**Код-блок для копирования:**





(Брельгин Василий Сергеевич) 30. Разработайте программу, которая заполняет список случайными числами. Количество

элементов и числовой диапазон вводятся пользователем. Программа должна проверять, входит ли число (также вводится пользователем) в данный список. Должен быть реализован бинарный поиск. Результаты должны сохраняться в MySQL/PostgreSQL и выводиться оттуда же.

import java.sql.\*;

import java.util.\*;

public class RandomSearch {

private static final String *url* = "jdbc:postgresql://localhost:5432/java\_lessons";

private static final String *user* = "ТВОЙ ЛОГИН";

private static final String *password* = "ТВОЙ ПАРОЛЬ";

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

private static List<Integer> getRandomList(int size, int min, int max) {

List<Integer> list = new ArrayList<>();

Random random = new Random();

for (int i = 0; i < size; i++) {

list.add(random.nextInt(max - min + 1) + min);

}

return list;

}

private static boolean binarySearch(List<Integer> list, int target) {

Collections.*sort*(list);

int left = 0;

int right = list.size() - 1;

while (left <= right) {

// Ищем середину

int mid = left + (right - left) / 2;

// Если элемент нашли – возвращаем True

if (list.get(mid) == target) {

return true;

// Если не нашли, смотрим, больше ли число по середине или меньше необходимого

} else if (list.get(mid) < target) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return false;

}

private static void saveAndShowResults(List<Integer> list, int target, boolean isFound) {

try (Connection connection = DriverManager.*getConnection*(*url*, *user*, *password*)) {

String createTableSQL = "CREATE TABLE IF NOT EXISTS search\_results (" + "id SERIAL PRIMARY KEY," + "numbers TEXT," + "target INT," + "found BOOLEAN" + ")";

try (Statement statement = connection.createStatement()) {

statement.execute(createTableSQL);

}

// Сохранение результатов в БД

String query = "INSERT INTO search\_results (numbers, target, found) VALUES (?, ?, ?)";

try (PreparedStatement stmt = connection.prepareStatement(query)) {

stmt.setString(1, list.toString());

stmt.setInt(2, target);

stmt.setBoolean(3, isFound);

stmt.executeUpdate();

}

// Вывод всех результатов из БД

String querySQL = "SELECT \* FROM search\_results";

try (Statement statement = connection.createStatement()) {

ResultSet resultSet = statement.executeQuery(querySQL);

System.*out*.println("\n\nПредыдещие результаты:");

while (resultSet.next()) {

String numbers = resultSet.getString("numbers");

int targetDigit = resultSet.getInt("target");

boolean found = resultSet.getBoolean("found");

System.*out*.println("Массив: " + numbers + ", Заданное число: " + targetDigit + ", Найдено?: " + found);

}

}

} catch (SQLException e) {

System.*out*.println("Ошибка при работе с БД" + e.getMessage());

}

}

public static void main(String[] args) {

System.*out*.println("Введи кол-во элементов массива: ");

int count = *scanner*.nextInt();

System.*out*.println("Введи минимальный элемент массива: ");

int minElem = *scanner*.nextInt();

System.*out*.println("Введи максимальный элемент массива: ");

int maxElem = *scanner*.nextInt();

List<Integer> generatedList = *getRandomList*(count, minElem, maxElem);

System.*out*.println("Сгенерированный массив: " + generatedList);

System.*out*.println("Число для поиска: ");

int elemToFind = *scanner*.nextInt();

boolean isFound = *binarySearch*(generatedList, elemToFind);

if (isFound) {

System.*out*.println("Заданное число было найдено!");

} else {

System.*out*.println("Заданное число НЕ найдено :(");

}

*saveAndShowResults*(generatedList, elemToFind, isFound);

}

}



(Брельгин Василий Сергеевич) 42. Написать на основе Spring Boot клиент-серверное приложение MyUser, в котором

можно управлять данными пользователей из базы данных через веб-интерфейс: имя, фамилия,

возраст, номер группы. База данных может быть любой – MySQL, PostgreSQL и т.д. При этом

должна быть доступна возможность добавления/удаления/редактирования пользователей.

<https://github.com/BBrel/javaExam42>

НЕ ЗАБУДЬ В ФАЙЛЕ application.properties ПОМЕНЯТЬ ВСЕ НА СВОИ ДАННЫЕ!!!



(Владимиров Илья Игоревич) 1. Условие: «Реализовать программу для выполнения следующих математических

операций с целочисленным, байтовым и вещественным типами данных: сложение, вычитание,

умножение, деление, деление по модулю (остаток), модуль числа, возведение в степень. Все

данные вводятся с клавиатуры (класс Scanner, System.in, nextint).» По данному условию

необходимо реализовать программу с интерактивным консольным меню, (т.е. вывод списка

действий по цифрам. При этом при нажатии на цифру у нас должно выполняться определенное

действие). При этом в программе данные пункты должны называться следующим образом:

1. Вывести все таблицы из MySQL.

2. Создать таблицу в MySQL.

3. Сложение чисел, результат сохранить в MySQL с последующим выводом в консоль.

4. Вычитание чисел, результат сохранить в MySQL с последующим выводом в консоль.

5. Умножение чисел, результат сохранить в MySQL с последующим выводом в консоль.

6. Деление чисел, результат сохранить в MySQL с последующим выводом в консоль.

7. Деление чисел по модулю (остаток), результат сохранить в MySQL с последующим

выводом в консоль.

8. Возведение числа в модуль, результат сохранить в MySQL с последующим выводом в

консоль.

9. Возведение числа в степень, результат сохранить в MySQL с последующим выводом в

консоль.

10. Сохранить все данные (вышеполученные результаты) из MySQL в Excel и вывести на

экран.

Класс Main

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Main {

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

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

int command = 0;

System.out.print("Введите название таблицы: ");

String tablename = scanner.nextLine();

while (command<10) {

System.out.println("""

Выбери одну из команд:\s

1. Вывести все таблицы из MySQL.\s

2. Создать таблицу в MySQL.\s

3. Сложение чисел\s

4. Вычитание чисел\s

5. Умножение чисел\s

6. Деление чисел\s

7. Деление чисел по модулю (остаток)\s

8. Возведение числа в модуль\s

9. Возведение числа в степень\s

10. Сохранить все данные и выйти\s

""");

command = get\_command();

chose\_module(command, tablename);

}

}

public static int get\_command() {

int operation = 0;

try {

do {

System.out.print("Введите нужную операцию: ");

operation = scanner.nextInt();

// Проверка условия неверного значения

if (operation < 1 || operation > 10) {

System.out.println("Неверный оператор. Попробуйте снова.");

}

} while (operation < 1 || operation > 10);

} catch (InputMismatchException e) {

System.out.println("Не совсем похоже на число :(");

scanner.next();

operation = get\_command();

}

return operation;

}

public static void chose\_module(int \_command, String tablename) throws SQLException {

switch(\_command) {

case 1:

ResultSet result\_table = SQL.get\_tables();

System.out.println("Доступные таблицы:");

while (result\_table.next()) {

System.out.println("-" + result\_table.getString(1));

}

break;

case 2:

System.out.print("Введите название для таблицы (название таблицы должно совпадать с вашим начальным названием): ");

String table\_name = scanner.next();

SQL.create\_new\_table(table\_name);

break;

case 10:

System.out.println("Ваши результаты:");

ResultSet result\_all = SQL.get\_all\_records(tablename);

System.out.printf("|%-10s | %-10s|\n", "ID", "РЕЗУЛЬТАТ");

while (result\_all.next()) {

System.out.printf("|%-10s | %-10s|\n", result\_all.getString(1), result\_all.getString(2));

}

SQL.save\_all\_to\_excel(tablename);

System.out.println("Данные успешно сохранены в Excel.");

break;

default:

Calculator.start(\_command, tablename);

break;

}

}

}



Класс Calculator

import java.sql.SQLException;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Calculator extends Main{

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

public static void start(int \_operation, String tablename) throws SQLException {

System.out.print("Введи первое число: ");

double num1 = get\_digit();

System.out.print("Введи второе число: ");

double num2 = get\_digit();

calculate(num1, num2, \_operation, tablename);

}

private static void calculate(double \_num1, double \_num2, int \_operation, String tablename) throws SQLException {

double result = 0;

switch(\_operation) {

case 3:

result = \_num1 + \_num2;

System.out.println(\_num1 + " + " + \_num2 + " = " + (result));

break;

case 4:

result = \_num1 - \_num2;

System.out.println(\_num1 + " - " + \_num2 + " = " + (result));

break;

case 5:

result = \_num1 \* \_num2;

System.out.println(\_num1 + " x " + \_num2 + " = " + (result));

break;

case 6:

result = \_num1 / \_num2;

System.out.println(\_num1 + " / " + \_num2 + " = " + (result));

break;

case 7:

result = \_num1 % \_num2;

System.out.println(\_num1 + " % " + \_num2 + " = " + (result));

break;

case 8:

result = Math.abs(\_num1);

double result2 = Math.abs(\_num2);

System.out.println("|" + \_num1 + "|" + " = " + result);

System.out.println("|" + \_num2 + "|" + " = " + result2);

break;

case 9:

result = Math.pow(\_num1, \_num2);

System.out.println(\_num1 + " \*\* " + \_num2 + " = " + result);

break;

}

SQL.add\_record(result, tablename);

}

private static double get\_digit() {

double digit = 0;

try {

digit = scanner.nextDouble();

} catch (InputMismatchException e) {

System.out.println("Не совсем похоже на число :(," +

"попробуй ввести его еще раз");

scanner.next();

digit = get\_digit();

}

return digit;

}

}



Класс SQL

import java.sql.\*;

public class SQL extends Main {

protected static String sql\_user = "root";

protected static String sql\_pswd = "root";

public static void setUser (String \_user) {

sql\_user = \_user;

}

public static void setPswd (String \_pswd) {

sql\_pswd = \_pswd;

}

public String getPswd () {

return sql\_pswd;

}

public String getUser () {

return sql\_user;

}

private static Connection ensure\_conn() throws SQLException{

DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver());

String conn\_string = "jdbc:mysql://localhost:3306/test";

return DriverManager.getConnection(conn\_string, sql\_user, sql\_pswd);

}

public static ResultSet get\_tables() throws SQLException {

Connection conn = ensure\_conn();

Statement stmt = conn.createStatement();

return stmt.executeQuery("SHOW TABLES");

}

public static void create\_new\_table(String \_name) throws SQLException {

Connection conn = ensure\_conn();

Statement stmt = conn.createStatement();

String query = "CREATE TABLE IF NOT EXISTS " + \_name + " (id int AUTO\_INCREMENT, " +

"result varchar(255), PRIMARY KEY (id))";

stmt.executeUpdate(query);

System.out.println("Таблица " + \_name + " была создана!");

}

public static void add\_record(double rec, String tablename) throws SQLException {

Connection conn = ensure\_conn();

Statement stmt = conn.createStatement();

String query = "INSERT INTO " + tablename + "(result) VALUES ("+rec+")";

stmt.executeUpdate(query);

}

public static ResultSet get\_all\_records(String tablename) throws SQLException {

Connection conn = ensure\_conn();

Statement stmt = conn.createStatement();

return stmt.executeQuery("SELECT \* from " + tablename + "");

}

public static void save\_all\_to\_excel(String tablename) throws SQLException {

Connection conn = ensure\_conn();

Statement stmt = conn.createStatement();

String query = "select 'id', 'result' union all select \* from " + tablename + " into outfile 'C:/ProgramData/MySQL/sql/test.xls' CHARACTER SET cp1251";

stmt.executeQuery(query);

