**MultiThreading AssignmentWeek-4:**

**Assignment By:**

**Abhinav Jain**

**9412005184**

[**Abhinavjainn412@gmail.com**](mailto:Abhinavjainn412@gmail.com)

**1)Create a Thread by implementing Runnable interface**

**class** Hii **implements** Runnable{

**public** **void** run() {

**for**(**int** i=1;i<=5;i++) {

System.***out***.println("Hi");

**try** {

Thread.*sleep*(500);

}

**catch**(Exception e) {}

}

}

}

**class** Helloo **implements** Runnable{

**public** **void** run() {

**for**(**int** i=1;i<=5;i++) {

System.***out***.println("Hello");

**try** {

Thread.*sleep*(500);

}

**catch**(Exception e) {}

}

}

}

**public** **class** SecondInterface {

**public** **static** **void** main(String args[]) {

Hii obj1=**new** Hii();

Helloo obj2=**new** Helloo();

Thread t1=**new** Thread(obj1);

Thread t2=**new** Thread(obj2);

t1.start();

**try** {

Thread.*sleep*(100);

}

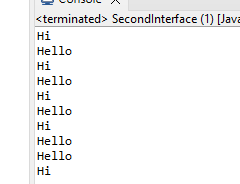
**catch**(Exception e) {}

t2.start();

}

}

**Output-**

****

**2)Print Even and Odd Numbers Using 2 Threads**

**import** java.util.Scanner;

**class** Even **extends** Thread{

**int** a,b;

Even(**int** a,**int** b){

**this**.a=a;

**this**.b=b;

}

**public** **void** run() {

**for**(**int** i=a;i<=b;i++) {

**if**(i%2==0) {

System.***out***.println("Even is : "+i);

}

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}

**class** Odd **extends** Thread{

**int** a,b;

Odd(**int** a,**int** b){

**this**.a=a;

**this**.b=b;

}

**public** **void** run() {

**for**(**int** i=a;i<=b;i++) {

**if**(i%2!=0) {

System.***out***.println("Odd is : "+i);

}

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}

**public** **class** ThreadQues2 {

**public** **static** **void** main(String args[]) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter Start number: ");

**int** a=sc.nextInt();

System.***out***.println("Enter End number: ");

**int** b=sc.nextInt();

Even e1=**new** Even(a,b);

Odd o1=**new** Odd(a,b);

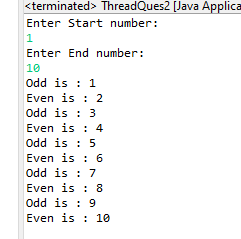
e1.start();

o1.start();

}

}

**Output-**



**3) Create two daemon thread and one user thread**

**public** **class** ThreadQues3 **extends** Thread

{

**public** ThreadQues3(String name){

**super**(name);

}

**public** **void** run()

{

**if**(Thread.*currentThread*().isDaemon())

{

System.***out***.println(getName() + " is Daemon thread");

}

**else**

{

System.***out***.println(getName() + " is User thread");

}

}

**public** **static** **void** main(String[] args)

{

ThreadQues3 t1 = **new** ThreadQues3("t1");

ThreadQues3 t2 = **new** ThreadQues3("t2");

ThreadQues3 t3 = **new** ThreadQues3("t3");

t1.setDaemon(**true**);

t1.start();

t2.start();

t3.setDaemon(**true**);

t3.start();

}

}

**Output-**

Graphical user interface, text, application

Description automatically generated

**4) Create a thread which contains below methods**

**getID();**

**isActive();**

**currentThread()**

**sleep(milliseconds)**

**public** **class** ThreadQues4 **extends** Thread

{

**public** **void** run()

{

System.***out***.println("running...");

**try** {

Thread.*sleep*(1000);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

System.***out***.println("After 1000 milliseconds ...");

}

**public** **static** **void** main(String args[])

{

ThreadQues4 t1=**new** ThreadQues4();

System.***out***.println("Name of t1: "+t1.*currentThread*().getName());

System.***out***.println("Id of t1: "+t1.*currentThread*().~~getId~~());

System.***out***.println("is t1 active "+t1.isAlive());

t1.start();

System.***out***.println("is t1 active "+t1.isAlive());

}

}

**Output-**

Text

Description automatically generated

**Collection FrameworkAssignmentWeek-5:**

**1. Implement custom HashMap**

**package** com.collection;

**import** java.util.HashMap;

**public** **class** CollectionQues1 {

**public** **static** **void** main(String[] args){

HashMap<Integer,String> h1=**new** HashMap<>();

h1.put(1, "Abhinav");

h1.put(2,"Jain");

