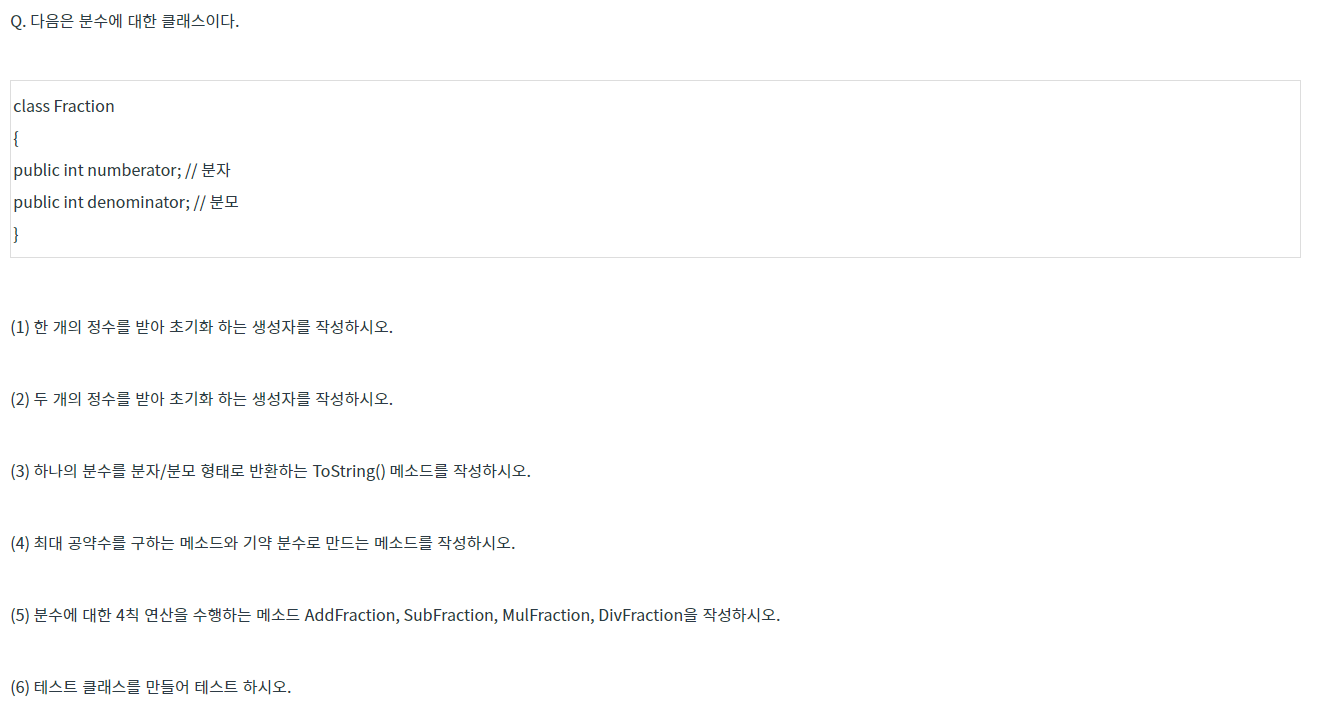
**윈도우즈 프로그래밍**

**컴퓨터공학과**

**20204062 이인규**

****

****

****

**소스코드**

|  |
| --- |
| // (1) 한 개의 정수를 받아 초기화 하는 생성자를 작성하시오.  public Fraction(int a){  this.numberator = a;  this.denominator = 1;  } |

**소스 코드 설명**

Fraction 클래스 내부에 있는 생성자이다. 멤버 변수로 numberator와 denominator 변수가 미리 선언되어 있으며, 정수 한 개를 입력받았을 경우 사용된다. 기존의 코드는 denominator의 값을 0으로 초기화하였는데, 이 값을 1로 바꾸어주었습니다. 따라서 다른 알고리즘 상에서 division error(0을 나누었을때 발생하는 오류)를 지울 수 있었습니다.

****

**소스 코드**

|  |
| --- |
| // (2) 두 개의 정수를 받아 초기화 하는 생성자를 작성하시오.  public Fraction(int a, int b){  this.numberator = a;  this.denominator = b;  } |

**소스 코드 설명**

Faction 클래스 내부에 있는 생성자이다. 멤버 변수로 numberator와 denominator 변수가 미리 선언되어 있으며, 정수 두 개를 입력받았을 경우 사용된다. Fraction 클래스의 생성자는 2개로 서로 메소드   
오버로딩되어 사용된다.