}

}



(Голодных Павел Константинович) 8. Разработать программу для вывода на экран кубов первых десяти положительных

чисел.

public class CubeNumbers {

public static void main(String[] args) {

// Цикл от 1 до 10

for (int i = 1; i <= 10; i++) {

// Вычисляем куб числа

int cube = i \* i \* i;

// Выводим число и его куб

System.out.println("Число: " + i + ", Куб: " + cube);

}

}

}



(Голодных Павел Константинович) 12. Разработать класс-оболочку для числового типа double. Реализовать статические

методы сложения, деления, возведения в степень.

public class DoubleWrapper {

private double value;

// Конструктор

public DoubleWrapper(double value) {

this.value = value;

}

// Получение значения

public double getValue() {

return value;

}

// Статический метод сложения

public static DoubleWrapper add(DoubleWrapper a, DoubleWrapper b) {

return new DoubleWrapper(a.value + b.value);

}

// Статический метод деления

public static DoubleWrapper divide(DoubleWrapper a, DoubleWrapper b) {

if (b.value == 0) {

throw new IllegalArgumentException("Деление на ноль");

}

return new DoubleWrapper(a.value / b.value);

}

// Статический метод возведения в степень

public static DoubleWrapper power(DoubleWrapper base, int exponent) {

return new DoubleWrapper(Math.pow(base.value, exponent));

}

@Override

public String toString() {

return Double.toString(value);

}

// Пример использования класса

public static void main(String[] args) {

DoubleWrapper num1 = new DoubleWrapper(5.0);

DoubleWrapper num2 = new DoubleWrapper(3.0);

DoubleWrapper sum = DoubleWrapper.add(num1, num2);

DoubleWrapper quotient = DoubleWrapper.divide(num1, num2);

DoubleWrapper powerResult = DoubleWrapper.power(num1, 2);

System.out.println("Сумма: " + sum);

System.out.println("Частное: " + quotient);

System.out.println("Возведение в степень: " + powerResult);

}

}



(Гюджюйенер Тимур) 31. На основе класса BitSet разработайте программу для реализации битовых операций

AND, OR, XOR, а также маскирования.

import java.util.BitSet;

public class BitSetOperations {

public static void main(String[] args) {

// Создаем два BitSet

BitSet bitSet1 = new BitSet();

BitSet bitSet2 = new BitSet();

// Устанавливаем биты для bitSet1

bitSet1.set(0);

bitSet1.set(2);

bitSet1.set(4);

// Устанавливаем биты для bitSet2

bitSet2.set(1);

bitSet2.set(2);

bitSet2.set(3);

System.out.println("BitSet1: " + bitSet1);

System.out.println("BitSet2: " + bitSet2);

// AND операция

BitSet andResult = (BitSet) bitSet1.clone();

andResult.and(bitSet2);

System.out.println("AND: " + andResult);

// OR операция

BitSet orResult = (BitSet) bitSet1.clone();

orResult.or(bitSet2); I’m

System.out.println("OR: " + orResult);

// XOR операция

BitSet xorResult = (BitSet) bitSet1.clone();

xorResult.xor(bitSet2);

System.out.println("XOR: " + xorResult);

// Маскирование

BitSet mask = new BitSet();

mask.set(2);

mask.set(3);

BitSet maskedResult = (BitSet) bitSet1.clone();

maskedResult.and(mask);

System.out.println("Masked BitSet1 with mask: " + maskedResult);

}

}

### **Описание программы**

1. **Создание BitSet**:
   * Создаются два объекта BitSet, в которых устанавливаются определенные биты.
2. **Битовые операции**:
   * **AND**: Пересечение битов двух множеств.
   * **OR**: Объединение битов.
   * **XOR**: Исключающее ИЛИ (биты, отличающиеся в двух множествах).
3. **Маскирование**:
   * Применяется маска, чтобы выбрать только определенные биты.

### **Вывод программы**

При запуске программа покажет результат операций AND, OR, XOR, а также результата маскирования.

(Гюджюйенер Тимур) 33. Разработайте программу, которая получает в качестве параметра два числа –

количество строк и столбцов двумерной коллекции целых чисел. Коллекция заполняется

случайными числами, после чего на экран выводятся максимальное и минимальное значения с

индексами ячеек.

import java.util.Random;

public class MinMaxIn2DArray {

public static void main(String[] args) {

int rows = 5; // количество строк

int cols = 5; // количество столбцов

int[][] matrix = *createRandomMatrix*(rows, cols);

System.*out*.println("Сгенерированная матрица:");

*printMatrix*(matrix);

*findMinMax*(matrix);

}

// Метод для создания двумерного массива с случайными числами

public static int[][] createRandomMatrix(int rows, int cols) {

Random random = new Random();

int[][] matrix = new int[rows][cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

matrix[i][j] = random.nextInt(100); // случайные числа от 0 до 99

}

}

return matrix;

}

// Метод для вывода матрицы на экран

public static void printMatrix(int[][] matrix) {

for (int[] row : matrix) {

for (int value : row) {

System.*out*.print(value + " ");

}

System.*out*.println();

}

}

// Метод для нахождения минимального и максимального значений

public static void findMinMax(int[][] matrix) {

int minValue = Integer.*MAX\_VALUE*;

int maxValue = Integer.*MIN\_VALUE*;

int minRow = -1, minCol = -1;

int maxRow = -1, maxCol = -1;

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

for (int j = 0; j < matrix[i].length; j++) {

if (matrix[i][j] < minValue) {

minValue = matrix[i][j];

minRow = i;

minCol = j;

}

if (matrix[i][j] > maxValue) {

maxValue = matrix[i][j];

maxRow = i;

maxCol = j;

}

}

}

System.*out*.println("Минимальное значение: " + minValue + " по индексу (" + minRow + ", " + minCol + ")");

System.*out*.println("Максимальное значение: " + maxValue + " по индексу (" + maxRow + ", " + maxCol + ")");

}

}



(Дубинин Константин Андреевич) 18. Создать класс Binary для работы с двоичными числами фиксированной длины. Число

должно быть массивом тип char, каждый элемент которого принимает значение 0 или 1.

Младший бит имеет младший индекс. Отрицательные числа представляются в дополнительном

коде. Дополнительный код получается инверсией всех битов с прибавлением 1 к младшему биту.

Например, +1 – это в двоичном коде будет выглядеть, как 0000 0001. А -1 в двоичном коде будет

выглядеть, как 1111 1110 + 0000 0001 = 1111 1111. Создать методы конвертации десятичного

числа в массив и обратно.



public class Binary {

private char[] bits; // массив для хранения двоичного числа

private final int length; // фиксированная длина числа

// Конструктор - принимает длину числа

public Binary(int length) {

if (length <= 0) {

throw new IllegalArgumentException("Length must be greater than 0");

}

this.length = length;

this.bits = new char[length];

for (int i = 0; i < length; i++) { // инициализация массива нулями

bits[i] = '0';

}

}

// Метод для конвертации десятичного числа в двоичный массив

public void fromDecimal(int decimal) {

boolean isNegative = decimal < 0;

int absValue = Math.abs(decimal);

// Преобразование абсолютного значения в двоичный вид (без учета знака)

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

bits[i] = (absValue % 2 == 0) ? '0' : '1';

absValue /= 2;

}

if (isNegative) {

toTwosComplement(); // Преобразование в дополнительный код

}

// Проверка на переполнение

if (absValue > 0) {

throw new ArithmeticException("Overflow: Binary representation exceeds fixed length");

}

}

// Метод для преобразования числа в дополнительный код (для отрицательных чисел)

private void toTwosComplement() {

// Инверсия всех битов

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

bits[i] = (bits[i] == '0') ? '1' : '0';

}

// Добавляем 1 к младшему биту

boolean carry = true;

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

if (bits[i] == '0') {

bits[i] = '1';

carry = false;

break;

} else {

bits[i] = '0';

}

}

// Если после добавления 1 произошел перенос, он игнорируется (так как число ограничено длиной)

}

// Метод для конвертации двоичного массива обратно в десятичное число

public int toDecimal() {

boolean isNegative = (bits[length - 1] == '1'); // знак определяется старшим битом

int result = 0;

if (isNegative) {

toTwosComplement(); // Преобразуем из дополнительного кода в прямой

}

// Вычисляем значение числа

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

if (bits[i] == '1') {

result += (1 << i); // 2^i

}

}

if (isNegative) {

result = -result;

toTwosComplement(); // Возвращаем число в дополнительный код

}

return result;

}

// Метод для отображения числа в виде строки

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

for (int i = length - 1; i >= 0; i--) {

sb.append(bits[i]);

}

return sb.toString();

}

// Тестирование работы класса Binary

public static void main(String[] args) {

Binary binary = new Binary(8); // создаем число длиной 8 бит

// Пример с положительным числом

binary.fromDecimal(5);

System.out.println("Двоичное представление числа 5: " + binary);

System.out.println("Десятичное представление: " + binary.toDecimal());

// Пример с отрицательным числом

binary.fromDecimal(-5);

System.out.println("Двоичное представление числа -5: " + binary);

System.out.println("Десятичное представление: " + binary.toDecimal());//число инвертировано +1

// Пример с переполнением

try {

binary.fromDecimal(256); // 256 не помещается в 8 бит

} catch (ArithmeticException e) {

System.out.println(e.getMessage());

}

}

}



(Дубинин Константин Андреевич) 43. Написать на основе Spring Boot Security форму для авторизации и регистрации

пользователя. При этом после авторизации пользователя должно быть перенаправление на

главную страницу. Главная страница должна содержать запись «Hello World!». При этом до

авторизации главная страница не должна быть доступна для пользователя.

Скачать архив, открыть в иде, заменить бд в application.properties на свою.

К формам не получилось стили по-человечески добавить, а так всё работает

https://github.com/k-dubinin/task\_43\_reg\_login\_form.git

(Дульская Анастасия Владиславовна) 15. Разработать программу, в которой требуется создать абстрактный класс. В этом абстрактном классе определить абстрактные методы вычисления функции в определенной точке. Создать классы-наследники абстрактного класса, описывающими уравнения прямой и параболы. Программа должна выводить в консоль значение функции при вводе определенного значения.

