프로그래밍 언어론

Report #6

2017.05.10

20154073

강다현

Q. C, C++, Java, C#의 열거 타입을 다음 항목들의 관점에서 비교하라. 차이점을 보여주는 각 언어들의 코드 예제를 들면서 설명하라.(코드를 실행시킬 것)

1 한 열거 상수가 한 개 이상의 타입 정의에 올 수 있는가?  
 (그럴 경우에, 그러한 상수 참조에 대한 타입 검사가 어떻게 이루어지는가?)

🏵 C언어

#include <stdio.h>

enum kakao { ryan, neo, frodo, tube };

enum summer { beach, ball, tube, sunglass };

int main() {

printf("tube = %d", tube);

return 0;

}



C언어는 열거체를 구분할 수 없으므로 열거 상수가 재정의될 경우 구분할 수 없음

🏵 C++

#include <iostream>

enum kakao { ryan, neo, frodo, tube };

enum summer { beach, ball, tube, sunglass };

int main() {

std::cout << "tube = %d" << tube ;

return 0;

}



C++도 C와 마찬가지로 열거체를 구분하여 사용하지 않음

🏵C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

enum kakao { ryan, neo, frodo, tube };

enum summer { beach, ball, tube, sunglass };

static void Main(string[] args)

{

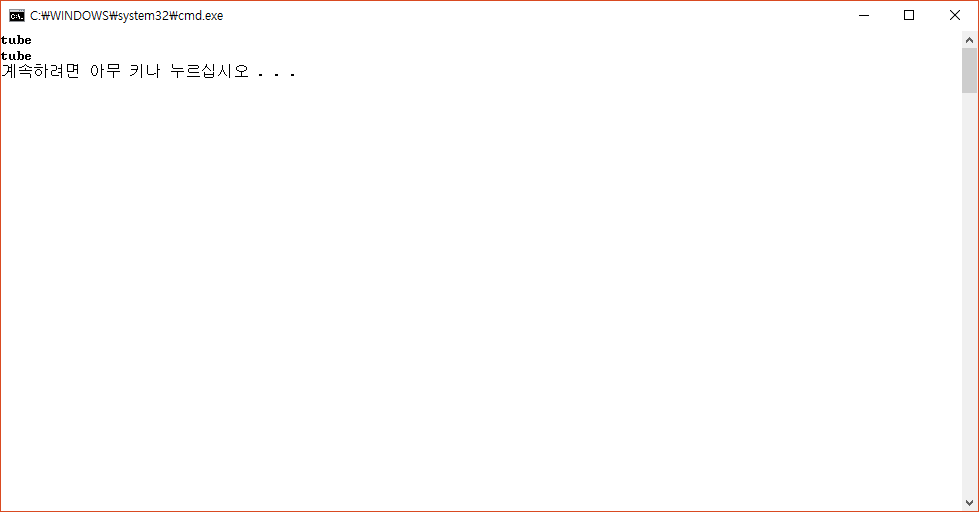
Console.WriteLine(kakao.tube);

Console.WriteLine(summer.tube);

}

}

}



C#은 한정자를 사용하여 열거형을 지정하면 열거 상수를 복수정의 가능하다.

🏵 java

**public** **class** pl6 {

**enum** kakao { ***ryan***, ***neo***, ***prodo***, ***tube*** }

**enum** summer { ***beach***, ***ball***, ***tube***}

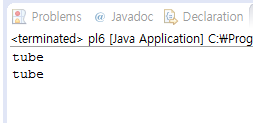
**public** **static** **void** main(String[] args) {

System.***out***.println(kakao.***tube***);

System.***out***.println(summer.***tube***);

}

}



자바에서는 열거체를 지정하여 출력할 수 있으므로 열거 상수를 중복정의 가능

2 열거 타입이 정수로 강제 변환 가능한가?

