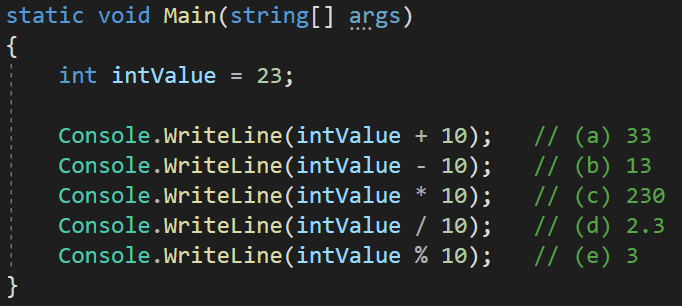
**C# 프로그래밍**

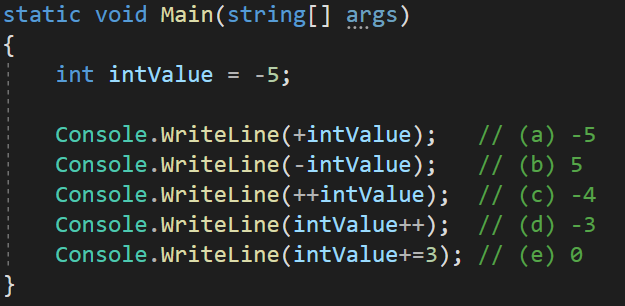
|  |  |  |  |
| --- | --- | --- | --- |
| 과정명 | 멀티 디바이스 메타버스 플랫폼 게임 개발자 과정 | | |
| 이름 | 임준희 | 평가일자 | 2024/01/08 |

* 해당 테스트는 구현능력보다 프로그래밍에 대한 내용들에 대한 기초지식을 요구합니다.
* 문제를 풀어보고 모르는 부분에 대해서 다시 학습하기 위한 용도로 사용합니다.
* 모르는 문제가 있다면 반드시 다시 학습하는 것을 권장합니다.

1. 다음 중 C#의 자료형에 대한 내용 중 틀린 답을 모두 고르시오.
   1. int는 정수형 자료형이며 4byte의 크기를 가진다.
   2. bool은 논리형 자료형이며 1bit의 크기를 가진다.
   3. float는 실수형 자료형이며 4byte의 크기를 가진다.
   4. double은 실수형 자료형이며 8byte의 크기를 가진다.
   5. char는 문자형 자료형이며 1byte의 크기를 가진다.
2. 다음 중 C#의 변수에 대한 설명으로 틀린 답을 고르시오.
   1. 변수를 선언하기 위해 자료형과 변수명이 필요하다.
   2. 변수에 처음으로 데이터를 저장하는 것을 초기화라 하며, 초기화 전까지는 값을 불러올 수 없다.
   3. 변수명은 변수들을 구분하기 위한 이름을 선정하는 것으로 의미가 같은 경우 중복해서 사용한다.
   4. 변수에 데이터를 저장하기 위해서는 변수를 =(대입연산자) 좌측에 배치한다.
   5. 변수의 데이터를 불러오기 위해서는 데이터가 필요한 곳에 변수명을 배치한다.
3. 다음 중 C#에서 허용하지 않은 문법을 고르시오.
   1. int intValue = 10;
   2. char charValue = ‘a’;
   3. float floatValue = 20.3;
   4. double doubleValue = 10.4;
   5. bool boolValue = true;
4. 다음 중 C#의 형변환으로 적합하지 않은 것을 고르시오.
   1. int intValue = (int)1.4;
   2. float floatValue = (float)”12.3”;
   3. char charValue = (char)79;
   4. int intValue = (int)’a’;
   5. double doubleValue = 3.14f;
5. 다음 코드와 출력이 틀린 것을 고르시오.

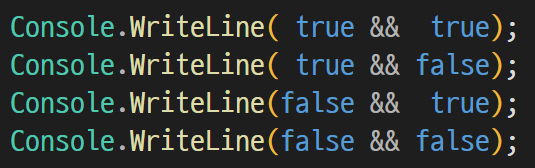
  
e d

1. 다음 코드와 출력이 틀린 것을 고르시오.