// Шаг 1: Определение абстрактного класса

abstract class Function {

abstract double calculate(double x);

}

// Шаг 2: Создание классов-наследников

// Класс уравнения прямой

class LinearFunction extends Function {

private double slope; // Наклон

private double intercept; // Свободный член

public LinearFunction(double slope, double intercept) {

this.slope = slope;

this.intercept = intercept;

}

@Override

double calculate(double x) {

return slope \* x + intercept;

}

}

// Класс уравнения параболы

class QuadraticFunction extends Function {

private double a; // Коэффициент при x^2

private double b; // Коэффициент при x

private double c; // Свободный член

public QuadraticFunction(double a, double b, double c) {

this.a = a;

this.b = b;

this.c = c;

}

@Override

double calculate(double x) {

return a \* x \* x + b \* x + c;

}

}

// Шаг 3: Основной класс программы

import java.util.Scanner;

public class FunctionCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Создаем экземпляры функций

Function linearFunction = new LinearFunction(2, 3); // y = 2x + 3

Function quadraticFunction = new QuadraticFunction(1, -2, 1); // y = x^2 - 2x + 1

System.out.print("Введите значение x: ");

double x = scanner.nextDouble();

// Вычисляем и выводим значения функций

System.out.println("Значение линейной функции (y = 2x + 3): " + linearFunction.calculate(x));

System.out.println("Значение квадратичной функции (y = x^2 - 2x + 1): " + quadraticFunction.calculate(x));

scanner.close();

}

}



(Дульская Анастасия Владиславовна) 39. Разработать программу нахождения наибольшего общего делителя двух натуральных чисел. Требуется реализовать рекурсивный и без рекурсии варианты. Результат сохранить в MySQL/PostgreSQL.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;

public class GCDProgram {

// Метод для рекурсивного вычисления НОД

public static int gcdRecursive(int a, int b) {

if (b == 0) {

return a;

}

return gcdRecursive(b, a % b);

}

// Метод для нерекурсивного вычисления НОД

public static int gcdIterative(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

// Метод для создания базы данных и таблицы

public static void createDatabaseAndTable() {

String url = "jdbc:mysql://localhost/";

String user = "ВАШЕ\_ИМЯ\_ПОЛЬЗОВАТЕЛЯ";

String password = "ВАШ\_ПАРОЛЬ";

try (Connection connection = DriverManager.getConnection(url, user, password);

Statement statement = connection.createStatement()) {

// Создание базы данных

String createDatabaseQuery = "CREATE DATABASE IF NOT EXISTS gcd\_database";

statement.executeUpdate(createDatabaseQuery);

// Создание таблицы

String createTableQuery = "CREATE TABLE IF NOT EXISTS gcd\_results (" +

"id INT AUTO\_INCREMENT PRIMARY KEY," +

"number1 INT NOT NULL," +

"number2 INT NOT NULL," +

"gcd\_value INT NOT NULL)";

statement.executeUpdate(createTableQuery);

System.out.println("База данных и таблица успешно созданы.");

} catch (SQLException e) {

System.out.println("Ошибка при работе с базой данных: " + e.getMessage());

}

}

// Метод для сохранения результата в базу данных

public static void saveGCD(int a, int b, int gcd) {

String url = "jdbc:mysql://localhost:3306/gcd\_database";

String user = "ВАШЕ\_ИМЯ\_ПОЛЬЗОВАТЕЛЯ";

String password = "ВАШ\_ПАРОЛЬ";

String query = "INSERT INTO gcd\_results (number1, number2, gcd\_value) VALUES (?, ?, ?)";

try (Connection connection = DriverManager.getConnection(url, user, password);

PreparedStatement preparedStatement = connection.prepareStatement(query)) {

preparedStatement.setInt(1, a);

preparedStatement.setInt(2, b);

preparedStatement.setInt(3, gcd);

preparedStatement.executeUpdate();

System.out.println("Результат успешно сохранен в базе данных.");

} catch (SQLException e) {

System.out.println("Ошибка при работе с базой данных: " + e.getMessage());

}

}

public static void main(String[] args) {

createDatabaseAndTable();

Scanner scanner = new Scanner(System.in);

System.out.print("Введите первое натуральное число: ");

int a = scanner.nextInt();

System.out.print("Введите второе натуральное число: ");

int b = scanner.nextInt();

// Вычисление НОД рекурсивным методом

int gcdRec = gcdRecursive(a, b);

System.out.println("Рекурсивный НОД: " + gcdRec);

// Вычисление НОД нерекурсивным методом

int gcdIter = gcdIterative(a, b);

System.out.println("Нерекурсивный НОД: " + gcdIter);

// Сохранение результатов в базу данных

saveGCD(a, b, gcdRec);

scanner.close();

}

}



(Жеребчиков Никита Алексеевич) 44. Разработать MVC-приложение арифметический калькулятор на основе Spring Boot.

Применить шаблонизатор Thymeleaf. Все результаты вычисления должны сохраняться и

выводиться из MySQL.



Скачать архив, открыть в иде, заменить ссылку на бд в application.properties на свою.  
https://github.com/DevNikitaZherebchikoff/Calculator\_44.git

(Жеребчиков Никита Алексеевич) 45. Разработка веб-MVC приложения на основе Spring Boot. Приложение должно

генерировать последовательность из 1000 случайных чисел в диапазоне, заданном

пользователем, и выводит эти числа на экран и вычисляет их среднее арифметическое



https://github.com/DevNikitaZherebchikoff/average\_45.git

(Звягин Данила Михайлович) 3. Разработайте программу, которая выводит в консоль все цифры, входящие в

натуральное число n. К примеру, если дано число 2359, то в консоль выводятся отдельно числа

2, 3, 5, 9.

 public static void task\_3() {

Scanner scanner = new Scanner(System.in); // Создаём объект класса Scanner

String line = scanner.nextLine(); // Считываем строку

// Интегрируемся по строке и печатаем каждый символ

for (String s : line.split("")) { // Разбиваем строку на символы

if (s.equals(line.split("")[line.length()-1])) { // Проверяем, являетс

я ли элемент последним в строке

System.out.print(s); // Печатаем элемент

} else { // Если элемент не последний

System.out.print(s + ","); // Печатаем элемент и запятую

}

}

}



(Звягин Данила Михайлович) 16. Создать интерфейс Progress c методами вычисления любого элемента прогрессии и

суммы прогрессии. Разработать классы арифметической и геометрической прогрессии, которые

имплементируют интерфейс Progress.

public class Main {

public static void main(String[] args) {

Progress arithmeticProgression = new ArithmeticProgression(2, 3);

System.*out*.println("Арифметическая прогрессия:");

System.*out*.println("5-й элемент: " + arithmeticProgression.getElement(5));

System.*out*.println("Сумма первых 5 элементов: " + arithmeticProgression.getSum(5));

Progress geometricProgression = new GeometricProgression(2, 3);

System.*out*.println("\nГеометрическая прогрессия:");

System.*out*.println("5-й элемент: " + geometricProgression.getElement(5));

System.*out*.println("Сумма первых 5 элементов: " + geometricProgression.getSum(5));

-----------------------------------------------------------------

public class GeometricProgression implements Progress {

private double firstTerm; // Первый элемент прогрессии

private double commonRatio; // Знаменатель прогрессии

public GeometricProgression(double firstTerm, double commonRatio) {

this.firstTerm = firstTerm;

this.commonRatio = commonRatio;

}

@Override

public double getElement(int n) {

return firstTerm \* Math.pow(commonRatio, n - 1);

}

@Override

public double getSum(int n) {

if (commonRatio == 1) {

return firstTerm \* n; // Если знаменатель равен 1

} else {

return firstTerm \* (1 - Math.pow(commonRatio, n)) / (1 - commonRatio);

}

}

}

--------------------------------------------------------------------------

public class ArithmeticProgression implements Progress {

private double firstTerm; // Первый элемент прогрессии

private double commonDifference; // Разность прогрессии

public ArithmeticProgression(double firstTerm, double commonDifference) {

this.firstTerm = firstTerm;

this.commonDifference = commonDifference;

}

@Override

public double getElement(int n) {

return firstTerm + (n - 1) \* commonDifference;

}

@Override

public double getSum(int n) {

return (n / 2.0) \* (2 \* firstTerm + (n - 1) \* commonDifference);

}

}

--------------------------------------------------------------------

public interface Progress {

double getElement(int n);

double getSum(int n);

}



(Исломов Мехродж Хасанович) 4. Написать калькулятор для строковых выражений вида "<число> <операция> <число>"

,

где <число> - положительное целое число меньшее 10, записанное словами, например, "четыре",

<арифметическая операция> - одна из операций "плюс", "минус", "умножить". Результат

выполнения операции вернуть в виде текстового представления числа. Пример: "пять плюс

четыре" --> "девять".

package com.example.demo;

import java.io.PrintStream;

import java.io.UnsupportedEncodingException;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class t1 {

// Словарь для преобразования текстовых чисел в числовые

private static final Map<String, Integer> *numberMap* = new HashMap<>();

static {

*numberMap*.put("ноль", 0);

*numberMap*.put("один", 1);

*numberMap*.put("два", 2);

*numberMap*.put("три", 3);

*numberMap*.put("четыре", 4);

*numberMap*.put("пять", 5);

*numberMap*.put("шесть", 6);

*numberMap*.put("семь", 7);

*numberMap*.put("восемь", 8);

*numberMap*.put("девять", 9);

}

// Словарь для преобразования текстовых операций в арифметические

private static final Map<String, String> *operationMap* = new HashMap<>();

static {

*operationMap*.put("плюс", "+");

*operationMap*.put("минус", "-");

*operationMap*.put("умножить на", "\*");

}

// Функция для преобразования числа в текстовое представление

private static String numberToText(int number) {

if (number < 0) {

return "минус " + *numberToText*(-number); // Обработка отрицательных чисел

}

String[] units = {"ноль", "один", "два", "три", "четыре", "пять", "шесть", "семь", "восемь", "девять"};

String[] teens = {"десять", "одиннадцать", "двенадцать", "тринадцать", "четырнадцать", "пятнадцать", "шестнадцать", "семнадцать", "восемнадцать", "девятнадцать"};

String[] tens = {"", "десять", "двадцать", "тридцать", "сорок", "пятьдесят", "шестьдесят", "семьдесят", "восемьдесят", "девяносто"};

if (number < 10) {

return units[number]; // Числа от 0 до 9

} else if (number < 20) {

return teens[number - 10]; // Числа от 10 до 19

} else if (number < 100) {

return tens[number / 10] + (number % 10 != 0 ? " " + units[number % 10] : ""); // Числа от 20 до 99

} else {

return "сто"; // Максимальное число, которое может получиться: 9 \* 9 = 81

}

}

public static String calculate(String expression) {

try {

// Разбиваем строку на части

String[] parts = expression.split(" ");

if (parts.length < 3 || parts.length > 4) {

throw new IllegalArgumentException("Неверный формат выражения. Ожидается: <число> <операция> <число>");

}

// Определяем операцию

String operationText;

if (parts.length == 3) {

operationText = parts[1]; // Операция из одного слова (например, "плюс")

} else {

operationText = parts[1] + " " + parts[2]; // Операция из двух слов (например, "умножить на")

}

// Преобразуем текстовые числа в числовые

String num1Text = parts[0];

String num2Text = parts[parts.length - 1]; // Последнее слово — второе число

int num1 = *numberMap*.getOrDefault(num1Text, -1);

int num2 = *numberMap*.getOrDefault(num2Text, -1);

if (num1 == -1 || num2 == -1) {

throw new IllegalArgumentException("Неверное число в выражении. Допустимые числа: ноль, один, два, ..., девять");

}

// Преобразуем текстовую операцию в арифметическую

String operation = *operationMap*.getOrDefault(operationText, "");

if (operation.isEmpty()) {

throw new IllegalArgumentException("Неверная операция. Допустимые операции: плюс, минус, умножить на");

}

// Выполняем арифметическую операцию

int result;

switch (operation) {

case "+":

result = num1 + num2;

break;

case "-":

result = num1 - num2;

break;

case "\*":

result = num1 \* num2;

break;

default:

throw new IllegalArgumentException("Неверная операция");

}

// Преобразуем результат обратно в текстовое представление

return *numberToText*(result);

} catch (Exception e) {

return "Ошибка: " + e.getMessage(); // Возвращаем сообщение об ошибке

}

}

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

// Устанавливаем кодировку UTF-8 для вывода

PrintStream out = new PrintStream(System.*out*, true, "UTF-8");

// Создаем Scanner для ввода с консоли

Scanner scanner = new Scanner(System.*in*, "UTF-8");

out.println("Введите выражение в формате: <число> <операция> <число>");

out.println("Пример: пять плюс четыре");

out.println("Доступные операции: плюс, минус, умножить на");

while (true) {

out.print("Введите выражение (или 'выход' для завершения): ");

String input = scanner.nextLine().trim(); // Считываем ввод пользователя

if (input.equalsIgnoreCase("выход")) {

out.println("Программа завершена.");

break;

