**<윈도우즈 프로그래밍 과제#6 - 클래스>**

**20194111 최민규**

Q. 다음은 분수에 대한 클래스이다.

|  |
| --- |
| class Fraction  {  public int numberator; // 분자  public int denominator; // 분모  } |

(1) 한 개의 정수를 받아 초기화 하는 생성자를 작성하시오.

(2) 두 개의 정수를 받아 초기화 하는 생성자를 작성하시오.

(3) 하나의 분수를 분자/분모 형태로 반환하는 ToString() 메소드를 작성하시오.

(4) 최대 공약수를 구하는 메소드와 기약 분수로 만드는 메소드를 작성하시오.

(5) 분수에 대한 4칙 연산을 수행하는 메소드 AddFraction, SubFraction, MulFraction, DivFraction을 작성하시오.

(6) 테스트 클래스를 만들어 테스트 하시오.

|  |
| --- |
| //20194111 최민규  class Fraction  {  public int numberator; // 분자  public int denominator; // 분모  public Fraction(int a)  {  numberator = a;  denominator = 0;  } // (1) 한 개의 정수를 받아 초기화하는 생성자.  public Fraction(int a, int b)  {  numberator = a;  denominator = b;  } // (2) 두 개의 정수를 받아 초기화 하는 생성자.  public override string ToString() // overide를 안쓰고 메소드를 새롭게 구현해도 됨  {  return "(" + numberator + " / " + denominator + ")";  } // (3) 하나의 분수를 분자/분모 형태로 변환하는 ToString() 메소드  public int GratestCommonDivisor(int a, int b)  {  int result = 1; //결과값 할당 변수  //대소 비교해서 a가 항상 더 작은 값을 갖도록 swap  if (a > b)  {  int tmp = a;  a = b;  b = tmp;  }  //1~a범위에서 a,b의 공약수 중 가장 큰 값을 result에 할당하도록 반복  for (int i = 1; i <= a; i++)  {  if (a % i == 0 && b % i == 0)  result = i;  }  return result; //결과 반환  } // (4) 최대 공약수를 반환하는 메서드.  public Fraction addFraction(Fraction a, Fraction b)  {  Fraction result = new Fraction(0,0); //결과값 할당 변수  result.numberator = a.numberator\*b.denominator + b.numberator\*a.denominator; //분자는 서로의 분모를 곱하고 더함.  result.denominator = a.denominator \* b.denominator; //분모끼리 곱하고 결과의 분모로 할당.  Console.WriteLine("f1 + f2 = " + "("+result.numberator + "/" + result.denominator+")"); //결과 출력  return result; //결과 반환  } // (5) 덧셈 결과를 출력 및 반환하는 메서드.  public Fraction subFraction(Fraction a, Fraction b)  {  Fraction result = new Fraction(0, 0); //결과값 할당 변수  result.numberator = a.numberator\*b.denominator - b.numberator\*a.denominator; //분자는 서로의 분모를 곱하고 뺌.  result.denominator = a.denominator \* b.denominator; //분모끼리 곱하고 결과의 분모로 할당.  Console.WriteLine("f1 - f2 = " +"("+ result.numberator + "/" + result.denominator+")"); //결과 출력  return result; //결과 반환  } // (6) 뺄셈 결과를 출력 및 반환하는 메서드.  public Fraction mulFraction(Fraction a, Fraction b)  {  Fraction result = new Fraction(0, 0); //결과값 할당 변수  result.numberator = a.numberator \* b.numberator; //분자끼리 곱함  result.denominator = a.denominator \* b.denominator; //분모끼리 곱함  Console.WriteLine("f1 \* f2 = " + "("+result.numberator +"/"+result.denominator+")"); //결과 출력  return result; //결과 반환  } // (7) 곱셈 결과를 출력 및 반환하는 메서드.  public Fraction divFraction(Fraction a, Fraction b)  {  Fraction result = new Fraction(0, 0); //결과값 할당 변수  result.numberator = a.numberator \* b.denominator; //분자에 분모 곱함. (나눗셈을 역수를 이용한 곱으로 나타냄.)  result.denominator = a.denominator \* b.numberator; //분모에 분자 곱함.  Console.WriteLine("f1 ÷ f2 = " +"("+ result.numberator + "/" + result.denominator+")"); //결과 출력  return result; //결과 반환  } // (8) 나눗셈 결과를 출력 및 반환하는 메서드.  public Fraction irreducibleFraction(Fraction a)  {  int x = a.numberator; //분자  int y = a.denominator; //분모  int tmp = a.GratestCommonDivisor(x, y); //분자와 분모의 최대공약수  while (tmp != 1) //분자와 분모의 최대 공약수가 1이 될 때까지 분자, 분모를 각각 최대공약수로 나눔.  {  x /= tmp;  y /= tmp;  tmp = a.GratestCommonDivisor(x, y);  }  //결과값을 a 변수에 할당  a.numberator = x;  a.denominator = y;  //결과 출력  Console.WriteLine("기약 분수로 표현한 분수 식: " + x + "/" + y);  //결과 반환  return a;  } // (9) 기약분수값을 출력 및 반환하는 메서드.  }  class Program  // 선언된 클래스 Fraction의 메소드를 테스트 하기 위해 테스트 클래스를 선언.  {  static void Main(string[] args)  {  Fraction f1, f2; // 분수를 나타낼 클래스 f1, f2를 선언.  f1 = new Fraction(5, 6);  // 생성자를 이용하여 분수식 f1을 분자 5, 분모 6으로 초기화  f2 = new Fraction(2, 4);  // 생성자를 이용하여 분수식 f2를 분자 2, 분모 4로 초기화  Console.WriteLine("분수식 f1: " + f1.ToString());  Console.WriteLine("분수식 f2: " + f2.ToString());  // f1, f2를 분수식의 형태로 출력  Console.WriteLine("분수식 f1의 최대공약수: "  + f1.GratestCommonDivisor(f1.numberator, f1.denominator));  Console.WriteLine("분수식 f2의 최대공약수: "  + f2.GratestCommonDivisor(f2.numberator, f2.denominator));  // 분수식 f1, f2의 최대공약수를 출력한다.  f1.irreducibleFraction(f1);  f2.irreducibleFraction(f2); // 분수식 f1, f2의 기약분수를 출력.  f1.addFraction(f1, f2);  f1.subFraction(f1, f2);  f1.mulFraction(f1, f2);  f1.divFraction(f1, f2); //분수식 f1, f2의 사칙연산 결과 출력.  }  } |
| 실행결과  \*기약분수 함수(irreducibleFraction)를 출력만 하고 반환은 하지 않았을 때  (출력 예시와 똑같이 출력되도록)    \*기약분수 함수(irreducibleFraction)를 출력도 하고 반환도 하도록 작성 했을 때  (f1, f2가 각각 기약분수 (5/6, 1/2)로 바뀐 후 사칙연산을 수행하므로 다음과  같이 결과 출력.). |