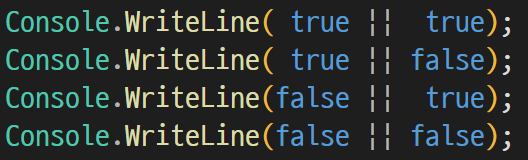


d

1. 다음 비교연산자의 설명으로 옳지 않은 것을 고르시오.
   1. 비교연산자 = 은 좌측값과 우측값이 같을 경우 결과가 true이다.
   2. 비교연산자 != 은 좌측값과 우측값이 다를 경우 결과가 true이다.
   3. 비교연산자 > 은 좌측값이 우측값보다 클 경우 결과가 true이다.
   4. 비교연산자 < 은 좌측값이 우측값보다 작을 경우 결과가 true이다.
2. 다음 코드의 출력을 적으시오.

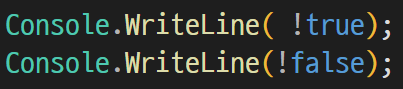
  
true, false, false, false

1. 다음 코드의 출력을 적으시오.



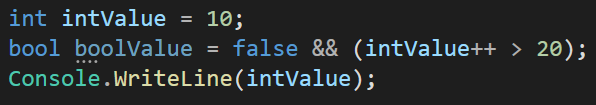
true, true, true, false

1. 다음 코드의 출력을 적으시오.



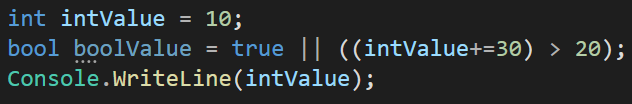
false, true

1. 다음 코드의 출력을 적으시오.



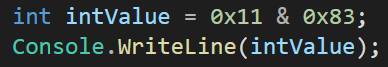
false 10 (앞에 false때문에 뒤의 계산이 무시되고 바로 진행됨)

1. 다음 코드의 출력을 적으시오.



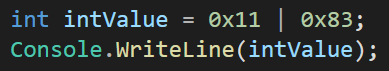
true 10 (앞에 true때문에 뒤의 계산이 무시되고 바로 진행됨)

1. 다음 코드의 출력을 적으시오. (16진수로 표현해도 정답 인정)



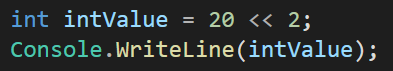
0000 0001

1. 다음 코드의 출력을 적으시오. (16진수로 표현해도 정답 인정)



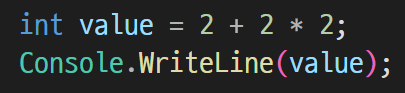
1001 0011

1. 다음 코드의 출력을 적으시오.



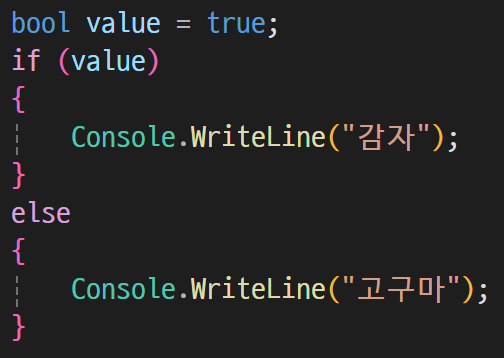
80

1. 다음 코드의 출력을 적으시오.