}

// Вычисляем результат и выводим его

String result = *calculate*(input);

out.println(input + " --> " + result);

}

scanner.close(); // Закрываем Scanner

}

}



(Исломов Мехродж Хасанович) 22. Разработайте программу-генератор рабочего календаря. Слесарь механосборочного

цеха работает сутки через трое. Если смена попадает на воскресенье, то переносится на

понедельник. По введенной дате программа должна генерировать расписание из дат на текущий

месяц на 2022 год.

import java.time.DayOfWeek;

import java.time.LocalDate;

import java.time.YearMonth;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class WorkCalendarGenerator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите начальную дату (гггг-мм-дд): ");

String inputDate = scanner.nextLine();

LocalDate startDate = LocalDate.parse(inputDate);

List<LocalDate> schedule = generateSchedule(startDate);

System.out.println("Расписание смен на месяц:");

for (LocalDate date : schedule) {

System.out.println(date);

}

}

public static List<LocalDate> generateSchedule(LocalDate startDate) {

List<LocalDate> schedule = new ArrayList<>();

LocalDate currentDate = startDate;

YearMonth yearMonth = YearMonth.from(startDate);

// Перебираем дни месяца

while (currentDate.getMonth() == yearMonth.getMonth()) {

// Если текущий день - воскресенье, переносим смену на понедельник

if (currentDate.getDayOfWeek() == DayOfWeek.SUNDAY) {

currentDate = currentDate.plusDays(1);

}

// Добавляем дату смены в расписание

schedule.add(currentDate);

// Переходим к следующей смене (через 3 дня)

currentDate = currentDate.plusDays(4);

}

return schedule;

}

}



(Карибов Георгис Одиссеевич) 6. Разработайте программу, которая выводит буквы английского алфавита, используя

цикл while в MySQL/PostgreSQL.

DELIMITER *//*

CREATE PROCEDURE PrintAlphabet()

BEGIN

DECLARE i INT DEFAULT 65; *-- ASCII-код для 'A'*

WHILE i <= 90 DO *-- ASCII-код для 'Z'*

SELECT CHAR(i); *-- выводим букву*

SET i = i + 1; *-- увеличиваем код на 1*

END WHILE;

END *//*

DELIMITER ;

*-- Вызов процедуры*

CALL PrintAlphabet();

ЭТО КОД ДЛЯ MYSQL !!!!!!!!!!!

(Карибов Георгис Одиссеевич) 24. Создайте класс Form - оболочку для создания и ввода пароля. Он должен иметь методы

input, submit, password. Создайте класс SmartForm, который будет наследовать от Form и

сохранять значения password.

import java.util.Scanner;

public class Form {

private String password;

*// Метод для ввода пароля*

public void input() {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите пароль: ");

password = scanner.nextLine();

}

*// Метод для отправки формы*

public void submit() {

if (password != null && !password.isEmpty()) {

System.out.println("Форма успешно отправлена.");

} else {

System.out.println("Пароль не введен. Пожалуйста, введите пароль перед отправкой.");

}

}

*// Метод для получения пароля*

public String getPassword() {

return password;

}

}



import java.util.ArrayList;

import java.util.List;

public class SmartForm extends Form {

private List<String> passwordHistory;

public SmartForm() {

passwordHistory = new ArrayList<>();

}

@Override

public void input() {

super.input(); *// Вызов метода input родительского класса*

String currentPassword = getPassword();

if (currentPassword != null && !currentPassword.isEmpty()) {

passwordHistory.add(currentPassword); *// Сохраняем пароль в истории*

System.out.println("Пароль сохранен.");

}

}

*// Метод для получения истории паролей*

public List<String> getPasswordHistory() {

return passwordHistory;

}

}





public class Main {

public static void main(String[] args) {

SmartForm smartForm = new SmartForm();

*// Ввод пароля*

smartForm.input();

*// Отправка формы*

smartForm.submit();

*// Вывод истории паролей*

System.out.println("История паролей: " + smartForm.getPasswordHistory());

}

}



(Комаров Иван Александрович) 7. Напишите программу, которая будет выводить в консоль введённое слово 6 раз и

сохранять в MySQL/PostgreSQL.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.Scanner;

public class WordSaver {

private static final String MYSQL\_URL = "jdbc:mysql://localhost:3306/your\_database";

private static final String MYSQL\_USER = "your\_mysql\_user";

private static final String MYSQL\_PASSWORD = "your\_mysql\_password";

private static final String POSTGRESQL\_URL = "jdbc:postgresql://localhost:5432/your\_database";

private static final String POSTGRESQL\_USER = "your\_postgres\_user";

private static final String POSTGRESQL\_PASSWORD = "your\_postgres\_password";

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите слово: ");

String word = scanner.nextLine();

// Вывод слова 6 раз

for (int i = 0; i < 6; i++) {

System.out.println(word);

}

// Сохранение в MySQL

saveToMySQL(word);

// Сохранение в PostgreSQL

saveToPostgreSQL(word);

scanner.close();

}

private static void saveToMySQL(String word) {

try (Connection connection = DriverManager.getConnection(MYSQL\_URL, MYSQL\_USER, MYSQL\_PASSWORD)) {

String createTableSQL = "CREATE TABLE IF NOT EXISTS words (id INT AUTO\_INCREMENT PRIMARY KEY, word VARCHAR(255) NOT NULL)";

String insertSQL = "INSERT INTO words (word) VALUES (?)";

try (PreparedStatement createTableStatement = connection.prepareStatement(createTableSQL)) {

createTableStatement.execute();

}

try (PreparedStatement insertStatement = connection.prepareStatement(insertSQL)) {

insertStatement.setString(1, word);

insertStatement.executeUpdate();

}

System.out.println("Слово сохранено в MySQL.");

} catch (SQLException e) {

e.printStackTrace();

}

}

private static void saveToPostgreSQL(String word) {

try (Connection connection = DriverManager.getConnection(POSTGRESQL\_URL, POSTGRESQL\_USER, POSTGRESQL\_PASSWORD)) {

String createTableSQL = "CREATE TABLE IF NOT EXISTS words (id SERIAL PRIMARY KEY, word VARCHAR(255) NOT NULL)";

String insertSQL = "INSERT INTO words (word) VALUES (?)";

try (PreparedStatement createTableStatement = connection.prepareStatement(createTableSQL)) {

createTableStatement.execute();

}

try (PreparedStatement insertStatement = connection.prepareStatement(insertSQL)) {

insertStatement.setString(1, word);

insertStatement.executeUpdate();

}

System.out.println("Слово сохранено в PostgreSQL.");

} catch (SQLException e) {

e.printStackTrace();

}

}

}



(Комаров Иван Александрович) 14. Разработать программу, в которой требуется создать класс, описывающий

геометрическую фигуру – треугольник. Методами класса должны быть – вычисление площади,

периметра. Создать класс-наследник, определяющий прямоугольный треугольник.

import java.util.Scanner;

// Класс, описывающий треугольник

class Triangle {

protected double a; // Сторона a

protected double b; // Сторона b

protected double c; // Сторона c

// Конструктор

public Triangle(double a, double b, double c) {

if (isValidTriangle(a, b, c)) {

this.a = a;

this.b = b;

this.c = c;

} else {

throw new IllegalArgumentException("Стороны не могут образовать треугольник.");

}

}

// Метод для проверки валидности треугольника

private boolean isValidTriangle(double a, double b, double c) {

return (a + b > c) && (a + c > b) && (b + c > a);

}

// Метод для вычисления периметра

public double perimeter() {

return a + b + c;

}

// Метод для вычисления площади по формуле Герона

public double area() {

double s = perimeter() / 2; // Полупериметр

return Math.*sqrt*(s \* (s - a) \* (s - b) \* (s - c));

}

}

// Класс, описывающий прямоугольный треугольник

class RightTriangle extends Triangle {

// Конструктор принимает два катета

public RightTriangle(double a, double b) {

super(a, b, Math.*sqrt*(a \* a + b \* b)); // Вычисляем гипотенузу

}

// Переопределяем метод для вычисления площади

@Override

public double area() {

return 0.5 \* a \* b; // Площадь прямоугольного треугольника

}

}

// Главный класс для тестирования

public class Main {

public static void main(String[] args) {

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

// Ввод для обычного треугольника

System.*out*.println("Введите стороны треугольника (a, b, c):");

double a = scanner.nextDouble();

double b = scanner.nextDouble();

double c = scanner.nextDouble();

try {

Triangle triangle = new Triangle(a, b, c);

System.*out*.println("Обычный треугольник:");

System.*out*.println("Периметр: " + triangle.perimeter());

System.*out*.println("Площадь: " + triangle.area());

} catch (IllegalArgumentException e) {

System.*out*.println(e.getMessage());

}

// Ввод для прямоугольного треугольника

System.*out*.println("\nВведите катеты прямоугольного треугольника (a, b):");

double catetA = scanner.nextDouble();

double catetB = scanner.nextDouble();

RightTriangle rightTriangle = new RightTriangle(catetA, catetB);

System.*out*.println("Прямоугольный треугольник:");

System.*out*.println("Периметр: " + rightTriangle.perimeter());

System.*out*.println("Площадь: " + rightTriangle.area());

scanner.close();

}

}



(Котов Егор Владимирович) 13. Разработать программу, которая заполняет двумерный массив случайными

положительными числами в диапазоне от 1 до 100 до тех пор, пока сумма граничных элементов

не станет равной 666. Пользователь вначале вводит размер матрицы.

import java.util.Random;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите размер матрицы (например, для матрицы 3x3 введите 3): ");

int size = scanner.nextInt();

scanner.close();

int[][] matrix = new int[size][size];

Random random = new Random();

while (true) {

// Заполнить матрицу случайными числами

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

matrix[i][j] = random.nextInt(100) + 1; // Случайное число от 1 до 100

}

}

// Проверить сумму граничных элементов

if (checkBorderSum(matrix, 666)) {

break;

}

}

// Вывести матрицу

printMatrix(matrix);

}

private static boolean checkBorderSum(int[][] matrix, int targetSum) {

int sum = 0;

// Сумма верхней и нижней границ

for (int j = 0; j < matrix[0].length; j++) {

sum += matrix[0][j]; // Верхняя граница

sum += matrix[matrix.length - 1][j]; // Нижняя граница

}

// Сумма левой и правой границ (за вычетом углов, чтобы не посчитать дважды)

for (int i = 1; i < matrix.length - 1; i++) {

sum += matrix[i][0]; // Левая граница

sum += matrix[i][matrix[0].length - 1]; // Правая граница

}

return sum == targetSum;

}

private static void printMatrix(int[][] matrix) {

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

for (int j = 0; j < matrix[0].length; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

}

}



(Котов Егор Владимирович) 26. Создайте класс ColorModel для определения цветовой модели. Разработайте

подклассы RGBconverter и CMYKconverter для конвертации цвета из одной модели в другую.