🏵 C

#include <stdio.h>

enum kakao { ryan, neo, frodo, tube };

int main() {

int i = 10;

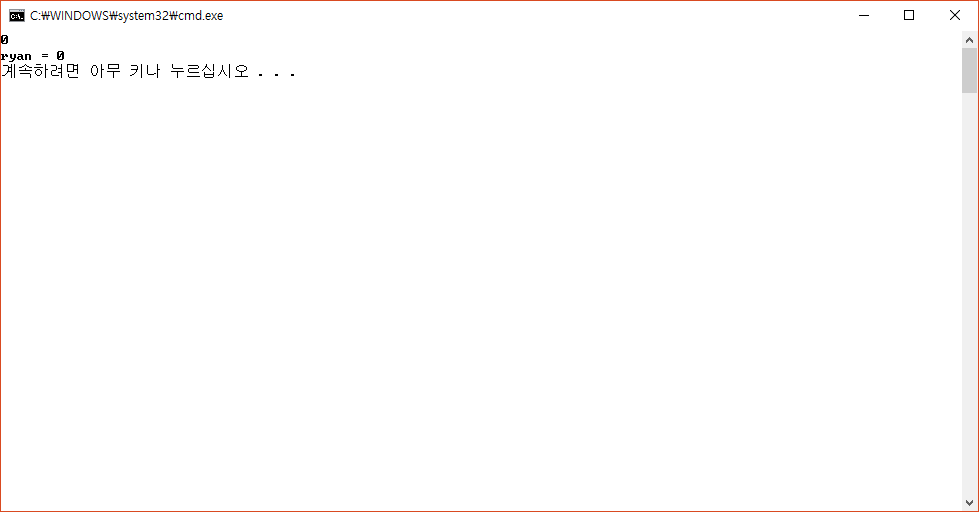
i = ryan;

printf("%d\n", i);

printf("ryan = %d\n", ryan);

return 0;

}



C언어에서 정수형에 열거 상수를 대입, 서식 문자를 정수형으로 하여 열거 상수를 출력 가능

🏵 C++

#include <iostream>

enum kakao { ryan, neo, frodo, tube };

int main() {

enum kakao freinds = neo;

int i = 20;

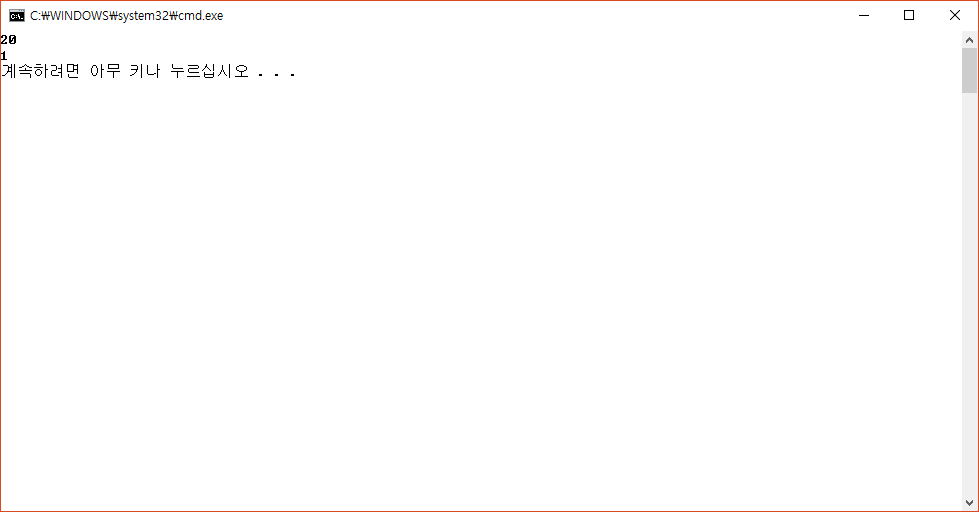
std::cout << i << std::endl;

i = freinds;

std::cout << i << std::endl;

return 0;

}



정수형 변수에 열거 상수 할당 시 값이 변경되어 출력

🏵 C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

enum kakao { ryan, neo, frodo, tube };

static void Main(string[] args)

{

int i = 10;

Console.WriteLine(i);