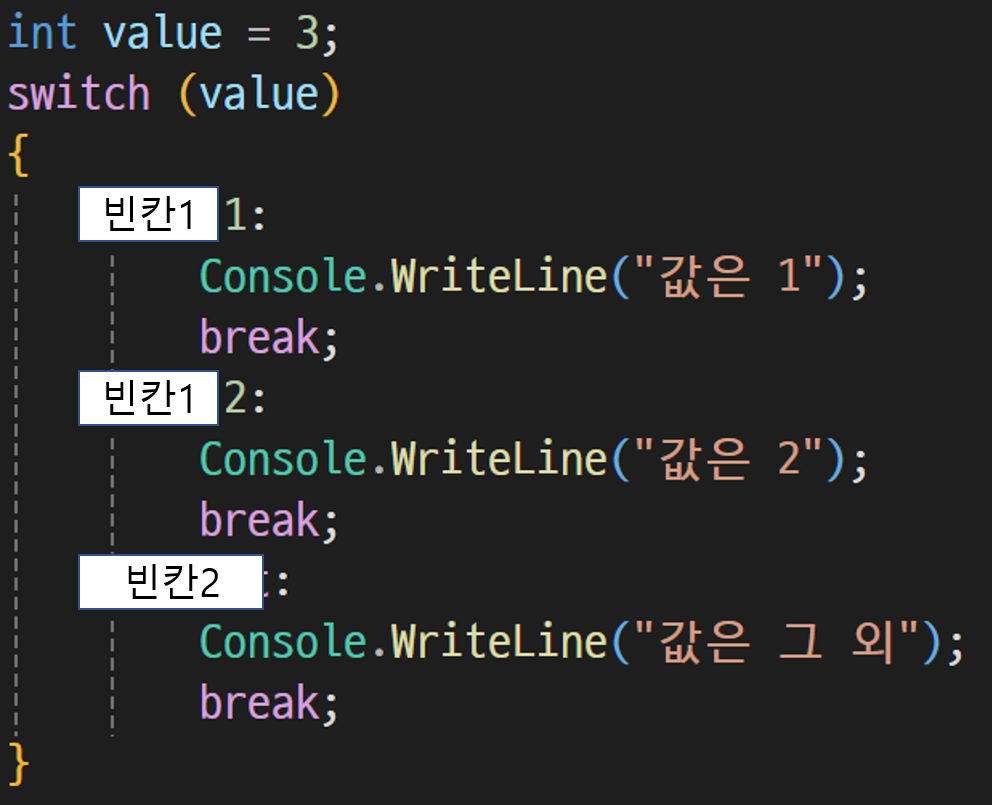
6

1. 다음 코드의 출력을 적으시오.



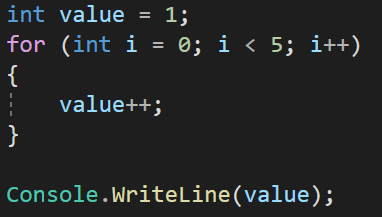
감자

1. 다음 코드의 빈칸으로 알맞은 내용을 적으시오.



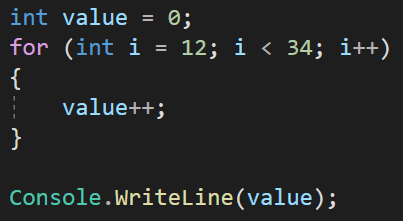
case, case, default

1. 다음 코드의 출력을 적으시오.



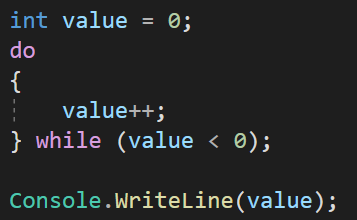
6

1. 다음 코드의 출력을 적으시오.



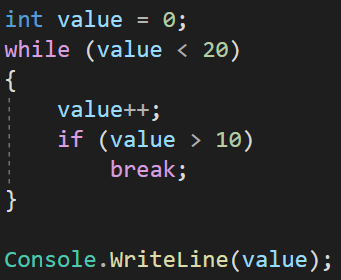
22

1. 다음 코드의 출력을 적으시오.



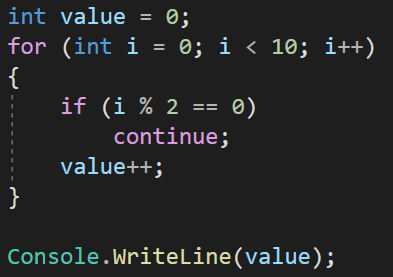
1

1. 다음 코드의 출력을 적으시오.



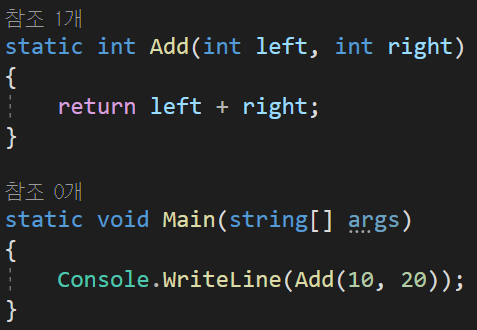
11

1. 다음 코드의 출력을 적으시오.