Конвертация CMYK в RGB производится по следующим формулам: R = 255 × (1-C) × (1-K), G =

255 × (1-M) × (1-K), B = 255 × (1-Y) × (1-K) (где R – red, G – green, B – black, C – Cyan, M - Magenta,

Y - Yellow, K- Black))

import java.util.Scanner;

public abstract class ColorModel {

public abstract void convertToRGB(double[] cmyk);

public abstract void convertToCMYK(int[] rgb);

}

class RGBConverter extends ColorModel {

@Override

public void convertToRGB(double[] cmyk) {

// Конвертация CMYK в RGB

int r = (int) (255 \* (1 - cmyk[0]) \* (1 - cmyk[3]));

int g = (int) (255 \* (1 - cmyk[1]) \* (1 - cmyk[3]));

int b = (int) (255 \* (1 - cmyk[2]) \* (1 - cmyk[3]));

System.out.println("RGB: (" + r + ", " + g + ", " + b + ")");

}

@Override

public void convertToCMYK(int[] rgb) {

// Конвертация RGB в CMYK

double r = rgb[0] / 255.0;

double g = rgb[1] / 255.0;

double b = rgb[2] / 255.0;

double k = 1 - Math.max(Math.max(r, g), b);

double c = (1 - r - k) / (1 - k);

double m = (1 - g - k) / (1 - k);

double y = (1 - b - k) / (1 - k);

if (k == 1) {

c = m = y = 0;

}

System.out.println("CMYK: (" + c + ", " + m + ", " + y + ", " + k + ")");

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Выберите действие:");

System.out.println("1. Конвертировать CMYK в RGB");

System.out.println("2. Конвертировать RGB в CMYK");

int choice = scanner.nextInt();

RGBConverter converter = new RGBConverter();

if (choice == 1) {

System.out.print("Введите значения CMYK (C, M, Y, K) через пробел: ");

double c = scanner.nextDouble();

double m = scanner.nextDouble();

double y = scanner.nextDouble();

double k = scanner.nextDouble();

converter.convertToRGB(new double[] {c, m, y, k});

} else if (choice == 2) {

System.out.print("Введите значения RGB (R, G, B) через пробел: ");

int r = scanner.nextInt();

int g = scanner.nextInt();

int b = scanner.nextInt();

converter.convertToCMYK(new int[] {r, g, b});

} else {

System.out.println("Неправильный выбор");

}

scanner.close();

}

}



(Курушин Антон Александрович) 11. Разработайте программу, реализующую рекурсивное вычисление факториала.

import java.util.Scanner;

public class FactorialCalculator {

// Метод для рекурсивного вычисления факториала

public static long factorial(int n) {

if (n == 0 || n == 1) {

return 1; // Базовый случай: 0! = 1! = 1

}

return n \* factorial(n - 1); // Рекурсивный случай: n! = n \* (n-1)!

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Введите неотрицательное целое число: ");

int number = scanner.nextInt();

if (number < 0) {

System.out.println("Ошибка: введите неотрицательное число.");

} else {

System.out.println("Факториал числа " + number + " равен " + factorial(number) + ".");

}

} catch (Exception e) {

System.out.println("Ошибка: введите целое число.");

} finally {

scanner.close(); // Закрываем сканер

}

}

}



(Курушин Антон Александрович) 27. Создайте класс Number для конвертации десятичного числа в бинарный,

восьмеричный, шестнадцатеричный вид. Реализовать в виде статических методов класса. Числа

вводятся с клавиатуры с запросом в какой численный вид конвертировать.

import java.util.Scanner;

public class Number {

// Метод для конвертации в двоичную систему

public static String toBinary(int decimalNumber) {

return Integer.toBinaryString(decimalNumber);

}

// Метод для конвертации в восьмеричную систему

public static String toOctal(int decimalNumber) {

return Integer.toOctalString(decimalNumber);

}

// Метод для конвертации в шестнадцатеричную систему

public static String toHexadecimal(int decimalNumber) {

return Integer.toHexString(decimalNumber).toUpperCase();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Введите десятичное число: ");

int decimalNumber = scanner.nextInt();

System.out.println("Выберите численный вид для конвертации:");

System.out.println("1. Двоичный");

System.out.println("2. Восьмеричный");

System.out.println("3. Шестнадцатеричный");

System.out.print("Введите номер (1/2/3): ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.println("Двоичный вид: " + toBinary(decimalNumber));

break;

case 2:

System.out.println("Восьмеричный вид: " + toOctal(decimalNumber));

break;

case 3:

System.out.println("Шестнадцатеричный вид: " + toHexadecimal(decimalNumber));

break;

default:

System.out.println("Ошибка: некорректный выбор.");

}

} catch (Exception e) {

System.out.println("Ошибка: введите корректное число.");

} finally {

scanner.close(); // Закрытие сканера

}

}

}



(Логинов Пётр Константинович) 41. Условие задачи: «Ввести две строки (не менее 50 символов каждая) с клавиатуры.

Необходимо вывести на экран две введенных ранее строки, подсчитать и вывести размер длины

каждой строки, объединить данные строки в одну, сравнить данные строки и результат сравнения

вывести на экран». По данному условию необходимо реализовать программу с интерактивным

консольным меню, (т.е. вывод списка действий по цифрам. При этом при нажатии на цифру у нас

должно выполняться определенное действие). При этом в программе данные пункты должны

называться следующим образом:

1. Вывести все таблицы из MySQL.

2. Создать таблицу в MySQL.

3. Ввести две строки с клавиатуры, результат сохранить в MySQL с последующим

выводом в консоль.

4. Подсчитать размер ранее введенных строк, результат сохранить в MySQL с

последующим выводом в консоль.

5. Объединить две строки в единое целое, результат сохранить в MySQL с последующим

выводом в консоль.

6. Сравнить две ранее введенные строки, результат сохранить в MySQL с последующим

выводом в консоль.

7. Сохранить все данные (вышеполученные результаты) из MySQL в Excel и вывести на

экран.





(Логинов Пётр Константинович) 23. Разработать класс для представления комплексных чисел с возможностью задания

вещественной и мнимой частей в виде массива из двух чисел типа int. Определить методы для

выполнения операций сложения, вычитания и умножения комплексных чисел.





(Мещерякова Ирина Владимировна) 28. Разработать класс Neuron для реализации нейронной сети из двух нейронов и одного

выхода. Сделать функцию прямого распространения с функцией активации в виде сигмоиды.

public class Neuron {

private double[] weights; // Веса нейронов

private double bias; // Смещение (bias)

// Конструктор для инициализации весов и смещения

public Neuron(int inputSize) {

weights = new double[inputSize];

for (int i = 0; i < inputSize; i++) {

weights[i] = Math.random(); // Инициализация случайными значениями

}

bias = Math.random(); // Инициализация смещения

}

// Функция активации (сигмоида)

private double sigmoid(double x) {

return 1 / (1 + Math.exp(-x));

}

// Прямое распространение

public double feedForward(double[] inputs) {

if (inputs.length != weights.length) {

throw new IllegalArgumentException("Input size must match the number of weights.");

}

double weightedSum = 0;

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

weightedSum += inputs[i] \* weights[i]; // Вычисляем взвешенную сумму

}

weightedSum += bias; // Добавляем смещение

return sigmoid(weightedSum); // Применяем функцию активации

}

// Метод для получения весов (для дальнейшего использования, например, в обучении)

public double[] getWeights() {

return weights;

}

// Метод для получения смещения

public double getBias() {

return bias;

}

}

public class SimpleNeuralNetwork {

private Neuron neuron1;

private Neuron neuron2;

private Neuron outputNeuron;

public SimpleNeuralNetwork() {

neuron1 = new Neuron(2); // Первый нейрон с двумя входами

neuron2 = new Neuron(2); // Второй нейрон с двумя входами

outputNeuron = new Neuron(2); // Выходной нейрон с двумя входами (выходы первых двух нейронов)

}

public double feedForward(double[] inputs) {

double output1 = neuron1.feedForward(inputs);

double output2 = neuron2.feedForward(inputs);

return outputNeuron.feedForward(new double[]{output1, output2}); // Прямое распространение через выходной нейрон

}

public static void main(String[] args) {

SimpleNeuralNetwork network = new SimpleNeuralNetwork();

double[] inputs = {0.5, 0.8}; // Пример входных данных

double output = network.feedForward(inputs);

System.out.println("Output of the neural network: " + output);

}

}



(Молокин Иван Сергеевич) 21. Разработать программу для игры «Угадайка». Программа загадывает случайное число от 1 до 10, требуется его отгадать с трех попыток. После каждой попытки, если результат неверен, игроку выводится сообщение, меньше или больше названное игроком число, чем загаданное. Сет заканчивается или если игрок угадывает число, или если исчерпывает три попытки, не угадав.

Игра должна быть выполнена в бесконечном цикле, и продолжается до тех пор, пока на

предложение «Сыграем еще раз?» игрок не напишет «Нет».

import java.util.Random;

import java.util.Scanner;

public class GuessGame {

public static void main(String[] args) {

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

Random random = new Random();

boolean playAgain = true;

while (playAgain) {

int numberToGuess = random.nextInt(10) + 1;

boolean guessed = false;

System.*out*.println("Я загадал число от 1 до 10. Угадай-ка!");

for (int i = 0; i < 3; i++) {

System.*out*.println("Попытка " + (i + 1) + ": ");

int guess = scanner.nextInt();

if (guess == numberToGuess) {

System.*out*.println("Поздравляю! Вы угадали число и победили!");

guessed = true;

break;

} else if (guess < numberToGuess) {

System.*out*.println("Загаданное число больше!");

} else {

System.*out*.println("Загаданное число меньше!");

}

}

if (!guessed) {

System.*out*.println("Вы не угадали число. Загаданное число было: " + numberToGuess);

}

System.*out*.print("Сыграем еще раз? (Да/Нет): ");

scanner.nextLine();

String response = scanner.nextLine().trim();

if (response.equalsIgnoreCase("Нет")) {

playAgain = false;

System.*out*.println("Спасибо за игру!");

}

}

}

}



(Молокин Иван Сергеевич) 36. Разработать программу для поочередной обработки текстовых файлов. Файлы созданы со следующими именами: n.txt, где n – натуральное число. В файлах записаны: в первой строке одно число с плавающей запятой, во второй строке – второе число. Пользователь вводит название файла и требуемую операцию над числами (сложение, умножение, разность). Результат выводится на экран и файл n\_out.txt.

import java.io.\*;

import java.util.Scanner;

public class FileCalculator {

public static void main(String[] args) {

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

System.*out*.println("Введите название файла (например, 1.txt): ");

String inputFilename = scanner.nextLine();

System.*out*.println("Выберите операцию (сложение, умножение, разность): ");

String operation = scanner.nextLine();

try {

BufferedReader reader = new BufferedReader(new FileReader(inputFilename));

double number1 = Double.*parseDouble*(reader.readLine());

double number2 = Double.*parseDouble*(reader.readLine());

reader.close();

double result;

switch (operation) {

case "сложение":

result = number1 + number2;

break;

case "умножение":

result = number1 \* number2;

break;

case "разность":

result = number1 - number2;

break;

default:

System.*out*.println("Некорректная операция. Попробуйте снова.");

return;

}

System.*out*.println("Результат операции " + '"' + operation + '"' + ": " + result);

String outputFileName = inputFilename.replace(".txt", "\_out.txt");

BufferedWriter writer = new BufferedWriter(new FileWriter(outputFileName, true));

writer.write("Результат операции " + '"' + operation + '"' + ": " + result);

writer.newLine();

writer.close();

System.*out*.println("Результат сохранен в файл: " + outputFileName);

} catch (FileNotFoundException e) {

System.*out*.println("Файл не найден. Проверьте имя файла и попробуйте снова.");

} catch (IOException e) {

System.*out*.println("Произошла ошибка при чтении или записи файла.");

} catch (NumberFormatException e) {

System.*out*.println("Ошибка формата чисел в файле.");

}

}

}



(Ногин Никита Андреевич) 32. Напишите программу, которая получает в качестве входных данных два числа. Эти

числа являются количество строк и столбцов двумерной коллекции целых чисел. Далее элементы

заполняются случайными числами и выводятся в консоль в виде таблицы.

import java.util.Random;

import java.util.Scanner;

public class RandomMatrix {

public static void main(String[] args) {

// Создаем объект Scanner для считывания данных с клавиатуры

Scanner scanner = new Scanner(System.in);

// Получаем количество строк и столбцов от пользователя

System.out.print("Введите количество строк: ");

int rows = scanner.nextInt();

System.out.print("Введите количество столбцов: ");

int cols = scanner.nextInt();

// Создаем двумерный массив для хранения целых чисел

int[][] matrix = new int[rows][cols];

// Заполняем массив случайными числами

Random random = new Random();

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

matrix[i][j] = random.nextInt(100); // Генерируем случайное число от 0 до 99

}

}

// Выводим массив в виде таблицы

System.out.println("Сгенерированная матрица:");

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

System.out.print(matrix[i][j] + "\t"); // Выводим элемент с табуляцией

}

System.out.println(); // Переход на новую строку после вывода каждого ряда

}

// Закрываем Scanner

scanner.close();

}

}



(Ногин Никита Андреевич) 38. Разработать программу шифровки-дешифровки по алгоритму AES-128. Данные

берутся из файла, зашифрованные данные сохраняются в указанный файл.