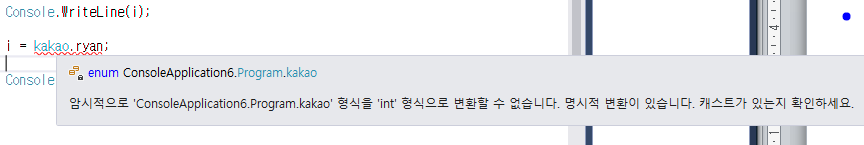
i = kakao.ryan;

Console.WriteLine(i);

}

}

}



암시적으로 정수로 강제 변환은 불가

🏵 java

**public** **class** pl6 {

**enum** kakao { ***ryan***, ***neo***, ***prodo***, ***tube*** }

**public** **static** **void** main(String[] args) {

**int** i = 10;

System.***out***.println(i);

i = kakao.***ryan***;

System.***out***.println(i);

}

}



정수형 변수에 열거 상수를 할당 시 열거 상수를 정수로 변환할 수 없다는 경고 메시지 표시

3 다른 타입이 열거 타입으로 강제 변환 가능한가?

🏵C

#include <stdio.h>

enum kakao { ryan, neo, frodo, tube };

int main() {

enum kakao freinds = neo;

int i = 10;

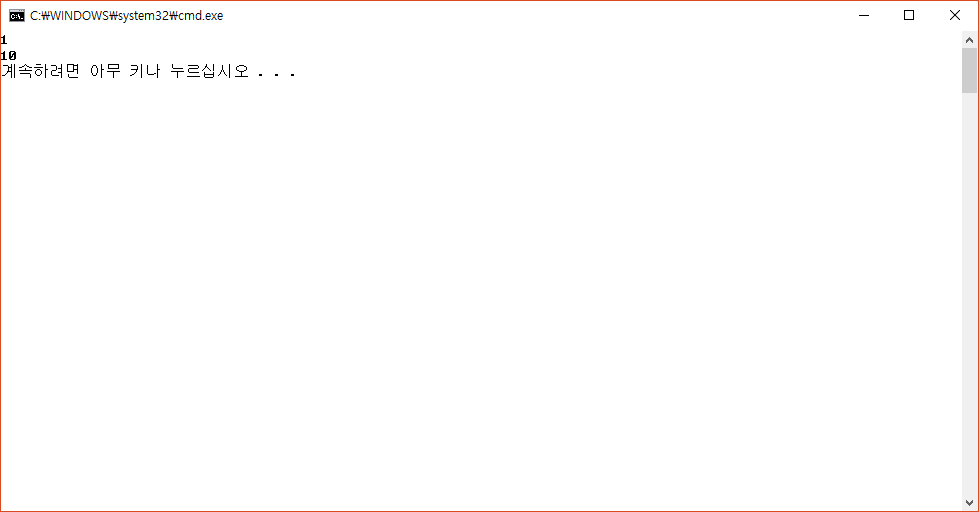
printf("%d\n", freinds);

freinds = i;

printf("%d\n", freinds);

return 0;

}



열거형 변수 friends에 정수형 변수 i를 대입 가능

🏵 C++

#include <iostream>

enum kakao { ryan, neo, frodo, tube };

int main() {

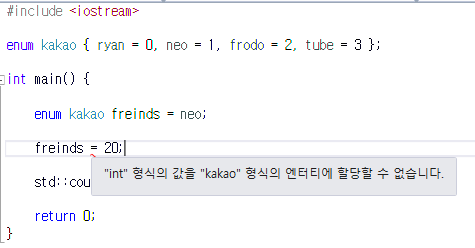
enum kakao freinds = neo;

freinds = 20;

std::cout << freinds << std::endl;;

return 0;

}



열거 타입에 정수를 할당할 수 없음

🏵 C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

enum kakao { ryan, neo, frodo, tube };

static void Main(string[] args)

{

kakao freinds = kakao.ryan;

freinds = 20;

Console.WriteLine(freinds);

}

}

}



정수형 타입을 암시적으로 열거 타입으로 강제 변환 불가

🏵 java

**public** **class** pl6 {

**enum** kakao { ***ryan***, ***neo***, ***prodo***, ***tube*** }

**public** **static** **void** main(String[] args) {

kakao freinds = kakao.***tube***;

