

TUẦN 06:

SỬ DỤNG METHOD TRONG JAVA

Bài 01: Viết chương trình giải phương trình bậc 2, phương trình bậc 2 có dạng:

$$ax^2 + bx + c = 0$$

Phân tích:

+ Đầu vào: hệ số a, hệ số b, hệ số c

+ Đầu ra: phương trình vô nghiệm, phương trình vô nghiệm, phương trình có nghiệm

+ Xử lý: biện luận và tìm nghiệm của phương trình

Viết code

Bước 01: tạo Java Project tên là **Buoi06**

Bước 02: tạo Package tên là **Bai01**

Bước 03: trong Package Bai01, tạo Class tên là **GiaiPhuongTrinhBacHai** và viết code:

```
import java.util.Scanner;

public class GiaiPhuongTrinhBacHai {

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

    public static void main(String[] args) {
        System.out.print("Nhập hệ số bậc 2, a = ");
        float a = scanner.nextFloat();
        System.out.print("Nhập hệ số bậc 1, b = ");
        float b = scanner.nextFloat();
        System.out.print("Nhập hằng số tự do, c = ");
        float c = scanner.nextFloat();
        giaiPTBac2(a, b, c);
    }

    public static void giaiPTBac2(float a, float b, float c) {
        // kiểm tra các hệ số
        if (a == 0) {
            if (b == 0) {
                System.out.println("Phương trình vô nghiệm!");
            } else {
                System.out.println("Phương trình có một nghiệm: " + "x = " + (-c / b));
            }
            return;
        }
        // tính delta
        float delta = b*b - 4*a*c;
        float x1;
        float x2;
        // tính nghiệm
        if (delta > 0) {
            x1 = (float) ((-b + Math.sqrt(delta)) / (2*a));
            x2 = (float) ((-b - Math.sqrt(delta)) / (2*a));
            System.out.println("Phương trình có 2 nghiệm là: " + "x1 = " + x1 + " và x2 = " + x2);
        } else if (delta == 0) {
            x1 = (-b / (2 * a));
            System.out.println("Phương trình có nghiệm kép: " + "x1 = x2 = " + x1);
        } else {
            System.out.println("Phương trình vô nghiệm!");
        }
    }
}
```

Bài 02: Viết chương trình tính giai thừa của 1 số nguyên được nhập từ bàn phím. giai thừa của 1 số là tích các số liên tiếp từ 1 đến số đó. Trường hợp đặc biệt, giai thừa của 0 và 1 là 1. Ví dụ: giai thừa của 5 là $1*2*3*4*5 = 120$.

2.1/ Tính giai thừa không sử dụng đệ quy

Bước 01: tạo Package tên là **Bai02**

Bước 02: Trong package Bai02, tạo class tên là **GiaiThuaKhongDeQuy** và code:

```
public class GiaiThuaKhongDeQuy {

    public static long tinhGiaiThua(int n) {
        long giai_thua = 1;
        if (n == 0 || n == 1) {
            return giai_thua;
        } else {
            for (int i = 2; i <= n; i++) {
                giai_thua *= i;
            }
            return giai_thua;
        }
    }

    public static void main(String[] args) {
        int a = 5;
        int b = 0;
        int c = 10;

        System.out.println("Giai thừa của " + a + " là: " + tinhGiaiThua(a));
        System.out.println("Giai thừa của " + b + " là: " + tinhGiaiThua(b));
        System.out.println("Giai thừa của " + c + " là: " + tinhGiaiThua(c));
    }
}
```

2.2/ Tính giai thừa sử dụng đệ quy

Trong package Bai02, tạo class tên là **GiaiThuaDeQuy** và code:

```
public class GiaiThuaDeQuy {

    public static long tinhGiaiThua(int n) {
        if (n > 0) {
            return n * tinhGiaiThua(n - 1);
        } else {
            return 1;
        }
    }
}
```

```

    public static void main(String[] args) {
        int a = 5;
        int b = 0;
        int c = 10;

        System.out.println("Giái thừa của " + a + " là: " + tinhGiaithua(a));
        System.out.println("Giái thừa của " + b + " là: " + tinhGiaithua(b));
        System.out.println("Giái thừa của " + c + " là: " + tinhGiaithua(c));
    }
}