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import java.io.File;

import java.io.FileOutputStream;

import java.nio.file.Files;

import java.nio.charset.StandardCharsets;

import java.util.Base64;

public class AESCrypto {

private static final String ALGORITHM = "AES";

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

// Генерация ключа для AES

KeyGenerator keyGenerator = KeyGenerator.getInstance(ALGORITHM);

keyGenerator.init(128); // Длина ключа

SecretKey secretKey = keyGenerator.generateKey();

// Чтение данных из файла с указанием кодировки

byte[] inputBytes = Files.readAllBytes(new File("Input.txt").toPath());

String inputData = new String(inputBytes, StandardCharsets.UTF\_8);

System.out.println("Исходные данные: " + inputData);

// Шифрование данных

Cipher cipher = Cipher.getInstance(ALGORITHM);

cipher.init(Cipher.ENCRYPT\_MODE, secretKey);

byte[] encryptedBytes = cipher.doFinal(inputBytes);

// Кодирование зашифрованных данных в Base64 для удобного отображения

String encryptedBase64 = Base64.getEncoder().encodeToString(encryptedBytes);

System.out.println("Зашифрованные данные (Base64): " + encryptedBase64);

// Сохранение зашифрованных данных в файл

try (FileOutputStream fos = new FileOutputStream("EncryptedOutput.txt")) {

fos.write(encryptedBytes);

}

// Дешифрование данных

cipher.init(Cipher.DECRYPT\_MODE, secretKey);

byte[] decryptedBytes = cipher.doFinal(encryptedBytes);

// Отображение расшифрованных данных

String decryptedData = new String(decryptedBytes, StandardCharsets.UTF\_8);

System.out.println("Расшифрованные данные: " + decryptedData);

// Сохранение расшифрованных данных в файл

try (FileOutputStream fos = new FileOutputStream("DecryptedOutput.txt")) {

fos.write(decryptedBytes);

}

System.out.println("Шифрование и дешифрование завершены.");

}

}



**(Нуритдинов Амаль Тимурович) 17. Разработать интерфейс InArray, в котором предусмотреть метод сложения двух массивов. Создать класс ArraySum, в котором имплементируется метод сложения массивов.Создать класс OrArray, в котором метод сложения массивов имплементируется как логическая операция ИЛИ между элементами массива.**

interface InArray {

int[] sum(int[] array1, int[] array2);

}

class ArraySum implements InArray {

@Override

public int[] sum(int[] array1, int[] array2) {

// Определяем размер результирующего массива

int length = Math.min(array1.length, array2.length);

int[] result = new int[length];

// Складываем элементы массивов

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

result[i] = array1[i] + array2[i];

}

return result;

}

}

class OrArray implements InArray {

@Override

public int[] sum(int[] array1, int[] array2) {

// Определяем размер результирующего массива

int length = Math.min(array1.length, array2.length);

int[] result = new int[length];

// Выполняем логическую операцию ИЛИ

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

result[i] = (array1[i] | array2[i]); // Логическая операция ИЛИ

}

return result;

}

}

public class Main {

public static void main(String[] args) {

int[] array1 = {1, 0, 1, 0};

int[] array2 = {0, 1, 0, 1};

InArray arraySum = new ArraySum();

InArray orArray = new OrArray();

// Сложение массивов

int[] sumResult = arraySum.sum(array1, array2);

System.out.println("Сумма массивов:");

for (int num : sumResult) {

System.out.print(num + " ");

}

System.out.println();

// Логическая операция ИЛИ

int[] orResult = orArray.sum(array1, array2);

System.out.println("Логическое ИЛИ массивов:");

for (int num : orResult) {

System.out.print(num + " ");

}

}

}



**(Нуритдинов Амаль Тимурович) 34. Разработайте программу, в которой создайте две коллекции с именами людей (строковые переменные). Результат сохранить в MySQL/PostgreSQL. Затем последовательно выводите в консоль имена.**

import java.sql.\*;

import java.util.List;

public class NameStorage {

public static void main(String[] args) {

// Две коллекции с именами

List<String> collection1 = List.of("Alice", "Bob", "Charlie");

List<String> collection2 = List.of("David", "Eve", "Frank");

// Данные для подключения к MySQL

String url = "jdbc:mysql://localhost:3306/test";

String user = "root"; // Имя пользователя базы данных

String password = "root"; // Пароль базы данных

// SQL-запросы

String createTableQuery = "CREATE TABLE IF NOT EXISTS names (id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(255) NOT NULL)";

String insertQuery = "INSERT INTO names (name) VALUES (?)";

String selectQuery = "SELECT name FROM names";

try (Connection connection = DriverManager.getConnection(url, user, password);

Statement statement = connection.createStatement()) {

// Создание таблицы, если её нет

statement.execute(createTableQuery);

// Вставка данных из коллекций

try (PreparedStatement preparedStatement = connection.prepareStatement(insertQuery)) {

for (String name : collection1) {

preparedStatement.setString(1, name);

preparedStatement.executeUpdate();

}

for (String name : collection2) {

preparedStatement.setString(1, name);

preparedStatement.executeUpdate();

}

}

// Чтение данных из таблицы

try (ResultSet resultSet = statement.executeQuery(selectQuery)) {

System.out.println("Сохранённые имена:");

while (resultSet.next()) {

System.out.println(resultSet.getString("name"));

}

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}



(Панасян Роман Романович) 5. Напишите программную реализацию бинарного дерева поиска.

class Node {

int value;

Node left;

Node right;

public Node(int value) {

this.value = value;

this.left = null;

this.right = null;

}

}

class BinarySearchTree {

private Node root;

public BinarySearchTree() {

this.root = null;

}

// Вставка нового узла

public void insert(int value) {

root = insertRec(root, value);

}

private Node insertRec(Node root, int value) {

if (root == null) {

root = new Node(value);

return root;

}

if (value < root.value) {

root.left = insertRec(root.left, value);

} else if (value > root.value) {

root.right = insertRec(root.right, value);

}

return root;

}

// Поиск узла

public boolean search(int value) {

return searchRec(root, value);

}

private boolean searchRec(Node root, int value) {

if (root == null) {

return false;

}

if (value == root.value) {

return true;

}

return value < root.value ? searchRec(root.left, value) : searchRec(root.right, value);

}

// Удаление узла

public void delete(int value) {

root = deleteRec(root, value);

}

private Node deleteRec(Node root, int value) {

if (root == null) {

return root;

}

if (value < root.value) {

root.left = deleteRec(root.left, value);

} else if (value > root.value) {

root.right = deleteRec(root.right, value);

} else {

// Узел с одним дочерним узлом или без дочерних узлов

if (root.left == null) return root.right;

else if (root.right == null) return root.left;

// Узел с двумя дочерними узлами: получить минимальное значение из правого поддерева

root.value = minValue(root.right);

// Удалить минимальное значение из правого поддерева

root.right = deleteRec(root.right, root.value);

}

return root;

}

private int minValue(Node root) {

int minValue = root.value;

while (root.left != null) {

minValue = root.left.value;

root = root.left;

}

return minValue;

}

// Инфиксный обход дерева

public void inorder() {

inorderRec(root);

}

private void inorderRec(Node root) {

if (root != null) {

inorderRec(root.left);

System.*out*.print(root.value + " ");

inorderRec(root.right);

}

}

}

public class binary\_tree {

public static void main(String[] args) {

BinarySearchTree bst = new BinarySearchTree();

bst.insert(70);

bst.insert(60);

bst.insert(80);

bst.insert(50);

bst.insert(30);

bst.insert(20);

bst.insert(40);

System.*out*.println("Инфиксный обход:");

bst.inorder(); // Вывод: 20 30 40 50 60 70 80

System.*out*.println("\nПоиск 40: " + bst.search(40)); // Вывод: true

System.*out*.println("Поиск 90: " + bst.search(90)); // Вывод: false

bst.delete(20);

System.*out*.println("Инфиксный обход после удаления 20:");

bst.inorder(); // Вывод: 30 40 50 60 70 80

}

}



(Панасян Роман Романович) 35. Напишите программу, которая реализует класс Matrix и следующие методы:

a. Сложение и вычитание матриц.

b. Умножение матрицы на число.

c. Произведение двух матриц.

d. Транспонированная матрица.

e. Возведение матрицы в степень.

f. Если метод, возвращает матрицу, то он должен возвращать новый объект, а не менять

базовый.

class Matrix {

private final double[][] data;

private final int rows;

private final int cols;

// Конструктор класса Matrix

public Matrix(double[][] data) {

this.rows = data.length;

this.cols = data[0].length;

this.data = new double[rows][cols];

for (int i = 0; i < rows; i++) {

System.*arraycopy*(data[i], 0, this.data[i], 0, cols);

}

}

// Метод для сложения матриц

public Matrix add(Matrix other) {

if (this.rows != other.rows || this.cols != other.cols) {

throw new IllegalArgumentException("Матрицы должны иметь одинаковые размеры для сложения.");

}

double[][] result = new double[this.rows][this.cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = this.data[i][j] + other.data[i][j];

}

}

return new Matrix(result);

}

// Метод для вычитания матриц

public Matrix subtract(Matrix other) {

if (this.rows != other.rows || this.cols != other.cols) {

throw new IllegalArgumentException("Матрицы должны иметь одинаковые размеры для вычитания.");

}

double[][] result = new double[this.rows][this.cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = this.data[i][j] - other.data[i][j];

}

}

return new Matrix(result);

}

// Метод для умножения матрицы на число

public Matrix multiplyByScalar(double scalar) {

double[][] result = new double[this.rows][this.cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[i][j] = this.data[i][j] \* scalar;

}

}

return new Matrix(result);

}

// Метод для произведения двух матриц

public Matrix multiply(Matrix other) {

if (this.cols != other.rows) {

throw new IllegalArgumentException("Число столбцов первой матрицы должно совпадать с числом строк второй матрицы.");

}

double[][] result = new double[this.rows][other.cols];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < other.cols; j++) {

for (int k = 0; k < this.cols; k++) {

result[i][j] += this.data[i][k] \* other.data[k][j];

}

}

}

return new Matrix(result);

}

// Метод для транспонирования матрицы

public Matrix transpose() {

double[][] result = new double[this.cols][this.rows];

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

result[j][i] = this.data[i][j];

}

}

return new Matrix(result);

}

// Метод для возведения матрицы в степень

public Matrix power(int exponent) {

if (rows != cols) {

throw new IllegalArgumentException("Матрица должна быть квадратной для возведения в степень.");