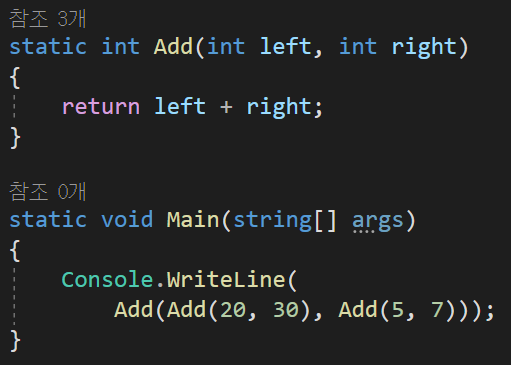
5

1. 다음 코드의 출력을 적으시오.



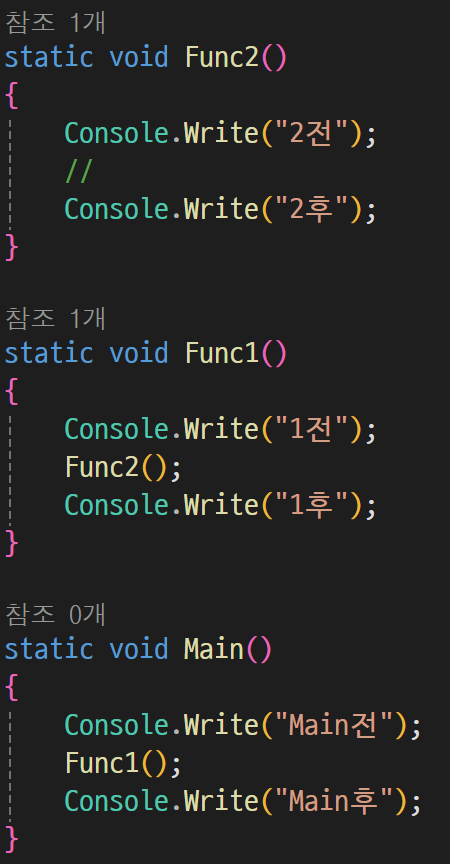
30

1. 다음 코드의 출력을 적으시오.



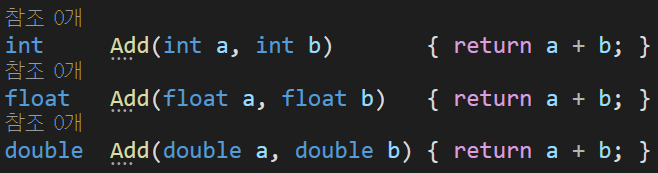
62

1. 다음 코드의 출력을 적으시오.



Main전1전2전2후1후Main후

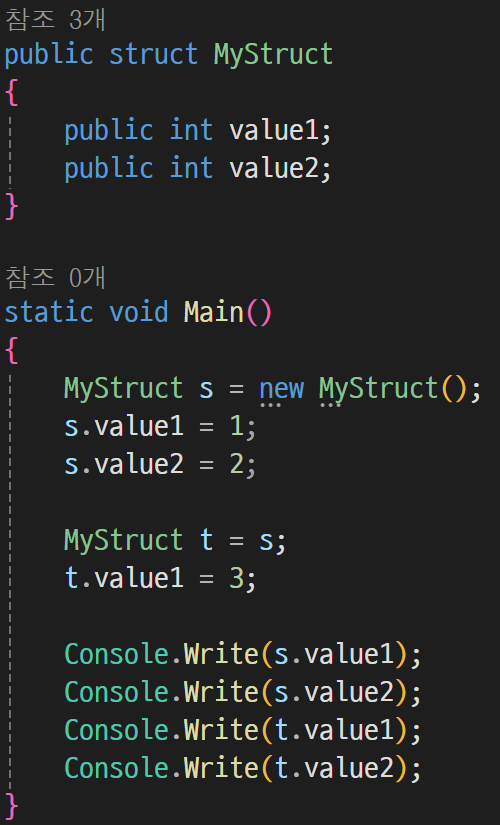
1. 아래의 코드를 보고 빈칸을 적으시오.



