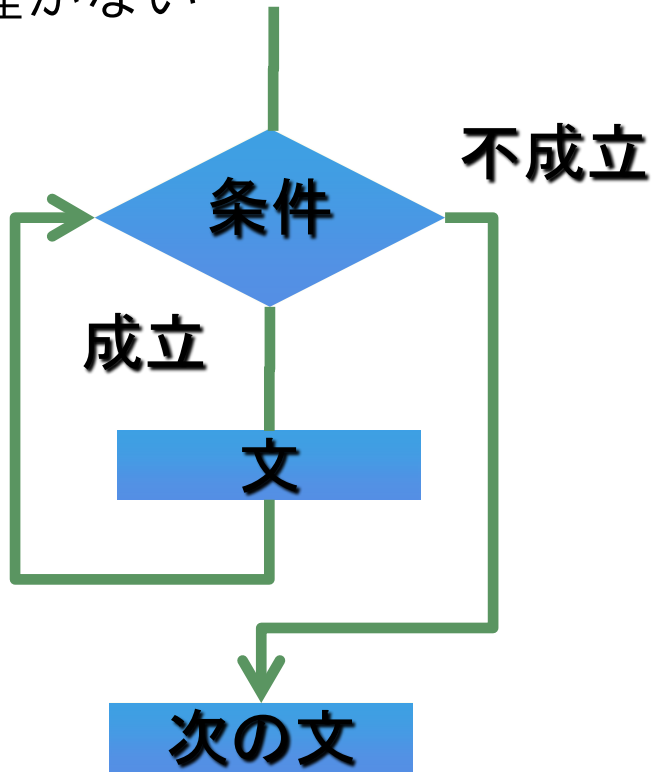
情報処理演習７

kamahara@port.kobe-u.ac.jp

ループ(WHILE文)