}

if (exponent == 0) {

double[][] identityMatrix = new double[rows][cols];

for (int i = 0; i < rows; i++) {

identityMatrix[i][i] = 1;

}

return new Matrix(identityMatrix);

}

Matrix result = this;

for (int i = 1; i < exponent; i++) {

result = result.multiply(this);

}

return result;

}

// Метод для вывода матрицы

@Override

public String toString() {

StringBuilder sb = new StringBuilder();

for (double[] row : data) {

for (double value : row) {

sb.append(value).append(" ");

}

sb.append("\n");

}

return sb.toString();

}

// Основной метод для тестирования класса

public static void main(String[] args) {

double[][] data1 = {{1, 2}, {3, 4}};

double[][] data2 = {{5, 6}, {7, 8}};

Matrix matrix1 = new Matrix(data1);

Matrix matrix2 = new Matrix(data2);

System.*out*.println("Matrix 1:\n" + matrix1);

System.*out*.println("Matrix 2:\n" + matrix2);

System.*out*.println("Сложение:\n" + matrix1.add(matrix2));

System.*out*.println("Вычитание:\n" + matrix1.subtract(matrix2));

System.*out*.println("Умножение на число:\n" + matrix1.multiplyByScalar(2));

System.*out*.println("Произведение двух матриц:\n" + matrix1.multiply(matrix2));

System.*out*.println("Транспонированная матрица:\n" + matrix1.transpose());

System.*out*.println("Возведение в степень:\n" + matrix1.power(2));

}

}



(Скрябин Айсен Максимович) 10. Написать класс, который при введении даты в формате ДД.ММ.ГГ (к примеру,

22.10.20) выводит номер недели. Даты начиная с 2020 по 2022 годы. К примеру, первая неделя в

2020 году: 1-5 января, вторая неделя – 6-12 января. Значит при вводе 08.01.20 вывод должен быть:

Неделя 2.

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.time.temporal.WeekFields;

import java.util.Locale;

import java.util.Scanner;

public class WeekNumberCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите дату в формате ДД.ММ.ГГ (например, 22.10.20): ");

String inputDate = scanner.nextLine();

// Определяем формат даты

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd.MM.yy");

try {

// Парсим введенную дату

LocalDate date = LocalDate.parse(inputDate, formatter);

// Проверяем, что дата находится в диапазоне 2020-2022

if (date.getYear() < 2020 || date.getYear() > 2022) {

System.out.println("Дата должна быть в диапазоне с 2020 по 2022 год.");

return;

}

// Определяем номер недели

WeekFields weekFields = WeekFields.of(Locale.getDefault());

int weekNumber = date.get(weekFields.weekOfWeekBasedYear());

// Выводим номер недели

System.out.println("Неделя " + weekNumber);

} catch (Exception e) {

System.out.println("Ошибка при вводе даты. Убедитесь, что дата введена в правильном формате.");

}

}

}



(Скрябин Айсен Максимович) 19. Создать класс Matrix для работы с двумерными матрицами. Создать методы для

генерации нулевой матрицы, а также для генерации матрицы со случайными величинами –

применить Math.random(). Реализовать метод сложения матриц.

import java.util.Random;

public class Matrix {

private int[][] data;

private int rows;

private int cols;

// Конструктор для создания матрицы заданного размера

public Matrix(int rows, int cols) {

this.rows = rows;

this.cols = cols;

this.data = new int[rows][cols];

}

// Метод для генерации нулевой матрицы

public static Matrix generateZeroMatrix(int rows, int cols) {

return new Matrix(rows, cols);

}

// Метод для генерации матрицы со случайными величинами

public static Matrix generateRandomMatrix(int rows, int cols) {

Matrix matrix = new Matrix(rows, cols);

Random random = new Random();

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

matrix.data[i][j] = random.nextInt(100); // Случайные числа от 0 до 99

}

}

return matrix;

}

// Метод для сложения двух матриц

public Matrix add(Matrix other) {

if (this.rows != other.rows || this.cols != other.cols) {

throw new IllegalArgumentException("Матрицы должны иметь одинаковые размеры для сложения.");

}

Matrix result = new Matrix(this.rows, this.cols);

for (int i = 0; i < this.rows; i++) {

for (int j = 0; j < this.cols; j++) {

result.data[i][j] = this.data[i][j] + other.data[i][j];

}

}

return result;

}

// Метод для вывода матрицы

public void print() {

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

System.out.print(data[i][j] + " ");

}

System.out.println();

}

}

// Пример использования

public static void main(String[] args) {

Matrix zeroMatrix = Matrix.generateZeroMatrix(3, 3);

System.out.println("Нулевая матрица:");

zeroMatrix.print();

Matrix randomMatrix1 = Matrix.generateRandomMatrix(3, 3);

System.out.println("Случайная матрица 1:");

randomMatrix1.print();

Matrix randomMatrix2 = Matrix.generateRandomMatrix(3, 3);

System.out.println("Случайная матрица 2:");

randomMatrix2.print();

Matrix sumMatrix = randomMatrix1.add(randomMatrix2);

System.out.println("Сумма матриц:");

sumMatrix.print();

}

}



(Староверова Анна Андреевна) 37. Написать приложение для сложения, вычитания, умножения, деления, возведения в степень логарифмов. Программа должна выполнять ввод данных, проверку правильности введенных данных, выдачу сообщений в случае ошибок. Результат выводится на экран и записывается в файл.

public class Logarithm {

private double base;

private double number;

private double result;

public Logarithm(double base, double number) {

this.base = base;

this.number = number;

this.result = result();

}

private double result() {

return Math.log(this.number) / Math.log(this.base);

}

public double getBase() {

return base;

}

public double getNumber() {

return number;

}

public double getResult() {

return result;

}

public void setBase(double base) {

this.base = base;

this.result = result();

}

public void setNumber(double number) {

this.number = number;

this.result = result();

}

public String logToString() {

try {

int b = (int) this.base;

int n = (int) this.number;

return "log" + b + "(" + n + ")";

} catch (Exception e) {}

return "log" + this.base + "(" + this.number + ")";

}

}



public class Calculator {

public static boolean logCheck(Logarithm log1, Logarithm log2) {

if (log1.getBase() != log2.getBase()) {

System.out.println("Основания логарифмов должны быть одинаковыми\n");

return false;

}

return true;

}

public static double addition(Logarithm log1, Logarithm log2) {

return log1.getResult() + log2.getResult();

}

public static double subtraction(Logarithm log1, Logarithm log2) {

return log1.getResult() - log2.getResult();

}

public static double multiplication(Logarithm log1, Logarithm log2) {

return log1.getResult()\*log2.getResult();

}

public static double division(Logarithm log1, Logarithm log2) {

return log1.getResult()/log2.getResult();

}

public static double power(Logarithm log1, double power) {

return log1.getResult()\*power;

}

}



import java.util.Scanner;

public class logInput {

public static Logarithm[] logInput1() {

double base1 = logInput.logInput("основание", " первого");

double number1 = logInput.logInput("число", " первого");

double base2 = logInput.logInput("основание", " второго");

double number2 = logInput.logInput("число", " второго");

Logarithm log1 = new Logarithm(base1, number1);

Logarithm log2 = new Logarithm(base2, number2);

Logarithm[] logs = new Logarithm[2];

logs[0] = log1;

logs[1] = log2;

return logs;

}

public static Logarithm logInput2() {

double base1 = logInput.logInput("основание", "");

double number1 = logInput.logInput("число", "");

Logarithm log1 = new Logarithm(base1, number1);

return log1;

}

public static double logInput(String param, String num) {

Scanner sc = new Scanner(System.in);

System.out.print("Введите " + param + num + " логарифма: ");

String p = sc.nextLine();

while (!doubleCheck(p)) {

System.out.println("Неверный формат ввода");

System.out.print("Введите " + param + " " + num + " логарифма: ");

p = sc.nextLine();

}

return Double.parseDouble(p);

}

public static boolean doubleCheck(String str) {

try {

Double.parseDouble(str);

return true;

} catch (NumberFormatException e) {}

return false;

}

public static boolean intCheck(String str) {

try {

Integer.parseInt(str);

return true;

} catch (NumberFormatException e) {}

return false;

}

public static boolean actionCheck(String str) {

if (intCheck(str)) {

int x = Integer.parseInt(str);

if (x>=1 && x<=6) {

return true;

}

}

return false;

}

}



import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String s = "";

try (FileWriter fw = new FileWriter("logarithms.txt", true)) {

while (!s.equals("6")) {

System.out.println("1. Сложение логарифмов");

System.out.println("2. Вычитание логарифмов");

System.out.println("3. Умножение логарифмов");

System.out.println("4. Деление логарифмов");

System.out.println("5. Возведение в степень логарифмов");

System.out.println("6. Завершение работы");

System.out.print("\nВыберите действие: ");

s = scanner.nextLine();

while (!logInput.actionCheck(s)) {

System.out.println("Неверный формат ввода\n");

System.out.print("Выберите действие: ");

s = scanner.nextLine();

}

System.out.println();

int x = Integer.parseInt(s);

switch (x) {

case 1 -> {

Logarithm[] logs = logInput.logInput1();

while (!Calculator.logCheck(logs[0], logs[1])) {

logs = logInput.logInput1();

}

double res = Calculator.addition(logs[0], logs[1]);

String result = logs[0].logToString() + " + " + logs[1].logToString() + " = " + res;

fw.write(result + "\n");

System.out.println("\n" + result);

}

case 2 -> {

Logarithm[] logs = logInput.logInput1();

while (!Calculator.logCheck(logs[0], logs[1])) {

logs = logInput.logInput1();

}

double res = Calculator.subtraction(logs[0], logs[1]);

String result = logs[0].logToString() + " - " + logs[1].logToString() + " = " + res;

fw.write(result + "\n");

System.out.println("\n" + result);

}

case 3 -> {

Logarithm[] logs = logInput.logInput1();

double res = Calculator.multiplication(logs[0], logs[1]);

String result = logs[0].logToString() + " \* " + logs[1].logToString() + " = " + res;

fw.write(result + "\n");

System.out.println("\n" + result);

}

case 4 -> {

Logarithm[] logs = logInput.logInput1();

double res = Calculator.division(logs[0], logs[1]);

String result = logs[0].logToString() + " / " + logs[1].logToString() + " = " + res;

fw.write(result + "\n");

System.out.println("\n" + result);

}

case 5 -> {

Logarithm log1 = logInput.logInput2();

System.out.print("Введите степень: ");

String p = scanner.nextLine();

while (!logInput.intCheck(p)) {

System.out.println("Неверный формат ввода");

System.out.print("Введите степень: ");

p = scanner.nextLine();

}

int power = Integer.parseInt(p);

double res = Calculator.power(log1, power);

String result = log1.logToString() + "^" + power + " = " + res;

fw.write(result + "\n");

System.out.println("\n" + result);

}

}

System.out.println();

}

fw.close();

} catch (IOException e) {

System.out.println("Ошибка записи в файл");

}

}

}



(Староверова Анна Андреевна) 2. Напишите программу, в которой из строки "I have 3 cats, 4 dogs, and 1 turtle" отбираются цифры. Из этих цифр формируется массив.

public class Main {

public static void main(String[] args) {

String str = "I have 3 cats, 4 dogs, and 1 turtle";

StringBuilder sb = new StringBuilder();

for (char c : str.toCharArray()) {

if (Character.isDigit(c)) {

sb.append(c).append(" ");

}

}

String[] strDigits = sb.toString().trim().split(" ");

int[] digits = new int[strDigits.length];