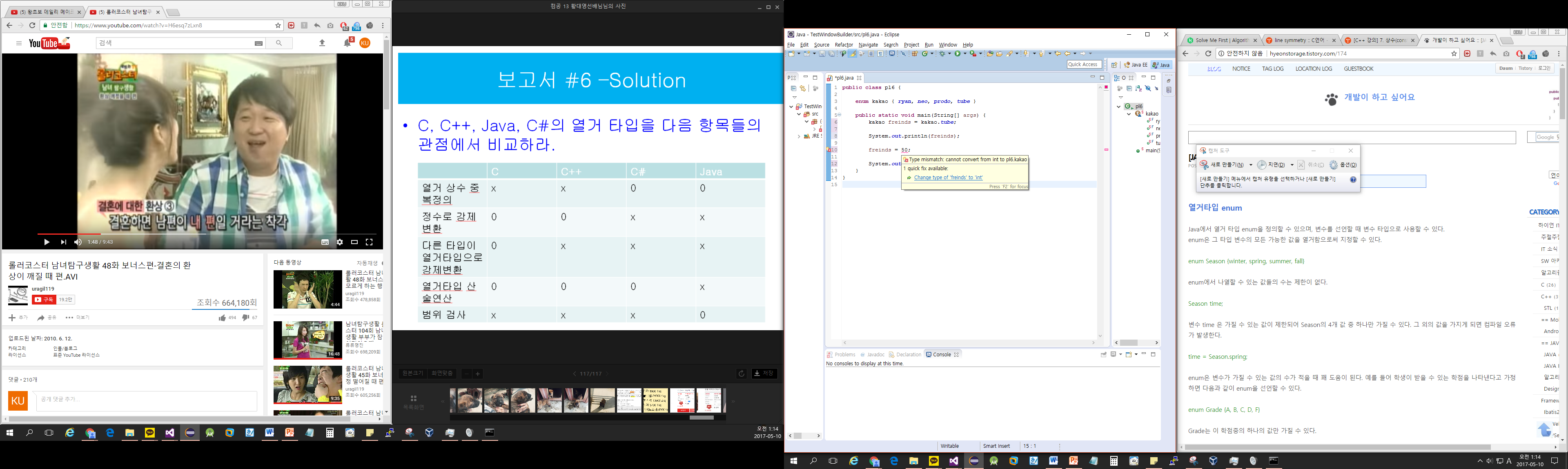
System.***out***.println(freinds);

freinds = 50;

System.***out***.println(freinds);

}

}



정수형 타입의 값을 열거 타입 변수에 할당할 경우 정수 타입에서 열거 타입으로 변환할 수 없다는 메시지가 표시

4 열거 타입에 대한 산술 연산이 가능한가?

🏵 C

#include <stdio.h>

enum kakao { ryan, neo, frodo, tube };

int main() {

printf("%.1lf\n", neo+1.5);

return 0;

}



열거 상수에 상수와 연산 가능

🏵 C++

#include <iostream>

enum kakao { ryan, neo, frodo, tube };

int main() {

std::cout << neo+1 << std::endl;

return 0;

}



열거 상수에 상수 연산 가능

🏵 C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

enum kakao { ryan, neo, frodo, tube };

static void Main(string[] args)