(빈칸) : 같은 이름의 함수를 여러개 가지면서 매개변수의 유형과 개수를 다르게 하는 기술

오버로딩

1. 다음 중 구조체에 대한 설명으로 옳지 않은 것을 고르시오.
   1. 구조체는 여러 데이터를 하나로 묶어 관리하는 자료형을 정의하는 방법이다.
   2. 구조체의 변수에 접근하고 싶을 때 . 을 통해 접근할 수 있다.
   3. 구조체 안의 반환형이 없는 구조체이름의 함수를 초기화 또는 생성자라고 한다.
   4. 구조체의 초기화를 매개변수를 달리하여 여러개 구성할 수 있다.
   5. 구조체는 여러 데이터의 집합이므로 데이터 관련 기능인 함수를 포함할 수 없다.
2. 다음 중 열거형에 대한 설명을 옳지 않은 것을 고르시오.
   1. 열거형은 이름으로 관리될 수 있어 코드의 가독성적인 측면에 도움이 된다.
   2. 열거형은 기본 정수 숫자의 명명된 상수 집합에 의해 정의되는 값 형식이다.
   3. 열거형의 멤버는 정수로 표현 가능하며 서로 형변환이 가능하다.
   4. 열거형의 멤버는 자동으로 지정되며 변경할 수 없다.
   5. 열거형의 멤버는 자동으로 0부터 시작하고 순서에 따라 1씩 증가한다.
3. 다음 코드의 출력을 적으시오.



1232

1. 다음 코드의 출력을 적으시오.

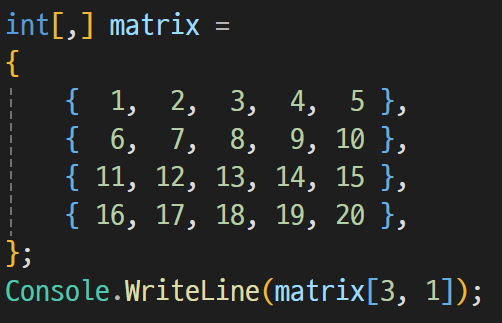


1

Winter

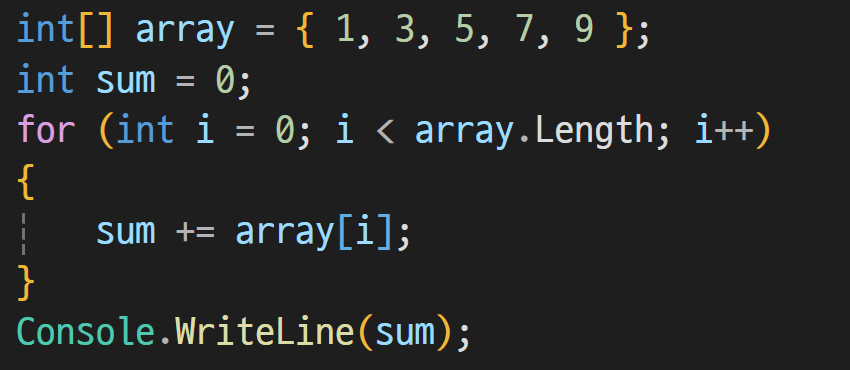
Spring 50

1. 다음 중 배열에 대한 설명으로 옳지 않은 것을 고르시오.
   1. 배열은 동일한 자료형의 요소들로 구성된 데이터 집합이다.
   2. 배열은 인덱스를 통하여 배열요소에 접근할 수 있다.
   3. 배열의 처음 요소의 인덱스는 1부터 시작한다.
   4. 크기가 n인 배열의 마지막 요소의 인덱스는 n-1이다.
   5. 배열이 갯수 뿐만 아니라 차원수를 늘려 관리가 필요한 경우 다차원 배열을 사용할 수 있다.
2. 아래의 코드의 출력을 적으시오.



