

情報生命科学演習 配列

東京大学院新領域・情報生命・特任准教授

加藤 毅

kato-tsuyoshi@k.u-tokyo.ac.jp

べき乗の合計のプログラムをもっと汎用的に

Test07a.java

```
public class Test07a {
    public static void main( String[] args ){
        int a, b, c, d, e, f, n;
        int pow_a, pow_b, pow_c, pow_d, pow_e, pow_f;
        int sum;
        n = 3;
        a = 3; b = 7; c = 4; d = 5; e = 9; f = 1;

        pow_a = power(a,n); pow_b = power(b,n); pow_c = power(c,n);
        pow_d = power(d,n); pow_e = power(e,n); pow_f = power(f,n);
        sum = pow_a+pow_b+pow_c+pow_d+pow_e+pow_f;
        System.out.println("sum="+sum);
    }
    public static int power( int x, int n ){
        int i, ret;
        ret = 1;
        for ( i = 0; i < n; i++ ){
            ret = ret * x;
        }
        return ret;
    }
}
```

どこかで
書き間違えそう

データ数を
10000個もあつたら
どうする？

配列を使って汎用的に

べき乗の合計を求める

```
public class Test07b {
    public static void main( String[] args ){
        int[] a;
        int[] pow_a;
        int i, n, sum;
        n = 3;
        a = new int[6];
        a[0]=3; a[1]=7; a[2]=4; a[3]=5; a[4]=9; a[5]=1;

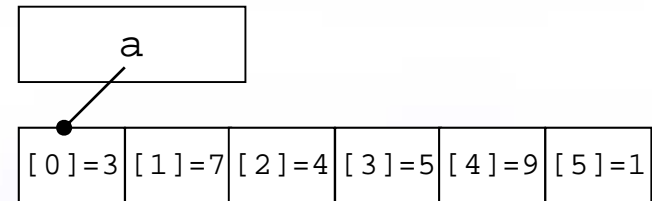
        pow_a = new int[a.length];
        for ( i = 0; i < a.length; i++ ){
            pow_a[i] = power(a[i],n);
        }
        sum = 0;
        for ( i = 0; i < pow_a.length; i++ ){
            sum = sum + pow_a[i];
        }
        System.out.println("sum="+sum);
    }
    public static int power( int x, int n ){略}
}
```

長さ6の配列を生成

int[] 型

int 配列を参照する型。
配列は複数の値を
格納できる。

番号でアクセスできる



添え字は0から始まる

a.length

a が参照している配列の
長さ

どのように計算されているのか



べき乗の合計を求める

```
public class Test07b {
    public static void main( String[] args ){
        int[] a;
        int[] pow_a;
        int i, n, sum;
        n = 3;
        a = new int[6];
        a[0]=3; a[1]=7; a[2]=4; a[3]=5; a[4]=9; a[5]=1;

        pow_a = new int[a.length];
        for ( i = 0; i < a.length; i++ ){
            pow_a[i] = power(a[i],n);
        }
        sum = 0;
        for ( i = 0; i < pow_a.length; i++ ){
            sum = sum + pow_a[i];
        }
        System.out.println("sum="+sum);
    }
    public static int power( int x, int n ){略}
}
```

長さ6の配列を生成

} べき乗の
計算

} 合計の
計算

最初 sum に 0 を入れて
一つずつ加えていっている

どのように計算されているのか



Test07b.java

べき乗の合計を求める

```
public class Test07b {
    public static void main( String[] args ){
        int[] a;
        int[] pow_a;
        int i, n, sum;
        n = 3;
        a = new int[3]; ← 長さ3の配列を生成
        a[0]=3; a[1]=7; a[2]=4;

        pow_a = new int[a.length];
        for ( i = 0; i < a.length; i++ ){
            pow_a[i] = power(a[i],n);
        }
        sum = 0;
        for ( i = 0; i < pow_a.length; i++ ){
            sum = sum + pow_a[i];
        }
        System.out.println("sum="+sum);
    }
    public static int power( int x, int n ){略}
}
```

} べき乗の
計算

} 合計の
計算

最初 sum に 0 を入れて
一つずつ加えていっている

練習7-b.



Test07b.java

プロジェクト名: test07

```
public class Test07b {
    public static void main( String[] args ){
        int[] a;
        int[] pow_a;
        int i, n, sum;
        n = 3;
        a = new int[3];
        a[0]=3; a[1]=7; a[2]=4;

        pow_a = new int[a.length];
        for ( i = 0; i < a.length; i++ ){
            pow_a[i] = power(a[i],n);
        }
        sum = 0;
        for ( i = 0; i < pow_a.length; i++ ){
            sum = sum + pow_a[i];
        }
        System.out.println("sum="+sum);
    }
    public static int power( int x, int n ){略}
}
```

長さ3の配列を生成

} べき乗の
計算

} 合計の
計算

デバッガを使って

- (1) pow_a および sum が計算されていく様子を
確認せよ
- (2) 変数 n の値を 2 などに
変更しても正しく計算
されることを確認せよ