MSSV:51900491

Tên :Cao Nguyễn Kỳ Duyên

**LECTURE 9**

**Slide 45-48**

class Fraction {

private final int numerator;

private final int denominator;

Fraction() {

numerator = 0;

denominator = 1;

}

Fraction (int numerator, int denominator) {

this.numerator = numerator;

if (denominator == 0) {

throw new IllegalArgumentException("Denominator cannot be zero.");

}

else {

this.denominator = denominator;

}

}

public String getFraction() {

return numerator + "/" + denominator;

}

public double getFractionDecimal() {

return ((double) numerator) / denominator;

}

// Mathematical functions

public String addFraction(Fraction otherFraction) {

// Sum formula: a/b + c/d = (ad + cb)/bd

int a = this.numerator;

int b = this.denominator;

int c = otherFraction.numerator;

int d = otherFraction.denominator;

return ((a\*d) + (c\*b)) + "/" + (b\*d);

}

public String subtractFraction(Fraction otherFraction) {

// Subtraction formula: a/b - c/d = (ad - cb)/bd

int a = this.numerator;

int b = this.denominator;

int c = otherFraction.numerator;

int d = otherFraction.denominator;

return ((a\*d) - (c\*b)) + "/" + (b\*d);

}

public String multiplyByFraction(Fraction otherFraction) {

// Multiplication formula: a/b \* c/d = ac/bd

int a = this.numerator;

int b = this.denominator;

int c = otherFraction.numerator;

int d = otherFraction.denominator;

return (a\*c) + "/" + (b\*d);

}

public String divideByFraction(Fraction otherFraction) {

// Division formula: (a/b) / (c/d) = ad/bc

int a = this.numerator;

int b = this.denominator;

int c = otherFraction.numerator;

int d = otherFraction.denominator;

return (a\*d) + "/" + (b\*c);

}

}

**Slide 49-50**

class FractionArr implements FractionI{

private int[] members = new int[2];

public FractionArr() {

this(1,1);

}

public FractionArr(int n, int d) {

setNumer(n);

setDenom(d);

}

public FractionI add(FractionI f){

this.setDenom(this.getDenom() \* f.getDenom());

this.setNumer(this.getNumer() \* f.getDenom()

+ f.getNumer() \* this.getDenom());

}

public FractionI minus(FractionI f){

this.setDenom(this.getDenom() \* f.getDenom());

this.setNumer(this.getNumer() \* f.getDenom()

- f.getNumer() \* this.getDenom());

}

public FractionI times(FractionI f){

this.setNumer(this.getNumer() \* f.getNumer());

this.setDenom(this.getDenom() \* f.getDenom());

}

public FractionI divide(FractionI f) throws ArithmeticException{

this.setNumer(this.getNumer() \* f.getDenom());

this.setDenom(this.getDenom() \* f.getNumer());

if(this.getDenom()==0){

throw new ArithmeticException("Division by zero error!");

}

}

public FractionI simplify(){

this.setNumer(this.getNumer()/gcd(this.getDenom(), this.getNumer()));

this.setDenom(this.getDenom()/gcd(this.getDenom(), this.getNumer()));

}

public int getNumer(){

return members[0];

}

public int getDenom(){

return members[1];

}

public void setNumer(int numer){

members[0] = numer;

}

public void setDenom(int denom){

members[1] = denom;

}

public int gcd(int first, int second){

if(second ==0){

return first;

}

else if(first%second==0){

return second;

} else {

return gcd(second, first%second);

}

}

}