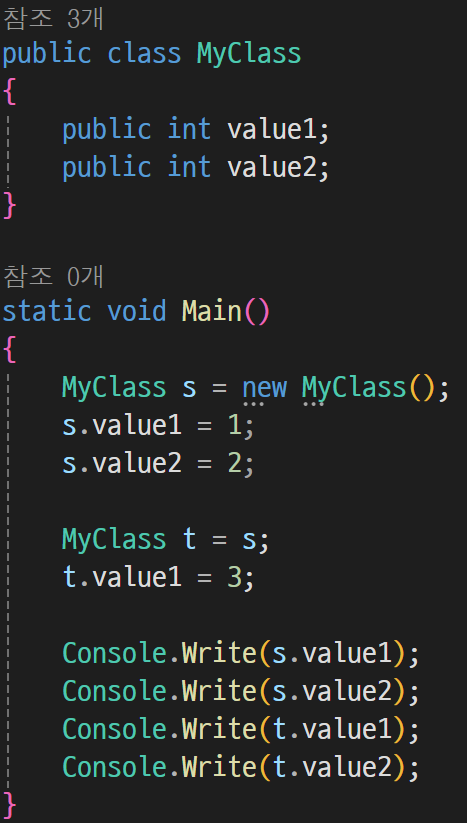
17

1. 다음 코드의 출력을 적으시오.



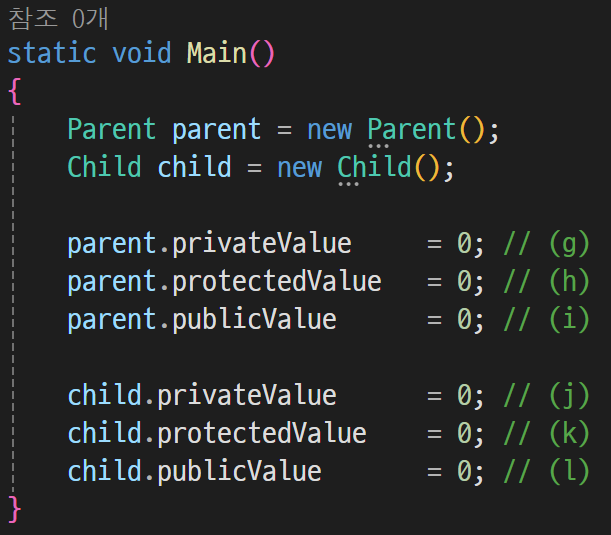
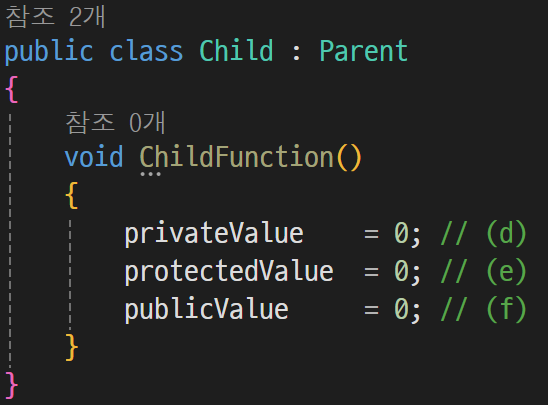
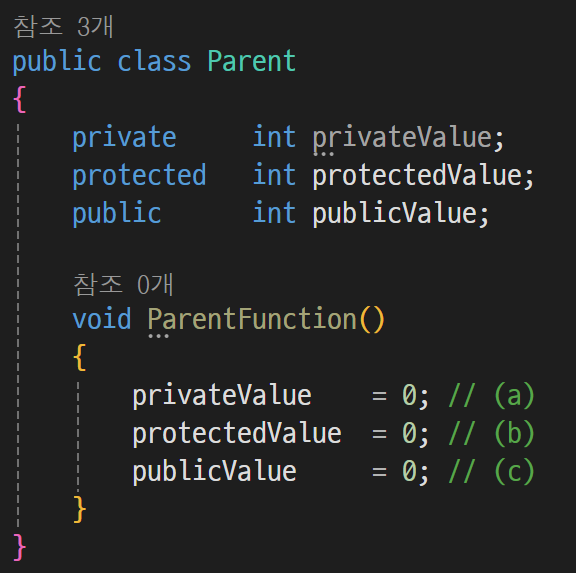
25

1. 다음 코드의 출력을 적으시오.