```

Bài 03: Viết chương kiểm tra một số có phải là số nguyên tố hay không. số nguyên tố là số lớn hơn 1 và chỉ chia hết cho 1 và chính nó. Ví dụ: 2, 3, 5, 7, 11, 13, 17, ... là những số nguyên tố.

Chú ý: Số 0 và 1 không phải là số nguyên tố. Chỉ có số 2 là số nguyên tố chẵn, tất cả các số chẵn khác không phải là số nguyên tố vì chúng chia hết cho 2.

Hướng dẫn

Bước 01: Tạo Package tên là **Bai03**

Bước 02: trong Package Bai03, tạo Class tên là **KiemTraSoNguyenTo** và code:

```

public class KiemTraSoNguyenTo
{
    public static boolean isPrimeNumber(int n)
    {
        // so nguyen n < 2 không phải là số nguyên tố
        if (n < 2) {
            return false;
        }
        // check số nguyên tố khi n >= 2
        int squareRoot = (int) Math.sqrt(n);
        for (int i = 2; i <= squareRoot; i++)
        {
            if (n % i == 0)
            {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args)
    {
        System.out.println("Các số nguyên tố nhỏ hơn 100 là: ");
        for (int i = 0; i < 100; i++)
        {
            if (isPrimeNumber(i))
            {
                System.out.print(i + " ");
            }
        }
    }
}

```

Bài 04: Viết chương trình tìm n số Fibonacci đầu tiên trong java. Số nguyên dương n được nhập từ bàn phím. Quy luật của dãy số Fibonacci: số tiếp theo bằng tổng của 2 số trước, 2 số đầu tiên của dãy số là 0, 1. Ví dụ: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

4.1/ Không sử dụng đệ quy

Bước 01: Tạo Package tên là **Bai04**

Bước 02: Trong Package Bai04, tạo Class tên là **FibonanciKhongDeQuy** và code:

```
public class FibonanciKhongDeQuy
{
    public static int fibonacci(int n)
    {
        int f0 = 0;
        int f1 = 1;
        int fn = 1;

        if (n < 0) {
            return -1;
        } else if (n == 0 || n == 1) {
            return n;
        } else {
            for (int i = 2; i < n; i++) {
                f0 = f1;
                f1 = fn;
                fn = f0 + f1;
            }
        }
        return fn;
    }

    public static void main(String[] args)
    {
        System.out.println("10 số đầu tiên của dãy số fibonacci: ");
        for (int i = 0; i < 10; i++)
        {
            System.out.print(fibonacci(i) + " ");
        }
    }
}

public static void main(String[] args) {
    System.out.println("10 số đầu tiên của dãy số fibonacci: ");
    for (int i = 0; i < 10; i++) {
        System.out.print(fibonacci(i) + " ");
    }
}
```

4.2/ Sử dụng đệ quy

Bước 01: Trong Package Bai04, tạo Class tên là **FibonacciDeQuy**

Bước 02: Viết code

```
public class FibonanciDeQuy
{
    public static int fibonacci(int n)
    {
        if (n < 0) {
            return -1;
        } else if (n == 0 || n == 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }

    public static void main(String[] args)
    {
        System.out.println("10 số đầu tiên của dãy số fibonacci: ");
        for (int i = 0; i < 10; i++) {
            System.out.print(fibonacci(i) + " ");
        }
    }
}
```

Bài 05: Viết chương trình tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của 2 số nguyên dương a và b nhập từ bàn phím.

- USCLN của 2 số nguyên dương a và b là một số k lớn nhất, sao cho a và b đều chia hết cho k.
- BSCNN của 2 số nguyên dương a và b là một số h nhỏ nhất, sao cho h chia hết cho cả a và b.