****

**소스 코드**

|  |
| --- |
| // (3) 하나의 분수를 분자/분모 형태로 반환하는 ToString() 메소드를 작성하시오.  public override string ToString(){  return "(" + this.numberator + " / " + this.denominator + ")";  } |

**소스 코드 설명**

Fraction 클래스 내부의 ToString 메소드이다. 이 메소드는 하나의   
분수를 분자/분모 형태로 String 반환을 해주는 메소드이다.

****

**소스 코드**

|  |
| --- |
| **// (4-1) 최대 공약수를 구하는 메소드를 작성하시오.**  **public int GratestCommonDivisor(int numberator, int denominator){**  **int min;**  **if(numberator > denominator){**  **min = denominator;**  **}**  **else{**  **min = numberator;**  **}**  **for(int i=1; i<=min; i++){**  **if(numberator % i == 0 && denominator % i == 0){**  **this.temp = i;**  **}**  **}**  **return this.temp;**  **}**  **// (4-2) 기약 분수로 만드는 메소드를 작성하시오.**  **public void irreducibleFraction(Fraction a){**  **Console.WriteLine("기약분수로 표현한 분수식: " + a.numberator / a.temp + " / " + a.denominator / a.temp);**  **}** |

**소스 코드 설명**

Fraction 클래스 내부의 GratestCommonDivisor 메소드와 irreducibleFraction 메소드이다. 각각 최대 공약수를 구하는 메소드와 기약 분수로 만드는 메소드이다. 최대 공약수를 구하는 방식은   
유클리드 호제법의 L = G \* a \* b 수식을 이용해서 구하였다. 기약분수로 만드는 메소드는 구했던 최소 공배수를 기존의 분수에 나누는   
방식을 사용했다.

****

**소스 코드**

|  |
| --- |
| // (5-1) [더하기] 분수에 대한 4칙 연산을 수행하는 메소드  public void addFraction(Fraction a, Fraction b){  int g = a.denominator \* b.denominator / this.temp;  int a\_upper = (g / a.denominator) \* a.numberator;  int b\_upper = (g / b.denominator) \* b.numberator;  Console.WriteLine("f1 + f2 = (" + (a\_upper + b\_upper) + " / " + g + ")");  }  // (5-2) [빼기] 분수에 대한 4칙 연산을 수행하는 메소드  public void subFraction(Fraction a, Fraction b){  int g = a.denominator \* b.denominator / this.temp;  int a\_upper = (g / a.denominator) \* a.numberator;  int b\_upper = (g / b.denominator) \* b.numberator;  Console.WriteLine("f1 - f2 = (" + (a\_upper - b\_upper) + " / " + g + ")");  }  // (5-3) [곱하기] 분수에 대한 4칙 연산을 수행하는 메소드  public void mulFraction(Fraction a, Fraction b){  Console.WriteLine("f1 \* f2 = (" + (a.numberator \* b.numberator) + " / " + (a.denominator \* b.denominator) + ")");  }  // (5-4) [나누기] 분수에 대한 4칙 연산을 수행하는 메소드  public void divFraction(Fraction a, Fraction b){  Console.WriteLine("f1 / f2 = (" + (a.numberator \* b.denominator) + " / " + (a.denominator \* b.numberator) + ")");  } |

**소스 코드 설명**

각 분수를 더하고, 빼고, 곱하고, 나누는 사칙 연산 메소드를   
작성하였다. 더하고 뺄 때에는 기존에 유클리드 호제법으로 구하였던 최소 공배수를 이용해 구하였고, 곱하고 나누는 사칙 연산은 값들을 그대로 곱하는 방식을 이용하여 구하였다.

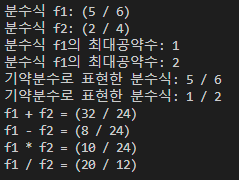
****

|  |
| --- |
| class Program{  static void Main(String[] args){  Fraction f1, f2;  // case 1  f1 = new Fraction(5, 6);  f2 = new Fraction(2, 4);  // case 2  // f1 = new Fraction(5);  // f2 = new Fraction(2, 4);  Console.WriteLine("분수식 f1: " + f1.ToString());  Console.WriteLine("분수식 f2: " + f2.ToString());  Console.WriteLine("분수식 f1의 최대공약수: " + f1.GratestCommonDivisor(f1.numberator, f1.denominator));  Console.WriteLine("분수식 f1의 최대공약수: " + f2.GratestCommonDivisor(f2.numberator, f2.denominator));  f1.irreducibleFraction(f1);  f2.irreducibleFraction(f2);  f1.addFraction(f1, f2);  f1.subFraction(f1, f2);  f1.mulFraction(f1, f2);  f1.divFraction(f1, f2);  }  } |

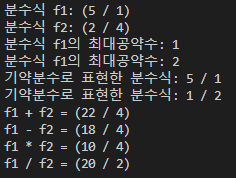
**소스 코드 설명**

테스트 클래스를 만들어서 만들었던 모든 메소드를 실행하였다.   
테스트 케이스는 2개를 작성하였다. 5/6과 2/4 연산, 5와 2/4 연산을 하였다.