- for文の前処理・後処理がない

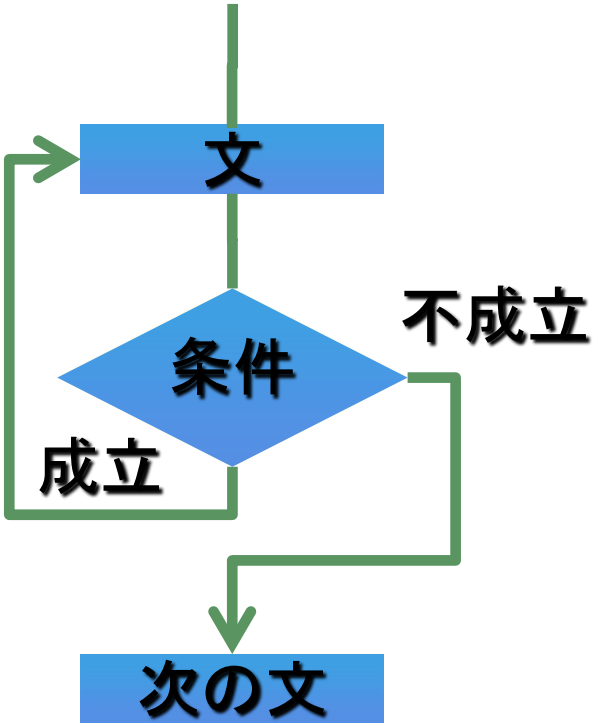
```
while ( 条件式 ) {
        文
}
次の文
```