Lưu ý: sử dụng giải thuật Euclid để giải quyết bài toán tìm ước số chung lớn nhất (USCLN) và bội số chung nhỏ nhất (BSCNN) của hai số nguyên dương a và b.

Hướng dẫn:

Bước 01: Tạo Package tên là **Bai05**

Bước 02: Trong Package Bai05, tạo class tên là **TimUsclnBscnn** và code:

Sinh viên dùng 1 trong 2 cách sau đây:

Cách 01: dùng đệ quy

```
public static int USCLN(int a, int b) {
    if (b == 0) return a;
    return USCLN(b, a % b);
}
```

Cách 02: không dùng đệ quy

```
public static int USCLN(int a, int b) {
    int temp1 = a;
    int temp2 = b;
    while (temp1 != temp2) {
        if (temp1 > temp2) {
            temp1 -= temp2;
        } else {
            temp2 -= temp1;
        }
    }
    int uscln = temp1;
    return uscln;
}
```

Viết Code:

```
import java.util.Scanner;

public class TimUsclnBscnn
{
    private static Scanner scanner = new Scanner(System.in);

    public static int USCLN(int a, int b)
    {
        if (b == 0) return a;
        return USCLN(b, a % b);
    }

    public static int BSCNN(int a, int b)
    {
        return (a * b) / USCLN(a, b);
    }

    public static void main(String[] args) {
        System.out.print("Nhập số nguyên dương a = ");
        int a = scanner.nextInt();
        System.out.print("Nhập số nguyên dương b = ");
        int b = scanner.nextInt();
        // tính USCLN của a và b
        System.out.println("USCLN của " + a + " và " + b + " là: " + USCLN(a, b));
        // tính BSCNN của a và b
        System.out.println("BSCNN của " + a + " và " + b + " là: " + BSCNN(a, b));
    }
}
```

Bài 06: Viết chương trình liệt kê tất cả các số nguyên tố nhỏ hơn n trong Java. Số nguyên dương n được nhập từ bàn phím.

Hướng dẫn:

Bước 01: Tạo Package tên là Bai06

Bước 02: Trong Package Bai06, tạo class tên là LietKeSoNguyenTo và code:

```
import java.util.Scanner;

public class LietKeSoNguyenTo
{
    private static Scanner scanner = new Scanner(System.in);

    public static boolean isPrimeNumber(int n)
    {
        // so nguyen n < 2 không phải là số nguyên tố
        if (n < 2) {
```

```

        return false;
    }
    // check so nguyen to khi n >= 2
    int squareRoot = (int) Math.sqrt(n);
    for (int i = 2; i <= squareRoot; i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

public static void main(String[] args)
{
    System.out.print("Nhập n = ");
    int n = scanner.nextInt();
    System.out.printf("Tất cả các số nguyên tố nhỏ hơn %d là: \n", n);
    if (n >= 2) {
        System.out.print(2);
    }
    for (int i = 3; i < n; i+=2) {
        if (isPrimeNumber(i)) {
            System.out.print(" " + i);
        }
    }
}
}

```

Bài 07: Viết chương trình phân tích số nguyên n thành các thừa số nguyên tố trong Java.
Ví dụ: $12 = 2 \times 2 \times 3$. Số nguyên dương n được nhập từ bàn phím.

Hướng dẫn:

Bước 01: Tạo Package tên là **Bai07**

Bước 02: Trong Package Bai07, tạo class tên là **ThuaSoNguyenTo** và code:

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class ThuaSoNguyenTo
{
    private static Scanner scanner = new Scanner(System.in);

    public static List<Integer> phanTichSoNguyen(int n)
    {
        int i = 2;
        List<Integer> listNumbers = new ArrayList<Integer>();
    }
}

```

```

    // phân tích
    while (n > 1) {
        if (n % i == 0) {
            n = n / i;
            listNumbers.add(i);
        } else {
            i++;
        }
    }

