Assignment for Java

Task 1

Find the greatest storage space

public class Task1

{

public static void main(String[] args)

{

int gyin[]={83,34,22,45,34,45,21,9};

int greatest=gyin [0];

for (int i=1;i<8;i++)

{

if (gyin[i]>greatest)

{

greatest=gyin[i];

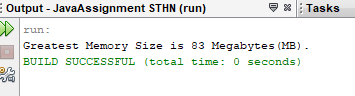
}

}

System.out.println ("Greatest Memory Size is "+greatest+" Megabytes(MB).");

}

}



Task 2

Find how many files sent ,total memory size and average file size

At first,we need to put the arrays into the data.

To measure the different memory sizes, we must use gigabytes.

And then, we need to find the largest memory size.

Then,we should find how many files are in the data with file length.

To find it,we need to use file length.

We count the memory size and find by using looping.

And,we combine all numbers and multiplied by the numbers of integers.

public class Task2

public static void main(String[] args)

int count= 0;

int total=0;

int avg=0;

for (int i=0;i<hello.length;i++)

{

total=total+hello[i];

count=count+1;

}

avg=total/count;

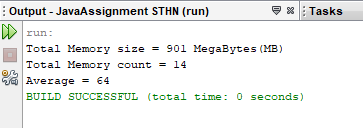
System.out.println("Total Memory size = "+total+" MegaBytes(MB)");

System.out.println("Total Memory count = "+count);

System.out.println("Average = "+avg);

}

}



# Task 3

## Create user name

At first, we give " Task 3" as a class name.

We used Input Streamreader to receive the data from keyboard.

And, used buffer reader to read the words from received data.

t

If we want to read each line,we need to use read line method.

When we want to cut the sentence, use substring.

But , this program is can only use for two word.

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Task3

{

public static void main(String[] args)throws IOException

{

System.out.print("Enter your First Name");

String fn=getchar();

String n=fn.toUpperCase();

System.out.println("Enter your last name");

String ln=getString();

System.out.print("Your name is"+n+" "+ln);

}

public static String getString()throws IOException

{

InputStreamReader Str=new InputStreamReader(System.in);

BufferedReader br=new BufferedReader(Str);

String s=br.readLine();

return s;

}

public static String getchar()throws IOException

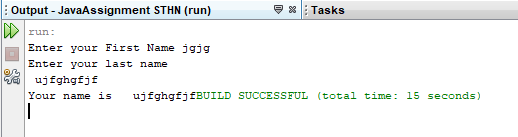
{

String s=getString();

return s.substring(0,1);

}

}



Task 4

Sorting times in order of speed

Firstable,we need to sorting which means to arrange the values in the array.

So,we need to use bibble sort to make it.

Bubble sort means compare and change the adjacent bells.

And then,arrays are working from the beginning to the end repeately.

public class Task4

{

Task4(double times[])

{

fasttime(times);

orderspeed(times);

}

public void orderspeed(double times[])

{

System.out.println("order speed");

for(int i=0;i<times.length;i++)

{

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

}

}

public void fasttime (double times[])

{

double temp;

for (int i=0;i<times.length;i++)

{

for(int j=1;j<(times.length-i);j++)

{

if(times[j-1]>times[j])

{

temp=times[j-1];

times[j-1]=times[j];

times[j]=temp;

}

}

}

}

public static void main(String[] args)

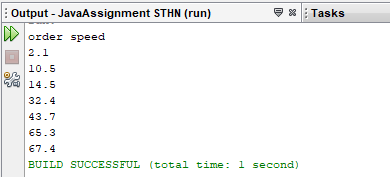
{

double count[]={32.4,43.7,14.5,65.3,67.4,10.5,2.1};

Task4 a = new Task4(count);

}

}



Task 5

Report(Java program\_Sorting times in order of speed)

At first,we need to put the arrays into the data.

To measure the different memory sizes, we must use gigabytes.

And then, we need to find the largest memory size.

Then,we should find how many files are in the data with file length.

To find it,we need to use file length.

We count the memory size and find by using looping.

And,we combine all numbers and multiplied by the numbers of integers.

At first, we give " Task 3" as a class name.

We used Input Streamreader to receive the data from keyboard.

And, used buffer reader to read the words from received data.

If we want to read each line,we need to use read line method.

When we want to cut the sentence, use substring.

But , this program is can only use for two word.

Firstable,we need to sorting which means to arrange the values in the array.

So,we need to use bibble sort to make it.

Bubble sort means compare and change the adjacent bells.

And then,arrays are working from the beginning to the end repeately.