Лабораторна робота 3  
 з дисципліни «Об’єктно-орієнтоване програмування»

Тема роботи: «Відношення між класами. Агрегація»  
 Варіант 4

Виконав студент

групи КН-22

Пашковський П. В.

Перевірила:

Красовська Г. В.

**Завдання.**

Клас Fraction для подання скороченого дробу, що подається парою цілих чисел: чисельник та знаменник. Реалізувати операції додавання/віднімання, множення/ділення, порівняння дробу. Реалізувати метод, який подає дріб число як рядок. Метод для скорочення дробу реалізувати як приватний.

**Fraction** (Рис.1) має два приватні поля для зберігання пари чисел

**Опис структури класу.**

Поля:

* double numerator – чисельник;
* double denominator – знаменник.

Методи:

* void optimize() – метод для спрощення дробу;
* private int wholeFraction() - метод виділення цілої частини дробу;
* void print() – метод друкує об’єкт;
* void scan() – метод зчитування об’єкту;
* bool equals(Object o) – оператор співпадіння дробів;
* String toString() - повертає об’єкт класу у вигляді строки .

**Calculator**(Рис. 1) клас калькулятор з наведеними арифметичними операціями над числами Fraction.

**Опис структури класу.**

Поля:

* Fraction base – агрегує клас Fraction, описує базовий дріб.

Методи:

* public Fraction mul(Fraction b) – метод множення base на b;
* public Fraction div(Fraction b) – метод ділення base на b;
* public Fraction min(Fraction b) – метод віднімання base на b;
* public Fraction add(Fraction b) – метод додавання base на b.



*Рисунок 1 UML діаграма класу Point2D та Triangle*

**Аналіз роботи програми.**

Тест 1:



*Рисунок 2 екземпляри класу та Calculator*

Enter fraction a/b:

1 4

Enter fraction a/b:

2 3

Fraction{numerator=2, denominator=12}

Fraction{numerator=3, denominator=8}

Fraction{numerator=11, denominator=12}

Fraction{numerator=-5, denominator=12}Тест 2:



*Рисунок 3 екземпляри класу Fraction та Calculator*

Enter fraction a/b:

4

9

Enter fraction a/b:

10

18

Fraction{numerator=40, denominator=162}

Fraction{numerator=72, denominator=90}

Fraction{numerator=1, denominator=1}

**Код програм.**

//Main.java

*public class Main* {  
  
 *public static void* main(*String*[] *args*) {  
 *Fraction* one = *new* Fraction();  
 *Fraction* two = *new* Fraction();  
 one.scan();  
 two.scan();  
 *Calculator* cal = *new* Calculator(one);  
  
 *System*.out.println(cal.mul(two));  
 *System*.out.println(cal.div(two));  
 *System*.out.println(cal.add(two));  
 *System*.out.println(cal.min(two));  
 *System*.out.println();  
 one.scan();  
 two.scan();  
 cal.setBase(one);  
 *System*.out.println(cal.mul(two));  
 *System*.out.println(cal.div(two));  
 *System*.out.println(cal.add(two));  
 *System*.out.println(cal.min(two));  
 *System*.out.println();  
 }  
}

//Fraction.java

*import java.util.Scanner*;  
  
*public class Fraction* {  
 *private int* numerator = 0;  
 *private int* denominator = 1;  
 *private void* optimize() {  
 *if*(denominator%numerator == 0){  
 denominator /= numerator;  
 numerator = 1;  
 }  
 }  
  
 *public* Fraction() {  
 }  
  
 *public* Fraction(*int numerator*, *int denominator*) {  
 *this*.numerator = *numerator*;  
 *this*.denominator = *denominator*;  
 optimize();  
 }  
  
 *public int* getNumerator() {  
 *return* numerator;  
 }  
  
 *public void* setNumerator(*int numerator*) {  
 *this*.numerator = *numerator*;  
 }  
  
 *public int* getDenominator() {  
 *return* denominator;  
 }  
  
 *public void* setDenominator(*int denominator*) {  
 *this*.denominator = *denominator*;  
 }  
  
 *public void* scan() {  
 *System*.out.println("Enter fraction a/b: ");  
 *Scanner* s = *new* Scanner(*System*.in);  
 numerator = s.nextInt();  
 denominator = s.nextInt();  
 optimize();  
 }  
  
 *@Override  
 public String* toString() {  
 *return* "Fraction{" +  
 "numerator=" + numerator +  
 ", denominator=" + denominator +  
 '}';  
 }  
}

//Calculator.java

*public class Calculator* {  
 *private Fraction* base;  
  