    // nếu listNumbers trống thì add n vào listNumbers
    if (listNumbers.isEmpty()) {
        listNumbers.add(n);
    }

    return listNumbers;
}

public static void main(String[] args)
{
    System.out.print("Nhập số nguyên dương n = ");
    int n = scanner.nextInt();
    // phân tích số nguyên dương n
    List<Integer> listNumbers = phanTichSoNguyen(n);
    // in kết quả ra màn hình
    System.out.printf("Kết quả: %d = ", n);
    int size = listNumbers.size();
    for (int i = 0; i < size - 1; i++) {
        System.out.print(listNumbers.get(i) + " x ");
    }
    System.out.print(listNumbers.get(size - 1));
}
}

```

Bài 08 : viết chương trình tính tổng các chữ số của một số nguyên n trong Java. Số nguyên dương n được nhập từ bàn phím. Ví dụ: với 1234, tổng các chữ số là $1 + 2 + 3 + 4 = 10$.

Hướng dẫn:

Bước 01: Tạo Package tên là **Bai08**

Bước 02: Trong Package Bai08, tạo class tên là **TongChuSo** và code:

```

import java.util.Scanner;

public class TongChuSo
{
    private static Scanner scanner = new Scanner(System.in);
    public static int DEC_10 = 10;

    public static int totalDigitsOfNumber(int n)
    {
        int total = 0;
    }
}

```



```

    do {
        total = total + n % DEC_10;
        n = n / DEC_10;
    } while (n > 0);
    return total;
}

public static void main(String[] args)
{
    System.out.print("Nhập số nguyên dương n = ");
    int n = scanner.nextInt();
    System.out.printf("Tổng của các chữ số " + "của %d là: %d", n, totalDigitsOfNumber(n));
}
}

```

Bài 09: Viết chương trình chuyển đổi một số tự nhiên ở hệ số 10 thành một số ở hệ cơ số B ($1 \leq B \leq 32$) bất kỳ. Giả sử hệ cơ số cần chuyển là $2 \leq B \leq 16$. Số đại diện cho hệ cơ số B > 10 là A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.

Gợi ý: Tham khảo [bảng ASCII](#) để chuyển đổi kiểu char thành String. Hàm chr(55 + m) trong ví dụ sau:

- Nếu m = 10 trả về chuỗi "A".
- Nếu m = 11 trả về chuỗi "B".
- Nếu m = 12 trả về chuỗi "C".
- Nếu m = 13 trả về chuỗi "D".
- Nếu m = 14 trả về chuỗi "E".
- Nếu m = 15 trả về chuỗi "F".

Hướng dẫn:

Bước 01: Tạo Package tên là **Bai09**

Bước 02: Trong Package Bai09, tạo class tên là **DoiCoSo** và code:

```

import java.util.Scanner;

public class DoiCoSo
{
    private static Scanner scanner = new Scanner(System.in);
    private static final char CHAR_55 = 55;

    public static String convertNumber(int n, int b)
    {
        if (n < 0 || b < 2 || b > 16 ) {
            return "";
        }

        StringBuilder sb = new StringBuilder();
        int m;
        int remainder = n;

        while (remainder > 0) {

```

```

        if (b > 10) {
            m = remainder % b;
            if (m >= 10) {
                sb.append((char) (CHAR_55 + m));
            } else {
                sb.append(m);
            }
        } else {
            sb.append(remainder % b);
        }
        remainder = remainder / b;
    }
    return sb.reverse().toString();
}

public static void main(String[] args) {
    int n = 15;
    System.out.print("Nhập số hệ thập phân n: ");
    n = scanner.nextInt();
    System.out.println("Số " + n + " trong hệ cơ số 2 = " + convertNumber(n, 2));
    System.out.println("Số " + n + " trong hệ cơ số 8 = " + convertNumber(n, 8));
    System.out.println("Số " + n + " trong hệ cơ số 16 = " + convertNumber(n, 16));
}
}

```