

(2) aというint型の変数を使い、「年齢は10才です。」と表示するプログラムを書きなさい。

```
C:¥yy>java Sb
年齢は10才です。
```

ファイル名: Sb.java

```
class Sb{
    public static void main(String args[]){
        int a=10;
        System.out.println("年齢は"+a+"才です。¥n");
    }
}
```

(3) aは10、bは20と設定(int型)し、「a+b=30です。」と表示するプログラムを書きなさい。

```
C:¥yy>java Sc
a + b = 30  です。
```

ファイル名: Sc.java

```
class Sc{
    public static void main(String args[]){
        int a=10,b=20;
        System.out.print(" a + b =" + (a+b) + "  です。¥n");
    }
}
```

【ワンポイント】変数

上記の(2)では、a=10;のところを変更すれば、次の行を変更しなくとも、「年齢は20才です」「年齢は40才です」というように、表示を返ることができとても便利です。

このように、数値や文字を入れておける文字のことを**変数**といいます。

変数は、「中身がいろいろと変えられる」ということから、その名前がついています。

したがって、

```
System.out.println("年齢は"+10+"才です。¥n");
```

と書いた場合には、それは**定数**(定まっている数)といいます。定数は**リテラル**ともいいます。

【ワンポイント】型宣言

変数ではいろいろな値を代入して使えますが、代入できるものはこの型宣言によって決まります。

```
int a;
```

とした後には、aには整数しか代入できません。

```
char a;
```

とした後には、aには英数字1文字しか代入できません。

このように、「この変数にはこういうものを入れて下さいね！」と明記することを、**型宣言**といいます。型宣言をした後にそれを破るような記述はエラーになります。

```
int a;
```

```
a="z"; ) これはエラー！
```

上記の例のように、型宣言と同時に初期値を代入することも可能です。

※Javaにおける型宣言は次の通りになります。

| | | | |
|-----------|---------|------|--|
| 数値型 | byte | 1バイト | -128~127 |
| 数値型 | short | 2バイト | -32768~32767 |
| 数値型 | int | 4バイト | -2147483648~2147483647 |
| 数値型 | long | 8バイト | -9223372036854775808~9223372036854775807 |
| 文字型 | char | 2バイト | 0~65535 |
| 単制度浮動小数点型 | float | 4バイト | |
| 倍制度浮動小数点型 | double | 8バイト | |
| 論理型 | boolean | 2バイト | (true、false) |

(4) aに0~9までの乱数(でたらめな数)を代入し、表示させるプログラムを書きなさい。

```
C:\¥¥¥>java Sd
でたらめな一数は3です。
```

```
C:\¥¥¥>java Sd
でたらめな一数は6です。
```

ファイル名: Sd.java

```
import java.util.*;
class Sd{
    public static void main(String args[]){
        int a;
        Random r=new Random();
        a=r.nextInt(10);
        System.out.println("でたらめな一数は"+a+"です。¥n");
    }
}
```

【ワンポイント】乱数

乱数とはデタラメな数のことです。
 プログラムを実行したとき、あるときは5、またあるときは9、またあるときは0、というようにデタラメな数をその都度発生させる機能を持つ数です。
 乱数を使用するには次のようになります。

①乱数を使えるようにするために、乱数の入った命令セットを組み込む

```
import java.util.*;
```

~~~~~  
 ~~~~~組み込みなさい ~~~~~乱数が見える、java.utilという名前の命令セットを

②乱数の型宣言(オブジェクト作成)をする

```
Random r=new Random();
```

~~~~~  
 ~~~~~乱数を使いますよ ~~~~~新しく乱数をという名前で作ります

③乱数を発生させる

```
a=r.nextInt(10);
```

~~~~~  
 ~~~~~0~9までの乱数を発生させよ

↑
 乱数であるrに対し
 つまり、上記の例では、aという変数に0~9までの乱数が代入されます(つまり10個の乱数。個数というのがポイント)。

【ワンポイント】エスケープシーケンス

¥nが改行を示すように、Javaではそのような特殊文字をさまざまな記号で表しています。
 そのような特殊文字をエスケープシーケンスといいます。

Javaには次のようなエスケープシーケンスがあります。

| | |
|----|--------------|
| ¥" | ダブルクォーテーション |
| ¥' | シングルクォーテーション |
| ¥¥ | 円マーク |
| ¥b | バックスペース |
| ¥t | タブ |
| ¥n | 改行 |

通常、"を表示しようとする、System.out.println(" ");と書きたくところですが、そうすると"の対になるという法則が崩れてしまいますので、その場合には上記のようなエスケープシーケンスを使用することになります。System.out.println("¥");

このようにエスケープシーケンスは、通常書くどうしてもプログラム上エラーになってしまう文字を表示させるための特殊文字といえます。

- (5) a[0]からa[2]を配列宣言し、それぞれ50、100、200と代入し、合計を表示させるプログラムを書きなさい。

```
C:¥¥>java Se
350
```

ファイル名: Se.java

```
class Se{
    public static void main(String args[ ]){
        int a[ ]=new int[3];
        a[0]=50;
        a[1]=100;
        a[2]=200;
        System.out.println(a[0]+a[1]+a[2]);
    }
}
```

- (6) a[0]からa[4]までを配列宣言し、それぞれ、353、332、221、252、222 を代入し、それらを
でたらしめに表示させるプログラムを書きなさい。

```
C:¥¥>java Sf
221
```

```
C:¥¥>java Sf
332
```

ファイル名: Sf.java

```
import java.util.*;
class Sf{
    public static void main(String args[]){
        int a[ ]=new int[5];
        a[0]=353;
        a[1]=332;
        a[2]=221;
        a[3]=252;
        a[4]=222;
        Random r=new Random();
        int i;
        System.out.println(a[r.nextInt(5)]);
    }
}
```

【ワンポイント】配列変数

変数を扱いはじめると、それらをいくつも管理しなくなってくるものです。その際に、管理番号をつけて数値的に処理したくなるわけですが、それを実現してくれるのが配列変数です。配列変数では[]という括弧を変数の右につけ、あたかも連番のように処理してくれるのです。

```
int a[ ]=new int[5];
```

int型のaという配列変数を作ります
新しく作る配列変数の数は5個ですよ
(個数というのがポイント。実際には、a[0]~a[4]までの5個)

このように配列変数を使うと一つの変数名で、いくつの変数を扱えるようになり、とても便利です。後述の繰り返し命令などを覚えるとその利用範囲はさらに広がります。