 *public* Calculator(*Fraction base*) {  
 *this*.base = *base*;  
 }  
  
 *public void* setBase(*Fraction base*) {  
 *this*.base = *base*;  
 }  
  
 *public Fraction* mul(*Fraction b*) {  
 *return new* Fraction(base.getNumerator() \* *b*.getNumerator(),  
 base.getDenominator() \* *b*.getDenominator());  
 }  
 *public Fraction* div(*Fraction b*) {  
 *return new* Fraction(base.getNumerator() \* *b*.getDenominator(),  
 base.getDenominator() \* *b*.getNumerator());  
 }  
 *public Fraction* add(*Fraction b*) {  
 *int* top = base.getNumerator()\**b*.getDenominator() + base.getDenominator() \**b*.getNumerator();  
 *int* bot = base.getDenominator() \* *b*.getDenominator();  
 *return new* Fraction(top, bot);  
 }  
 *public Fraction* min(*Fraction b*) {  
 *int* top = base.getNumerator()\**b*.getDenominator() - base.getDenominator() \**b*.getNumerator();  
 *int* bot = base.getDenominator() \* *b*.getDenominator();  
 *return new* Fraction(top, bot);  
 }  
}

**Код програм.**

//main.cpp

#include <iostream>  
#include "Fraction.h"  
#include "Calculator.h"  
  
*using namespace* std;  
  
*int* main() {  
 Fraction one = Fraction();  
 Fraction two = Fraction();  
 one.scan();  
 two.scan();  
 Calculator cal = Calculator(one);  
  
 cout **<<** cal.mul(two) **<<** endl;  
 cout **<<** cal.div(two) **<<** endl;  
 cout **<<** cal.add(two) **<<** endl;  
 cout **<<** cal.min(two) **<<** endl;  
 cout **<<** endl;  
 one.scan();  
 two.scan();  
 cal.setBase(one);  
 cout **<<** cal.mul(two) **<<** endl;  
 cout **<<** cal.div(two) **<<** endl;  
 cout **<<** cal.add(two) **<<** endl;  
 cout **<<** cal.min(two) **<<** endl;  
 cout **<<** endl;  
 *return* 0;  
}

//fraction.h

*//  
// Created by Artem Martus on 17.11.2019.*#pragma once  
#include <iostream>  
#include <string>  
*using namespace* std;  
  
*class* Fraction {  
 *int* numerator = 0;  
 *int* denominator = 1;  
  
 *void* optimize() {  
 *if* (denominator % numerator == 0) {  
 denominator /= numerator;  
 numerator = 1;  
 }  
 }  
  
*public*:  
  
 Fraction() {  
 }  
  
 Fraction(*int numerator*, *int denominator*) : numerator(*numerator*), denominator(*denominator*) {  
 optimize();  
 }  
  
 *int* getNumerator() {  
 *return* numerator;  
 }  
  
  
 *int* getDenominator() {  
 *return* denominator;  
 }  
  
  
 *void* scan() {  
 cout**<<**("Enter fraction a/b: ");  
 cin**>>** numerator **>>** denominator;  
 optimize();  
 }  
  
 Fraction **operator** \* (Fraction *b*) {  
 *return* {getNumerator() \* *b*.getNumerator(),  
 getDenominator() \* *b*.getDenominator()};  
 }  
  
 Fraction **operator** / (Fraction *b*) {  
 *return* {getNumerator() \* *b*.getDenominator(),  
 getDenominator() \* *b*.getNumerator()};  
 }  
 Fraction **operator** + (Fraction *b*) {  
 *int* top = getNumerator() \* *b*.getDenominator() + getDenominator() \* *b*.getNumerator();  
 *int* bot = getDenominator() \* *b*.getDenominator();  
 *return* {top, bot};  
 }  
 Fraction **operator** - (Fraction *b*) {  
 *int* top = getNumerator() \* *b*.getDenominator() - getDenominator() \* *b*.getNumerator();  
 *int* bot = getDenominator() \* *b*.getDenominator();  
 *return* {top, bot};  
 }  
 *string* toString() {  
 *return string*("Fraction{").append(  
 "numerator=") **+** to\_string(numerator).append(  
 ", denominator=") **+** to\_string(denominator).append("}");  
 }  
};  
  
*ostream*& **operator** << (*ostream*& *o*, Fraction *f*){  
 *o* **<<** *f*.toString()**<<**endl;  
 *return o*;  
}

//calculator.h

*//  
// Created by Artem Martus on 17.11.2019.  
//*#pragma once  
  
#include "Fraction.h"  
  
*class* Calculator {  
 Fraction base;  
*public*:  
 Calculator(*const* Fraction &*base*) : base(*base*) {}  
  
 *void* setBase(Fraction *base*) {  
 *this*->base = *base*;  
 }  
  
 Fraction mul(Fraction *b*) {  
 *return* base **\*** *b*;  
 }  
  
 Fraction div(Fraction *b*) {  
 *return* base **/** *b*;  
 }  
  
 Fraction add(Fraction *b*) {  
 *return* base **+** *b*;  
 }  
  
 Fraction min(Fraction *b*) {  
 *return* base **-** *b*;  
 }  
};