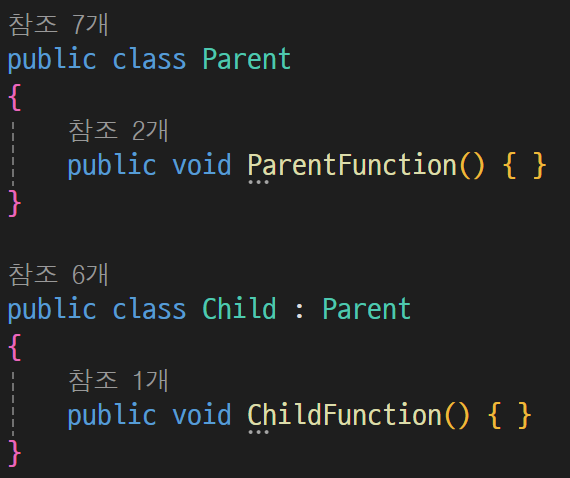
3232

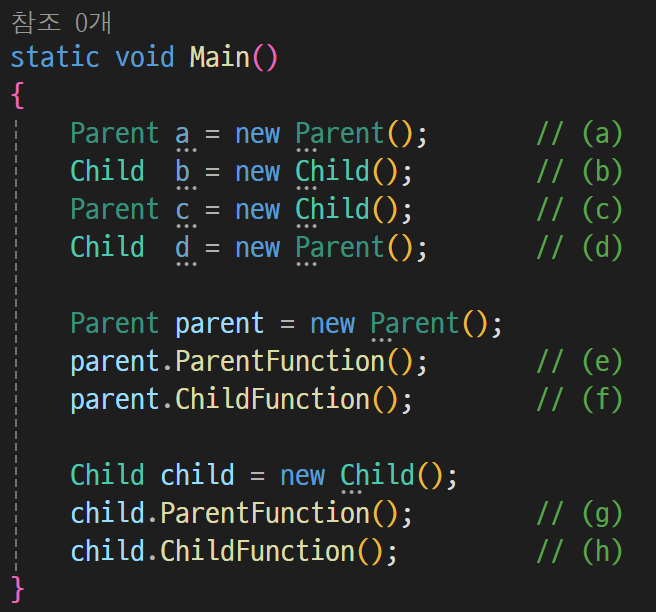
1. 다음 중 오류가 발생하는 코드를 모두 고르시오.



d,g,h,j,k

1. 다음 중 오류가 발생하는 코드를 모두 고르시오.





c,f d,f

(38~41) 아래의 코드를 보고 이어지는 문제의 답을 적으시오.

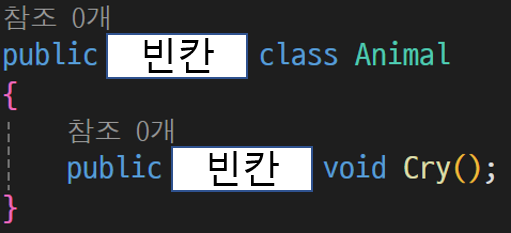




1. 문제 코드 자리에 다음 코드가 있을 경우, 코드의 출력을 적으시오.

 슬라임이 분열합니다.

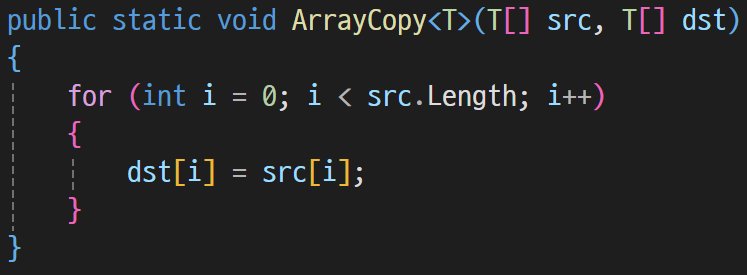
1. 문제 코드 자리에 다음 코드가 있을 경우, 코드의 출력을 적으시오.  
   드래곤이 분노합니다.
2. 문제 코드 자리에 다음 코드가 있을 경우, 코드의 출력을 적으시오.  
   슬라임이 데미지를 받습니다.
3. 문제 코드 자리에 다음 코드가 있을 경우, 코드의 출력을 적으시오.  
   드래곤이 분노합니다.
4. 다음 코드의 빈칸으로 알맞은 내용을 적으시오.