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

digits[i] = Integer.parseInt(strDigits[i]);

}

System.out.println(Arrays.toString(digits));

}

}



(Сырейщиков Артемий Евгеньевич) 20. Реализовать класс MyMath для работы с числами. Реализовать статический метод

класса MyMath.round(), который округляет дробь до целого числа. Также статический метод

abs(), который находит модуль числа. Статический метод MyMath.pow() для нахождения степени

числа. Библиотеку Math не использовать.

public class MyMath {

// Метод для округления числа до ближайшего целого

public static int round(double value) {

int intPart = (int) value; // Получаем целую часть числа

double fractionalPart = value - intPart; // Вычисляем дробную часть

if (fractionalPart >= 0.5) {

return intPart + 1; // Если дробная часть >= 0.5, округляем вверх

} else if (fractionalPart <= -0.5) {

return intPart - 1; // Если дробная часть <= -0.5, округляем вниз

} else {

return intPart; // Иначе оставляем целую часть без изменений

}

}

// Метод для нахождения модуля числа

public static double abs(double value) {

return value < 0 ? -value : value; // Если число отрицательное, возвращаем его противоположность

}

// Метод для возведения числа в степень

public static double pow(double base, int exponent) {

if (exponent == 0) {

return 1; // Любое число в степени 0 равно 1

}

double result = 1;

boolean isNegativeExponent = exponent < 0; // Проверяем, отрицательная ли степень

if (isNegativeExponent) {

exponent = -exponent; // Делаем степень положительной для расчета

}

for (int i = 0; i < exponent; i++) {

result \*= base; // Умножаем число на себя столько раз, сколько указано в степени

}

return isNegativeExponent ? 1 / result : result; // Если степень была отрицательной, возвращаем обратное значение

}

// Пример тестирования методов

public static void main(String[] args) {

// Тестируем метод round()

System.out.println("Round examples:");

System.out.println("round(4.5) = " + MyMath.round(4.5)); // Ожидаем 5

System.out.println("round(4.4) = " + MyMath.round(4.4)); // Ожидаем 4

System.out.println("round(-4.5) = " + MyMath.round(-4.5)); // Ожидаем -5

System.out.println("round(-4.4) = " + MyMath.round(-4.4)); // Ожидаем -4

// Тестируем метод abs()

System.out.println("\nAbs examples:");

System.out.println("abs(5) = " + MyMath.abs(5)); // Ожидаем 5

System.out.println("abs(-5) = " + MyMath.abs(-5)); // Ожидаем 5

System.out.println("abs(0) = " + MyMath.abs(0)); // Ожидаем 0

// Тестируем метод pow()

System.out.println("\nPow examples:");

System.out.println("pow(2, 3) = " + MyMath.pow(2, 3)); // Ожидаем 8

System.out.println("pow(2, -3) = " + MyMath.pow(2, -3)); // Ожидаем 0.125

System.out.println("pow(5, 0) = " + MyMath.pow(5, 0)); // Ожидаем 1

System.out.println("pow(-2, 3) = " + MyMath.pow(-2, 3)); // Ожидаем -8

}

}



(Сырейщиков Артемий Евгеньевич) 46. Разработать приложение для работы с локальной базой данных MySQL. Создайте базу

данных мобильных телефонов (не менее 10 позиций), со следующими полями: производитель,

модель, год выпуска, диагональ экрана. Напишите методы для выполнения запросов к базе

данных. Все данные должны выводиться в консоли на экран.





(Чесноков Тимофей Евгеньевич) 25. Сделайте класс User, в котором будут следующие protected поля - name (имя), age

(возраст), public методы setName, getName, setAge, getAge. Сделайте класс Worker, который

наследует от класса User и вносит дополнительное private поле salary (зарплата), а также методы

public getSalary и setSalary. Создайте объект этого класса 'Иван', возраст 25, зарплата 1000.

Создайте второй объект этого класса 'Вася', возраст 26, зарплата 2000. Найдите сумму зарплата

Ивана и Васи. Сделайте класс Student, который наследует от класса User и вносит

дополнительные private поля стипендия, курс, а также геттеры и сеттеры для них.

// Класс User

class User {

protected String name; // Имя

protected int age; // Возраст

// Метод для установки имени

public void setName(String name) {

this.name = name;

}

// Метод для получения имени

public String getName() {

return name;

}

// Метод для установки возраста

public void setAge(int age) {

this.age = age;

}

// Метод для получения возраста

public int getAge() {

return age;

}

}

// Класс Worker, наследующий от User

class Worker extends User {

private double salary; // Зарплата

// Метод для установки зарплаты

public void setSalary(double salary) {

this.salary = salary;

}

// Метод для получения зарплаты

public double getSalary() {

return salary;

}

}

// Класс Student, наследующий от User

class Student extends User {

private double scholarship; // Стипендия

private int course; // Курс

// Метод для установки стипендии

public void setScholarship(double scholarship) {

this.scholarship = scholarship;

}

// Метод для получения стипендии

public double getScholarship() {

return scholarship;

}

// Метод для установки курса

public void setCourse(int course) {

this.course = course;

}

// Метод для получения курса

public int getCourse() {

return course;

}

}

// Основной класс для тестирования

public class Main {

public static void main(String[] args) {

// Создание объектов Worker

Worker ivan = new Worker();

ivan.setName("Иван");

ivan.setAge(25);

ivan.setSalary(1000);

Worker vasya = new Worker();

vasya.setName("Вася");

vasya.setAge(26);

vasya.setSalary(2000);

// Вычисление суммы зарплат Иван и Васи

double totalSalary = ivan.getSalary() + vasya.getSalary();

System.out.println("Сумма зарплат Ивана и Васи: " + totalSalary);

// Пример создания объекта Student (по желанию)

Student student = new Student();

student.setName("Алексей");

student.setAge(20);

student.setScholarship(500);

student.setCourse(2);

System.out.println("Студент: " + student.getName() + ", Возраст: " + student.getAge() +

", Стипендия: " + student.getScholarship() + ", Курс: " + student.getCourse());

}

}



(Чесноков Тимофей Евгеньевич) 29. Напишите программу, которая заполняет списочный массив случайными числами

типа Integer (значения этих чисел были от 1 до 100). Список должен содержать 100 элементов.

Затем отсортируйте по убыванию список и выведите первые 10 значений в консоль. Результатыы

сохраните в MySQL/PostgreSQL.

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Random;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class RandomNumbers {

public static void main(String[] args) {

// Создаем список для хранения случайных чисел

List<Integer> numbers = new ArrayList<>();

Random random = new Random();

// Заполняем список 100 случайными числами от 1 до 100

for (int i = 0; i < 100; i++) {

numbers.add(random.nextInt(100) + 1);

}

// Сортируем список по убыванию

Collections.sort(numbers, Collections.reverseOrder());

// Выводим первые 10 значений

System.out.println("Первые 10 значений:");

for (int i = 0; i < 10; i++) {

System.out.println(numbers.get(i));

}

// Сохраняем результаты в MySQL

saveToDatabase(numbers);

}

private static void saveToDatabase(List<Integer> numbers) {

String url = "jdbc:mysql://localhost:3306/your\_database"; // Замените на ваш URL базы данных

String user = "your\_username"; // Замените на ваше имя пользователя

String password = "your\_password"; // Замените на ваш пароль

String query = "INSERT INTO random\_numbers (number) VALUES (?)";

try (Connection connection = DriverManager.getConnection(url, user, password);

PreparedStatement preparedStatement = connection.prepareStatement(query)) {

for (Integer number : numbers) {

preparedStatement.setInt(1, number);

preparedStatement.executeUpdate();

}

System.out.println("Числа успешно сохранены в базе данных.");

} catch (SQLException e) {

e.printStackTrace();

}

}

}



(Шелудкин Павел Дмитриевич) 9. Напишите программу, которая по дате определяет день недели, на который эта дата

приходится.

import java.time.LocalDate;

import java.time.format.DateTimeFormatter;

import java.time.format.DateTimeParseException;

import java.util.Scanner;

public class DayOfWeekCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Сообщаем пользователю формат ввода

System.out.println("Введите дату в формате YYYY-MM-DD: ");

// Читаем строку из консоли

String inputDate = scanner.nextLine();

try {

// Парсим введённую строку в объект LocalDate

LocalDate date = LocalDate.parse(inputDate, DateTimeFormatter.ISO\_LOCAL\_DATE);

// Получаем день недели

String dayOfWeek = date.getDayOfWeek().toString();

// Выводим результат

System.out.println("Этот день недели: " + dayOfWeek);

} catch (DateTimeParseException e) {

// Обрабатываем ошибку неверного формата ввода

System.out.println("Некорректный формат даты. Попробуйте снова.");

}

scanner.close();

}

}



(Шелудкин Павел Дмитриевич) 40. Напишите программу, которая каждые 5 секунд отображает на экране данные о

времени, прошедшем от начала запуска программы, а другой её поток выводит сообщение

каждые 7 секунд. Третий поток выводит на экран сообщение каждые 10 секунд. Программа

работает одну минуту, затем останавливается. Все результаты после вывода необходимо

сохранить в MySQL/PostgreSQL.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.time.Duration;

import java.time.Instant;

public class MultiThreadedTimer {

public static void main(String[] args) {

Instant start = Instant.now();

// Поток для каждых 5 секунд

Thread task5Seconds = new Thread(() -> runTask(5, "5 секунд прошли", start));

// Поток для каждых 7 секунд

Thread task7Seconds = new Thread(() -> runTask(7, "7 секунд прошли", start));

// Поток для каждых 10 секунд

Thread task10Seconds = new Thread(() -> runTask(10, "10 секунд прошли", start));

// Запускаем потоки

task5Seconds.start();

task7Seconds.start();

task10Seconds.start();

// Основной поток останавливается через 1 минуту

try {

Thread.sleep(60\_000);

} catch (InterruptedException e) {

e.printStackTrace();

}

// Прерываем потоки после завершения 1 минуты

task5Seconds.interrupt();

task7Seconds.interrupt();

task10Seconds.interrupt();

System.out.println("Программа завершена.");

}

/\*\*

\* Выполнение задачи с указанным интервалом.

\*

\* @param intervalSeconds Интервал выполнения в секундах.

\* @param message Сообщение для вывода.

\* @param start Начальное время запуска программы.

\*/

private static void runTask(int intervalSeconds, String message, Instant start) {

while (!Thread.currentThread().isInterrupted()) {

try {

// Выводим сообщение

long elapsedSeconds = Duration.between(start, Instant.now()).getSeconds();

System.out.println(message + " (Прошло времени: " + elapsedSeconds + " сек)");

saveToDatabase(elapsedSeconds, message);

// Ждём следующий интервал

Thread.sleep(intervalSeconds \* 1000);

} catch (InterruptedException e) {

// Завершаем поток при прерывании

Thread.currentThread().interrupt();

}

}

}

/\*\*

\* Сохранение данных в базу данных MySQL.

\*

\* @param elapsedSeconds Время в секундах от запуска программы.

\* @param message Сообщение.

\*/

private static void saveToDatabase(long elapsedSeconds, String message) {

String url = "jdbc:mysql://localhost:3306/testdb"; // Замените на ваш URL базы данных

String user = "root"; // Замените на ваш логин

String password = "password"; // Замените на ваш пароль

String sql = "INSERT INTO log\_data (elapsed\_time, message) VALUES (?, ?)";

try (Connection connection = DriverManager.getConnection(url, user, password);

PreparedStatement statement = connection.prepareStatement(sql)) {

statement.setLong(1, elapsedSeconds);

statement.setString(2, message);

statement.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

