平成23年度基盤システム演習A第3回レポ　ト

学籍番号：0312010142

講座名　：澤本研

氏名　　：藤田　拓

目 次

◆課題

1　円クラス

```
//Circle.java

public class Circle {
    protected double x, y, r;

    public Circle(double a, double b, double c) {
        x = a;
        y = b;
        r = c;
    }

    public Circle() {
        x = 0.0;
        y = 0.0;
        r = 1.0;
    }

    public double circumference() {
        return 3.1415 * r * 2;
    }

    public double area() {
        return 3.1415 * r * r;
    }

    public void transfer(double a, double b) {
        x = x + a;
        y = y + b;
    }

    public static void main(String [] args) {
        Circle r1 = new Circle(10.0, 5.0, 4.0);
        System.out.println("Circle 1:" );
        System.out.println("Circumference is " + r1.circumference() );
        System.out.println("Area is " + r1.area());
        System.out.println("---------------------------" );
        System.out.println("Transfer:" );
        r1.transfer(7.0, 10.0);
        System.out.println("After Transfer (x , y) (" + r1.x + " , " + r1.y + " )" );
    }
}
```

```
$ javac Circle.java
$ java Circle
Circle 1:
Circumference is 25.132
Area is 50.264
---------------------------
Transfer:
After Transfer (x , y) (17.0 , 15.0 )
```

2　クイックソート

```
//Qsort.java

public class Qsort {

    public static void quickSort(int[] arr, int left, int right){
        if ( left <= right ) {
            int p = arr[ (left + right) / 2 ];
            int l = left;
            int r = right;

            while( l <= r ) {
                while( arr[l] < p )
                    l++;

                while( arr[r] > p )
                    r--;

                if ( l <= r ) {
                    int tmp = arr[l];
                    arr[l]  = arr[r];
                    arr[r]  = tmp;
                    l++;
                    r--;
                }
            }

            quickSort(arr, left,     r);
            quickSort(arr,    l, right);
        }
    }

    public static void arrayPrintln(int[] arr){
        for(int i=0; i<arr.length; i++){
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
    }

    public static void main(String[] args){
        int[] arr = {1, 6, 15, 12, 7, 9, 23, 2, 10, 4, 20};

        System.out.print("Before: ");
        arrayPrintln(arr);

        quickSort(arr, 0, arr.length-1);

        System.out.print("After: ");
        arrayPrintln(arr);
    }
}
```

```
$ javac Qsort.java
$ java Qsort
Before: 1 6 15 12 7 9 23 2 10 4 20
After: 1 2 4 6 7 9 10 12 15 20 23
```

3　キュー

```
//Queue.java

public class Queue {
    private int count;
    private int capacity;
    private int capacityIncrement;
    String [] itemList;

    public Queue() {
        count    = 0;
        capacity = 5;
        capacityIncrement = 2;
        itemList = new String[capacity];
    }

    public Queue(String [] list) {
        count = list.length;
        capacity = list.length;
        capacityIncrement = 5;
        itemList = list;
    }

    public void push (String obj) {
        if(count == capacity) {
            capacity += capacityIncrement;
            String [] tempList =new String[capacity];
            for (int i = 0; i < count; i++) {
                tempList[i] = itemList[i];
            }
            itemList = tempList;
        }

        itemList[count] = obj;
        count++;
    }

    public String shift() {

        if (count == 0) {
            return null;
        }
        else {
            String temps = itemList[0];
            int y = 0;
            for(int i = 1; i < count; i++) {
                itemList[y] = itemList[i];
                y++;
            }
            count--;
            return temps;
        }
    }

    public void printItems() {
        for(int i = 0; i < count; i++) {
            System.out.print(itemList[i] + ",");
        }
        System.out.println(" ");
    }
}
```

```
$ javac Queue.java
$ javac QueueTest.java
$ java QueueTest
Initial data :


------------
Data after push Komachi :
Komachi,
------------
Data after push Yamabiko :
Komachi,Yamabiko,

------------
Data after push Tsubasa :
Komachi,Yamabiko,Tsubasa,

------------
Data after push Nasuno :
Komachi,Yamabiko,Tsubasa,Nasuno,

------------
Data after pop object Komachi :
Yamabiko,Tsubasa,Nasuno,

------------
Data after pop object Yamabiko :
Tsubasa,Nasuno,

------------
$
```