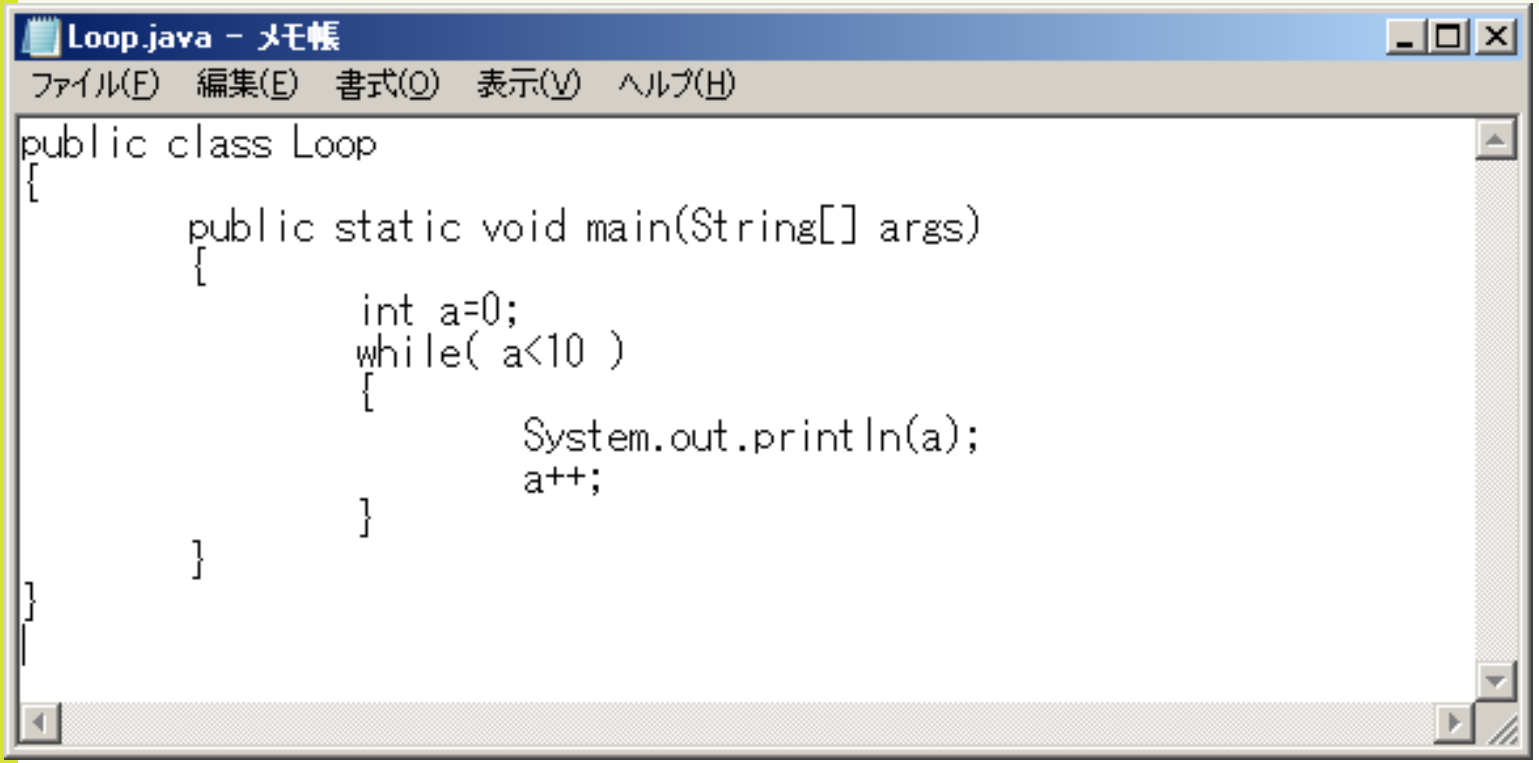
Do-WHILE文

- 文は必ず１回は実行する

```
do {
      文
} while ( 条件式 );
```

