Crunch The Numbers

public static Numbers[] Met(int []intArray){

Numbers []numbersArr=new Numbers[intArray.length];

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

numbersArr[i]=new Numbers();

numbersArr[i].num=intArray[i];

if(numbersArr[i].num%2==0){

numbersArr[i].even=true;

numbersArr[i].odd=false;

}

else{

numbersArr[i].odd=true;

numbersArr[i].even=false;

}

int c=0;if(intArray[i]==1){c++;}

else{

for(int j=2;j<intArray[i];j++){

if(intArray[i]%j==0){

c++;}

}}

if(c==0){

numbersArr[i].prime=true;

}

else{numbersArr[i].prime=false;}

}

return numbersArr;

}

Number Game V2

class Number{

int num;

public void plus(Number N){

num+=N.num;

}

public void minus(Number N){

num-=N.num;

}

public void times(Number N){

num\*=N.num;

}

public void dividedBy(Number N){

num/=N.num;

}

}

A Board Game

class Game{

int board[][];

int pos\_row,pos\_col;

Game(int a,int b,int c){

board=new int[a][a];

pos\_row=b;

pos\_col=c;

}

public void draw(){

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

for(int j=0;j<board[0].length;j++){

if(i==pos\_row&&j==pos\_col){

System.out.print(1+" ");}

else{

System.out.print(0+" ");}

}System.out.println();

}

}

}

Board Game With Movement

class Game{

int board[][];

int pos\_row,pos\_col;

Game(int a,int b,int c){

board=new int[a][a];

pos\_row=b;

pos\_col=c;

}

public void moveUp(){

if(pos\_row>0){

pos\_row--;

}

}

public void moveDown(){

if(pos\_row<board.length-1){

pos\_row++;

}

}

public void moveLeft(){

if(pos\_col>0){

pos\_col--;

}

}

public void moveRight(){

if(pos\_col<board.length-1){

pos\_col++;

}

}

public void draw(){

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

for(int j=0;j<board[0].length;j++){

if(i==pos\_row&&j==pos\_col){

System.out.print(1+" ");}

else{

System.out.print(0+" ");}

}System.out.println();

}

}

}

Taxing Issues

class IndiaTax extends WorldTax{

public double salesTax(int amount){

return (18\*amount)/100;

}

public double propertyTax(int amount){

return (20\*amount)/100;

}

}

class ChinaTax extends WorldTax{

public double salesTax(int amount){

return (10\*amount)/100;

}

}

class JapanTax extends WorldTax{

public double salesTax(int amount){

return (5\*amount)/100;

}

public double propertyTax(int amount){

return (5\*amount)/100;

}

public double incomeTax(int amount){

return (5\*amount)/100;

}

}

class CanadaTax extends WorldTax{

public double salesTax(int amount){

return (25\*amount)/100;

}

public double incomeTax(int amount){

return (25\*amount)/100;

}

}

Adventure Land

class Racer extends Player{

public void accelerate(){

if(boost){

speed=speed+25;}

}

}

class Astronaut extends Player{

public void damage(){

if(shield){

health=health-35;}

else{

health=health-50;}

}

}

class Mermaid extends Player{

public void accelerate(){

if(boost){

speed=speed+18;}}

}

class Dolphin extends Player{

public void accelerate(){

if(boost){

speed=speed+10;}

}

public void damage(){

if(shield){

health=health-20;}

else{

health=health-50;}

}

}

class Lion extends Player{

public void accelerate(){

if(boost){

speed=speed+5;}

}

}

Largest Mirror

class MyReflect{

String data;

MyReflect(String data){

this.data=data;

}

public String largestMirror(){

int i=0,j=data.length()-1;

if(data.length()==1||data.length()==2){

return data;

}

else{

while(i<j){

if(data.charAt(i)==data.charAt(j)){

i++;j--;

}

else{break;}

}

if(i==0){

this.data=" ";

}

else if(i==j||data.length()==2){

data=data;

}else{

data=data.substring(0,i);

}

return data;}

}

}

Sum Literally

class Word{

String data;

Word(String data){

this.data=data;

}

public int literalSum(){

int c=0;

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

if(Character.isDigit(data.charAt(i))){

c+=Character.getNumericValue(data.charAt(i));

}

}

return c;

}

}

What's Your Domain?

class Website{

String address;

Website(String address){

this.address=address;

}

public void printDomain(){

try{

URL u=new URL(address);

System.out.println(u.getHost());}

catch(Exception e){System.out.println(e);}

}

}