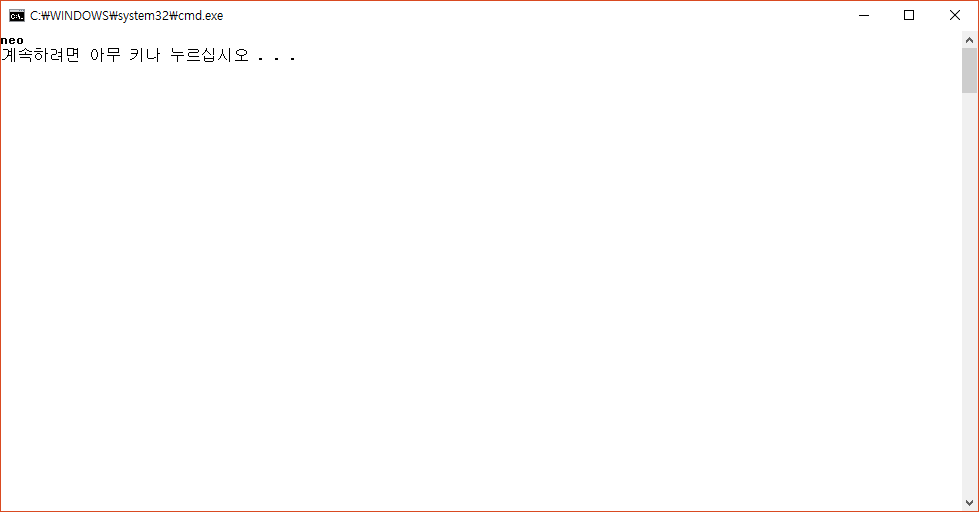
{

Console.WriteLine(kakao.ryan+1);

}

}

}



열거 상수에 1을 증가시켜 출력하면, 다음 열거 상수가 출력됨

🏵 java

**public** **class** pl6 {

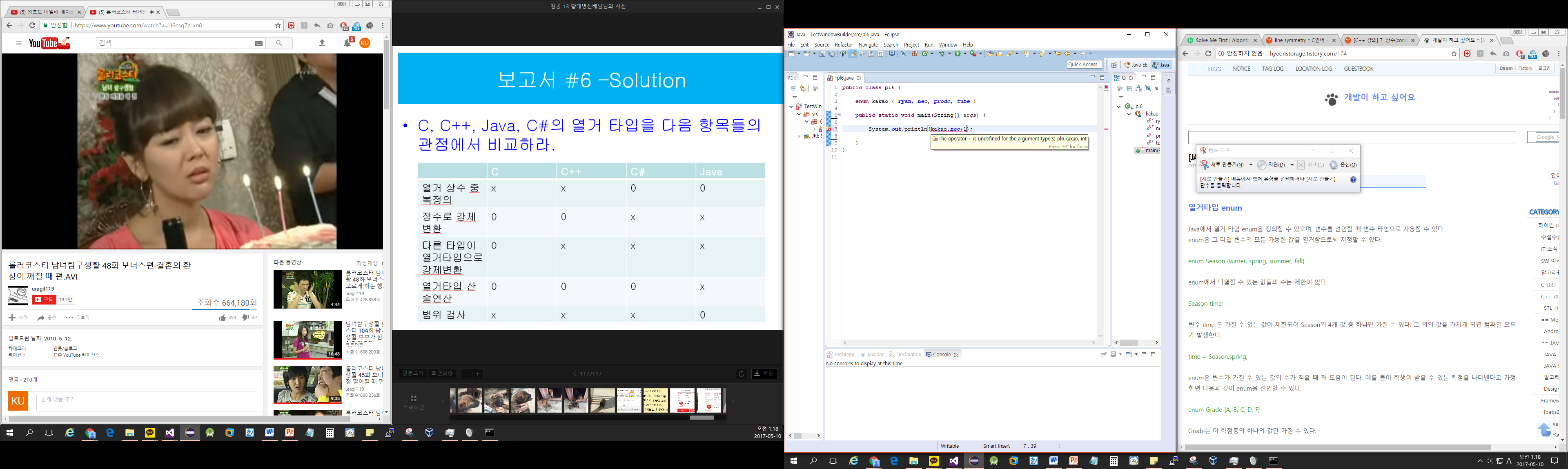
**enum** kakao { ***ryan***, ***neo***, ***prodo***, ***tube*** }

**public** **static** **void** main(String[] args) {

System.***out***.println(kakao.***neo***+1);

}

}



열거타입 상수에 상수 연산을 한 결과 연산자 정의가 되어있지 않다는 메시지가 표시되고, 자바는 연산자 오버라이딩이 불가하므로 열거타입의 산술 연산 불가

5 열거 타입 변수에 범위를 벗어난 값을 배정 가능한가?