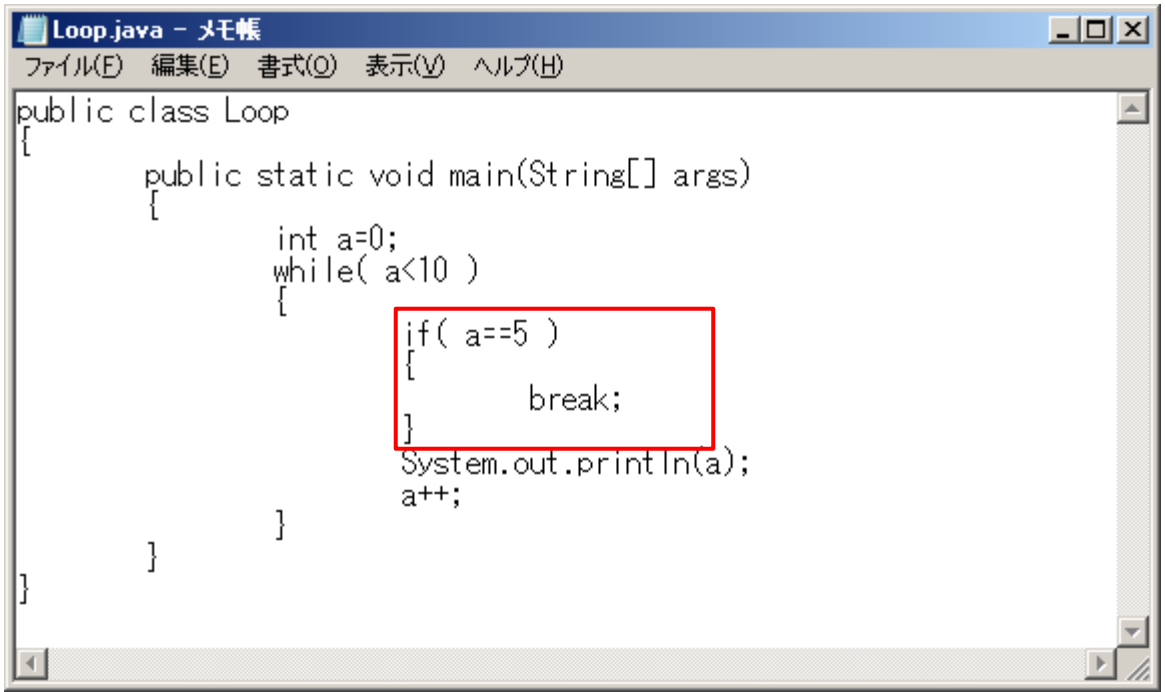


Whileループのサンプル



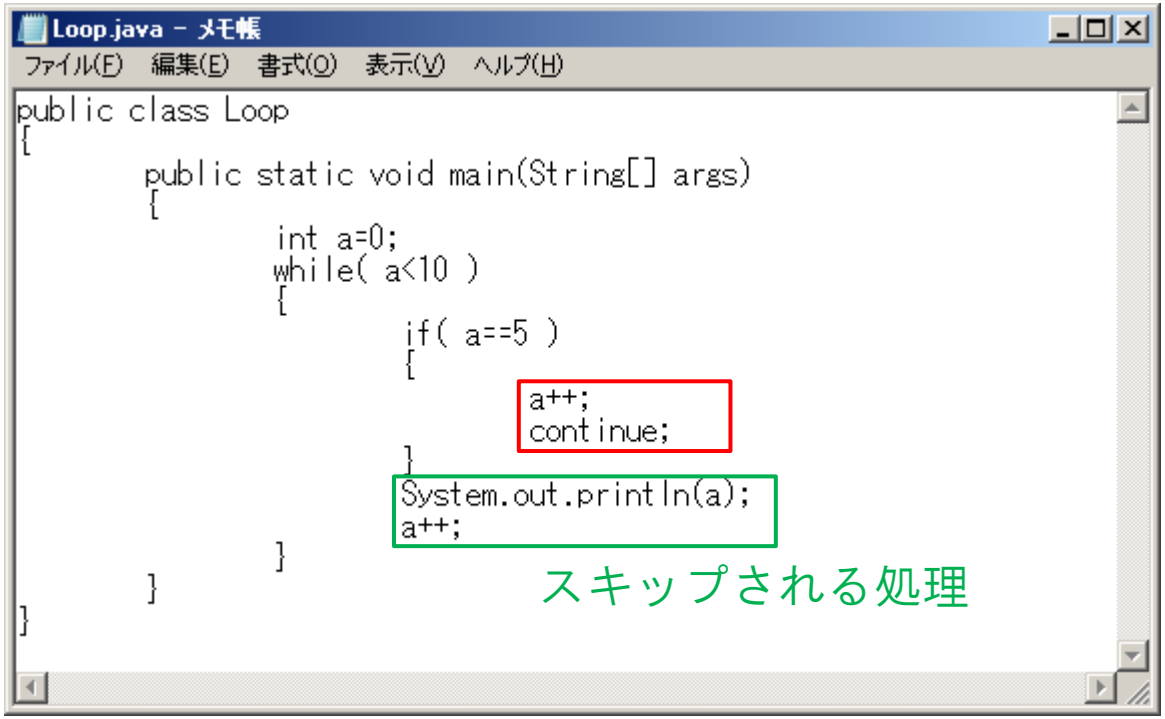
ループの制御(1)