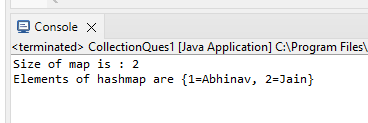
System.***out***.println("Size of map is : "+h1.size());

System.***out***.println("Elements of hashmap are "+h1);

}

}

**Output-**

****

**2. Implement Stack using Arrays or LinkedList**

**package** com.collection;

**import** java.util.Scanner;

**class** Stack{

**int** stack[]=**new** **int**[5];

**int** top=0;

**void** push(**int** data) {

**if**(top==5) {

System.***out***.println("\nStack full\n");

}**else** {

stack[top]=data;

top++;}

}

**int** pop() {

**int** data=stack[top-1];

stack[top-1]=0;

top--;

**return** data;

}

**int** peek() {

**int** data=stack[top-1];

**return** data;

}

}

**public** **class** CollectionQues2 {

**public** **static** **void** main(String args[]) {

Stack s1=**new** Stack();

**int** a;

System.***out***.println("Enter Choice : 1.Push 2.Pop 3.Peek");

Scanner sc=**new** Scanner(System.***in***);

a=sc.nextInt();

**while**(a!=0){

**switch**(a) {

**case** 1:

System.***out***.println("Enter number to push- ");

**int** data=sc.nextInt();

s1.push(data);

**break**;

**case** 2:

System.***out***.println("Element popped- ");

System.***out***.println(s1.pop());

**break**;

**case** 3:

System.***out***.println("Peak element is- ");

System.***out***.println(s1.peek());

**break**;

}

System.***out***.println("\nEnter Choice : 1.Push 2.Pop 3.Peek");

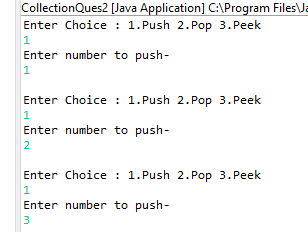
a=sc.nextInt();

}

}

}

**Output-**

**Text

Description automatically generated**

**3. Write a program to sort an array having Os and 1s inefficient**

**way. Ex: Input->1.0.1.0.0.0.1.1**

**Output-> 0,0,0,0,1,1,1,1,1**

**package** com.collection;

**import** java.util.Scanner;

**public** **class** CollectionQues3 {

**public** **static** **void** main(String args[]) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter size of array - ");

**int** n=sc.nextInt();

**int** arr[]=**new** **int**[n];

**int** zero=0;

System.***out***.println("Enter elements of array - ");

**for**(**int** i=0;i<n;i++) {

arr[i]=sc.nextInt();

**if**(arr[i]==0) {

zero++;

}

}

**for**(**int** i=0;i<zero;i++) {

arr[i]=0;

}

**for**(**int** i=zero;i<n;i++) {

arr[i]=1;

}

System.***out***.println("sorted list is- \n");

**for**(**int** i=0;i<n;i++) {

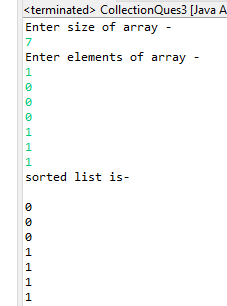
System.***out***.println(arr[i]+" ");

}

}

}

**Output-**

****

**4. Write a program to check for the balanced parenthesis**

**Ex:Input-> {()])**

**Output-> false**

**Input-> {0]**

**Output-> true**

**package** com.collection;

**import** java.util.ArrayDeque;

**import** java.util.Deque;

**import** java.util.Scanner;

**class** Solution {

**boolean** isValid(String s) {

Deque<Character> ss = **new** ArrayDeque<Character>();

**for**(**int** i=0;i<s.length();i++){

**char** ch=s.charAt(i);

**if**(ch=='(' || ch=='{' || ch=='['){

ss.push(ch);

}

**else** **if**(ss.isEmpty()||(ch=='}'&&ss.peek()!='{') || (ch==')'&&ss.peek()!='(') || (ch==']'&&ss.peek()!='[')){

**return** **false**;

}

**else**

ss.pop();

}

**return** ss.isEmpty();

}

};

**public** **class** CollectionQues4 {

**public** **static** **void** main(String args[]) {

Scanner sc= **new** Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String str= sc.nextLine();

Solution s1=**new** Solution();

System.***out***.println(s1.isValid(str));

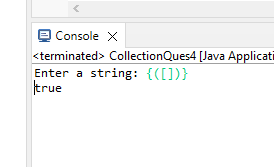
}

}

**Output-**

**Graphical user interface, text, application

Description automatically generated**

****

**Assignment By:**

**Abhinav Jain**

**9412005184**

[**Abhinavjainn412@gmail.com**](mailto:Abhinavjainn412@gmail.com)