out***.println("Is Directory: " + file.isDirectory());

System.***out***.println("Can Read: " + file.canRead());

System.***out***.println("Can Write: " + file.canWrite());

System.***out***.println("Can Execute: " + file.canExecute());

} **else** {

// If file does not exist

System.***out***.println("The file does not exist.");

}

scanner.close();

}

}