**테스트 케이스 1 (5/6과 2/4 연산) 실행 결과**

****

**테스트 케이스 2 (5와 2/4 연산) 실행 결과**

****

**전체 소스 코드**

|  |
| --- |
| class Fraction{  public int numberator;  public int denominator;  public int temp; // 기약분수  // (1) 한 개의 정수를 받아 초기화 하는 생성자를 작성하시오.  public Fraction(int a){  this.numberator = a;  this.denominator = 1;  }  // (2) 두 개의 정수를 받아 초기화 하는 생성자를 작성하시오.  public Fraction(int a, int b){  this.numberator = a;  this.denominator = b;  }  // (3) 하나의 분수를 분자/분모 형태로 반환하는 ToString() 메소드를 작성하시오.  public override string ToString(){  return "(" + this.numberator + " / " + this.denominator + ")";  }  // (4-1) 최대 공약수를 구하는 메소드를 작성하시오.  public int GratestCommonDivisor(int numberator, int denominator){  int min;  if(numberator > denominator){  min = denominator;  }  else{  min = numberator;  }  for(int i=1; i<=min; i++){  if(numberator % i == 0 && denominator % i == 0){  this.temp = i;  }  }  return this.temp;  }  // (4-2) 기약 분수로 만드는 메소드를 작성하시오.  public void irreducibleFraction(Fraction a){  Console.WriteLine("기약분수로 표현한 분수식: " + a.numberator / a.temp + " / " + a.denominator / a.temp);  }  // (5-1) [더하기] 분수에 대한 4칙 연산을 수행하는 메소드  public void addFraction(Fraction a, Fraction b){  int g = a.denominator \* b.denominator / this.temp;  int a\_upper = (g / a.denominator) \* a.numberator;  int b\_upper = (g / b.denominator) \* b.numberator;  Console.WriteLine("f1 + f2 = (" + (a\_upper + b\_upper) + " / " + g + ")");  }  // (5-2) [빼기] 분수에 대한 4칙 연산을 수행하는 메소드  public void subFraction(Fraction a, Fraction b){  int g = a.denominator \* b.denominator / this.temp;  int a\_upper = (g / a.denominator) \* a.numberator;  int b\_upper = (g / b.denominator) \* b.numberator;  Console.WriteLine("f1 - f2 = (" + (a\_upper - b\_upper) + " / " + g + ")");  }  // (5-3) [곱하기] 분수에 대한 4칙 연산을 수행하는 메소드  public void mulFraction(Fraction a, Fraction b){  Console.WriteLine("f1 \* f2 = (" + (a.numberator \* b.numberator) + " / " + (a.denominator \* b.denominator) + ")");  }  // (5-4) [나누기] 분수에 대한 4칙 연산을 수행하는 메소드  public void divFraction(Fraction a, Fraction b){  Console.WriteLine("f1 / f2 = (" + (a.numberator \* b.denominator) + " / " + (a.denominator \* b.numberator) + ")");  }  }  // (6) 테스트 클래스를 만들어 테스트 하시오.  class Program{  static void Main(String[] args){  Fraction f1, f2;  // case 1  // f1 = new Fraction(5, 6);  // f2 = new Fraction(2, 4);  // case 2  f1 = new Fraction(5);  f2 = new Fraction(2, 4);  Console.WriteLine("분수식 f1: " + f1.ToString());  Console.WriteLine("분수식 f2: " + f2.ToString());  Console.WriteLine("분수식 f1의 최대공약수: " + f1.GratestCommonDivisor(f1.numberator, f1.denominator));  Console.WriteLine("분수식 f1의 최대공약수: " + f2.GratestCommonDivisor(f2.numberator, f2.denominator));  f1.irreducibleFraction(f1);  f2.irreducibleFraction(f2);  f1.addFraction(f1, f2);  f1.subFraction(f1, f2);  f1.mulFraction(f1, f2);  f1.divFraction(f1, f2);  }  } |

**-보고서 끝-**