|  |
| --- |
| 윈도우즈 프로그래밍 |

-과제-

6주차 : 인덱서 실습

20144638

이한범

# 인덱서 실습

인덱서를 이용하여 1부터 100까지 홀수값을 저장한 객체와 1부터 100까지 짝수값을 저장한 객체를 만들고 출력한다. 그리고 각 객체에서 소수(prime number)만 새로운 객체에 추가한다.

1. 작성한 코드내용

# IndexerApp class (main)

|  |
| --- |
| class IndexerApp  {  public static void Main()  {  Num odd = new Num();  Num even = new Num();  Num prime = new Num();  int co = 0; //소수를 저장할 인덱서를 위한 오프셋 값  //인덱서 값을 설정자를 통해 정의  for (int i = 0; i < 50; i++)  {  odd[i] = 2 \* i + 1; //1,3,5,7,9...  even[i] = 2 \* i + 2; //2,4,6,8,10...  }  Console.WriteLine("객체에 저장된 소수 : ");  for (int i = 0; i < 50; i++)  {  int o\_count = 0;//홀수 소수 카운트  int e\_count = 0;//짝수 소수 카운트  for (int j = 2; j <= 2 \* i + 1; j++)  {  if (odd[i] % j == 0) o\_count++;  }  for (int j = 2; j <= 2 \* i + 2; j++)  {  if (even[i] % j == 0) e\_count++;  }  if (o\_count == 1)  Console.Write((prime[co++] = odd[i]) + " ");  if (e\_count == 1)  Console.Write((prime[co++] = even[i]) + " ");  }  }  } |

▲ 메인 클래스. 세 개의 객체를 만드는데, odd는 1~100까지의 홀수값을 저장. even은 1~100까지의 짝수값을 저장. prime은 위의 odd, even내의 소수값을 저장.

따라서 for문을 통해 짝수, 홀수값을 저장한 객체내의 배열에서 소수값만 빼내어 prime 객체내의 인덱스에 저장.

# Num class

|  |
| --- |
| class Num  {  private int[] num = new int[50];  //인덱서 정의  public int this[int index]  {  get { return num[index]; }  set { num[index] = value; }  }  } |

▲Num 클래스 정의. num 배열의 인덱서를 정의하여 메인클래스에서 보았듯, prime[1] 처럼 값을 객체 내 배열값을 가지는 필드에 접근 및 설정이 가능하다.

2) 실행결과

|  |
| --- |
|  |