🏵 C

#include <stdio.h>

enum kakao { ryan, neo, frodo, tube };

int main() {

enum kakao freinds = (enum kakao)4;

printf("%d\n", freinds);

return 0;

}

  
열거 타입 변수에 범위를 벗어난 값을 형변환하여 열거 타입 변수에 할당할 경우 배정 가능

🏵 C++

#include <iostream>

enum kakao { ryan, neo, frodo, tube };

int main() {

enum kakao freinds = neo;

freinds = (kakao)5;

std::cout << freinds << std::endl;

return 0;

}



열거 타입 변수에 범위를 벗어난 값을 형변환하여 할당한 결과 할당된 값이 출력. 범위를 벗어난 값 배정 가능

🏵 C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApplication6

{

class Program

{

enum kakao { ryan, neo, frodo, tube };

static void Main(string[] args)

{

kakao freinds = (kakao)6;

Console.WriteLine(freinds);

}

}

}



범위를 벗어난 값을 열거 타입으로 형변환하여 열거 타입 변수에 할당한 결과 범위를 벗어난 값이 출력

🏵 java

**public** **class** pl6 {

**enum** kakao { ***ryan***, ***neo***, ***prodo***, ***tube*** }

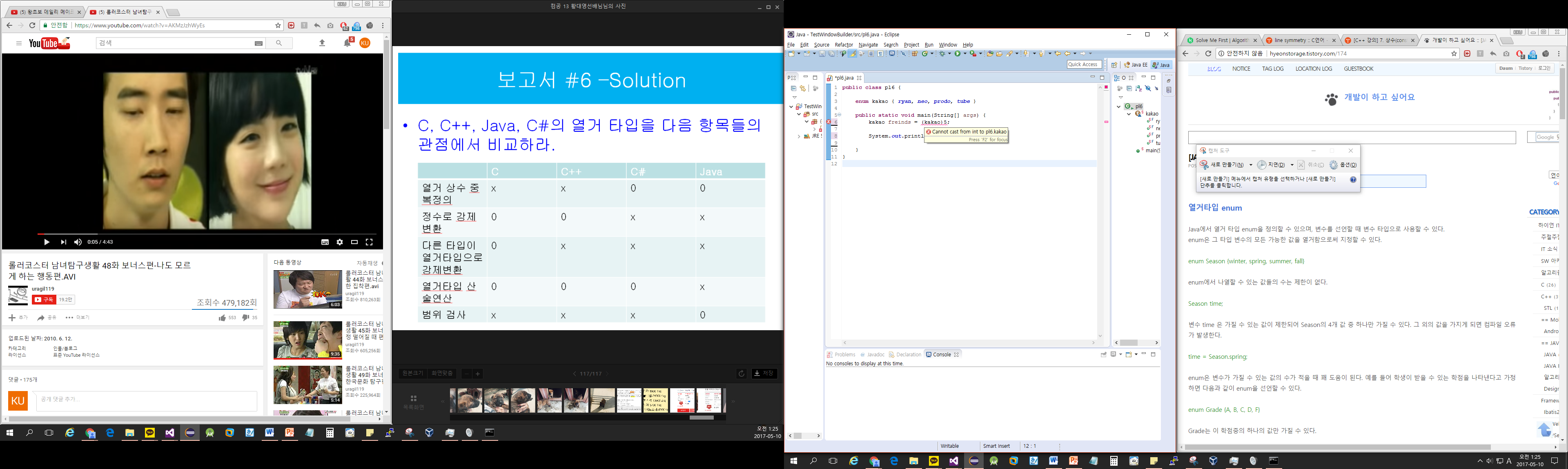
**public** **static** **void** main(String[] args) {

kakao freinds = (kakao)5;

System.***out***.println(freinds);

}

}



자바의 열거타입 변수는 정수형으로 캐스트가 되지 않으므로 범위를 벗어난 값을 배정할 수 없다.

🖉의견

이번 과제를 하면서 오랜만에 여러 언어를 사용하여 복습도 하면서 몇 언어에서 배우지 않았던 열거체를 스스로 공부하고, 사용해 보는 좋은 계기가 된 것02 같습니다. 감사합니다 ☺