- 文はブロックの中で複数書ける
- 途中で中断したい場合はbreakを使う



ループの制御(2)

- 処理をスキップさせたい場合はcontinueを使う



結果の違いを確認すること

ループの条件

- 条件式はループでも複数使える
- どういう結果になるか

Loop.java - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
public class Loop
{
        public static void main(String[] args)
        {
                int a=0;
                boolean f=false;
                while( (a<10) && (f!=true) )
                {
                        if( a==5 )
                        {
                                f=true;
                        }
                        System.out.println(a);
                        a++;
                }
        }
}
```

BREAK,CONTIINUE

- 流れを制御する文は、使わなくても似たことは実現できるが、簡単に書くことができる

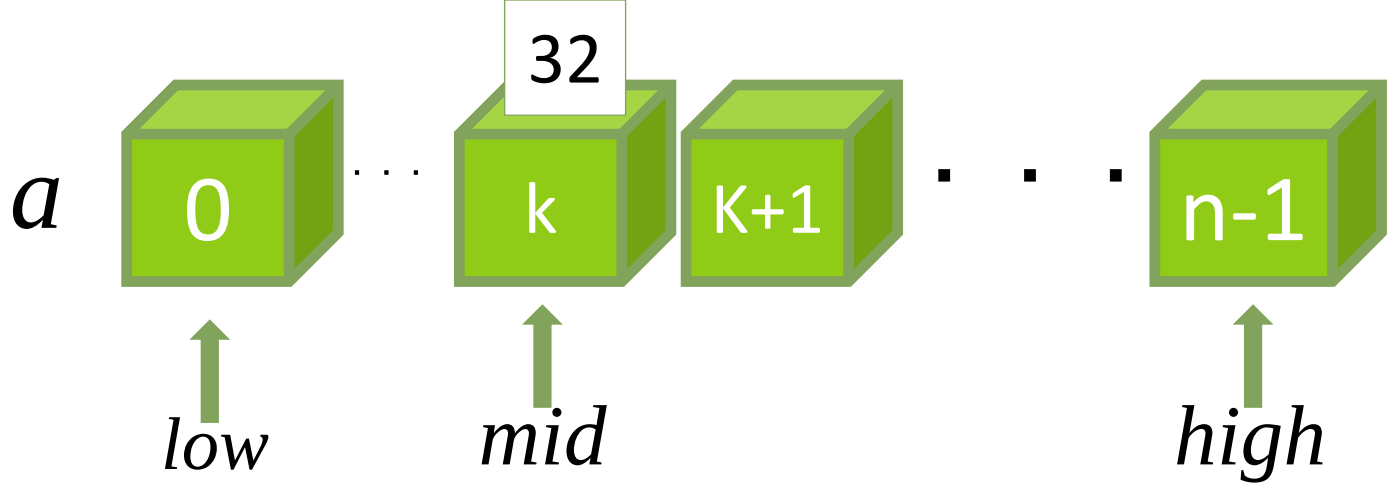
課題

- 次のプログラムは、ソートされた配列の中から、特定の数字(変数targetで指定)を発見するプログラムである
- 間違いがあるので正常に動作しない。間違いを探してプログラムを正常に動作するようにしなさい。間違いは１つとは限らない。
- 配列中のどの値でも発見できること。また最少のstep数で発見できることが望ましい。
- プログラムがどのように動作しているか変数midの値に着目した説明とプログラムをメールすること

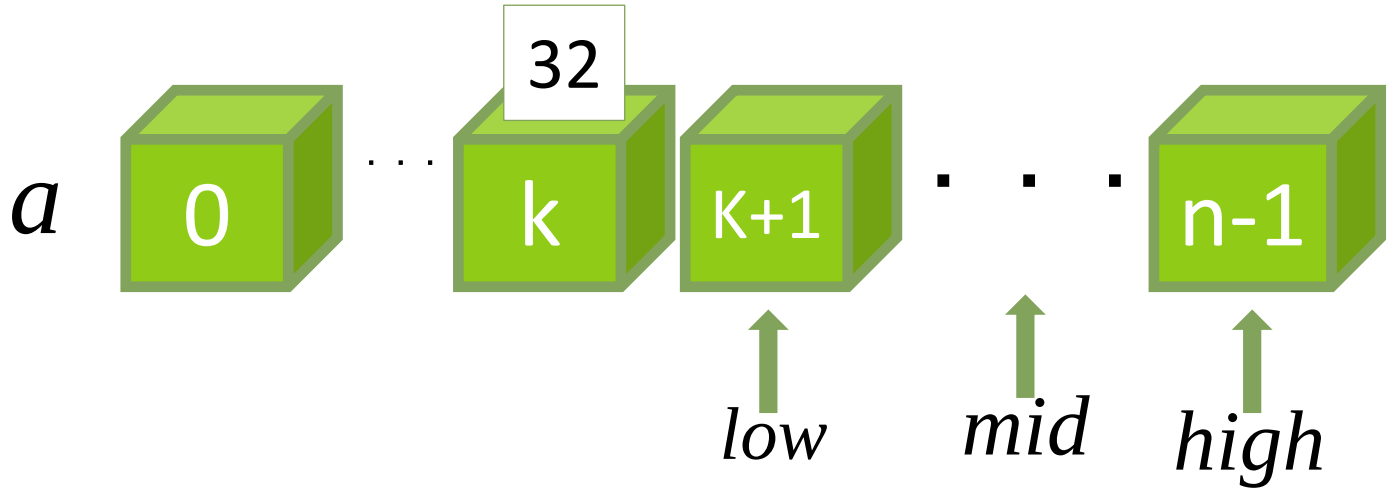
```
public class Loop
{
	public static void main(String[] args)
	{
		int[] array = { 32, 42, 2, 34, 67, 95, 4, 54, 81 };
		java.util.Arrays.sort(array); // 昇順に並び替える

		int low = 0;
		int high = array.length-1;
		int mid;
		int step=1;
		int target = 54;
		boolean f=false;
		while( low < high )
		{
			mid = (low + high ) /2;
			System.out.println("low="+low+",mid="+mid+",high="+high);
			if( target == array[mid]  )
			{
				System.out.println("発見。step="+step);
				f=true;
				break;
			}
			if( target < array[mid]  )
			{
				low = mid + 1;
			} else {
				high = mid - 1;
			}
			step++;
		}
		if( !f )
		{
			System.out.println("発見できません。");
		}

	}
}
```



target=54　　54 > 32



lowより下はさがさなくていい

課題の提出

- プログラムの動作の説明を本文に書き Loop.javaを添付ファイルとして提出する
- 件名：情報処理演習７ 学籍番号　名前
- 宛先：kamahara@port.kobe-u.ac.jp