(빈칸) : 클래스를 정의할 당시 구체화 할 수 없는 기능의 표현을 정의하며, 상속한 클래스가 구체화하여 사용

abstruct

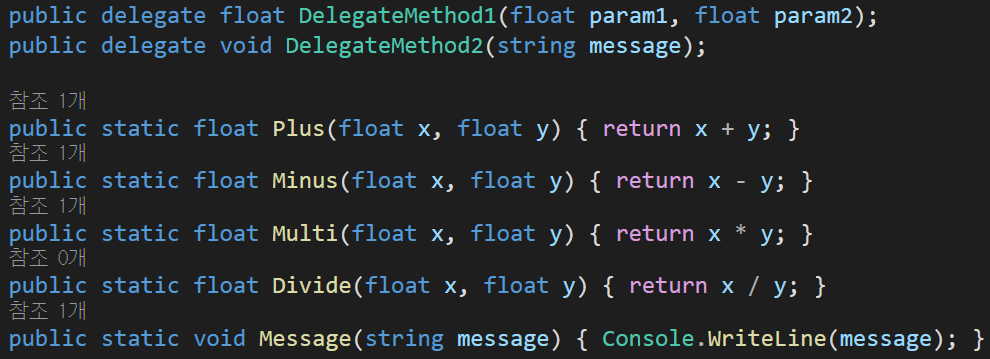
1. 다음 중 인터페이스에 대한 설명으로 옳지 않은 것을 고르시오.
   1. 인터페이스는 interface 키워드를 사용하여 정의한다.
   2. 인터페이스는 멤버변수를 포함할 수 없다.
   3. 인터페이스는 멤버함수를 포함하며, 직접 구현하지 않고 정의만 진행한다.
   4. 인터페이스를 포함하는 클래스는 인터페이스에서 정의한 함수를 반드시 구현해야 한다.
   5. 인터페이스는 상속과 동일하게 클래스마다 하나씩만 포함할 수 있다.
2. 아래의 코드를 보고 빈칸을 적으시오.

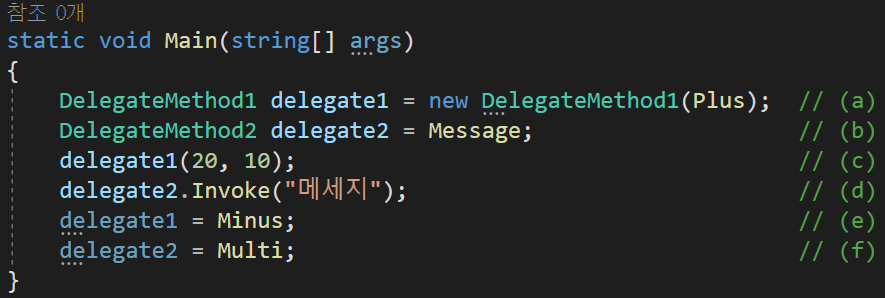


(빈칸) : 클래스 또는 함수가 코드에 의해 선언되고 인스턴스화될 때까지 형식의 사양을 연기하는 디자인

Genetic

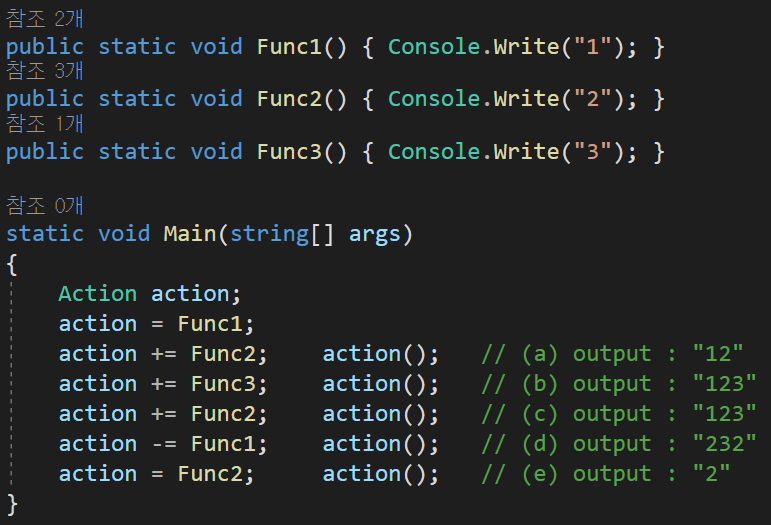
1. 다음 코드 중 오류가 발생하는 부분을 고르시오.





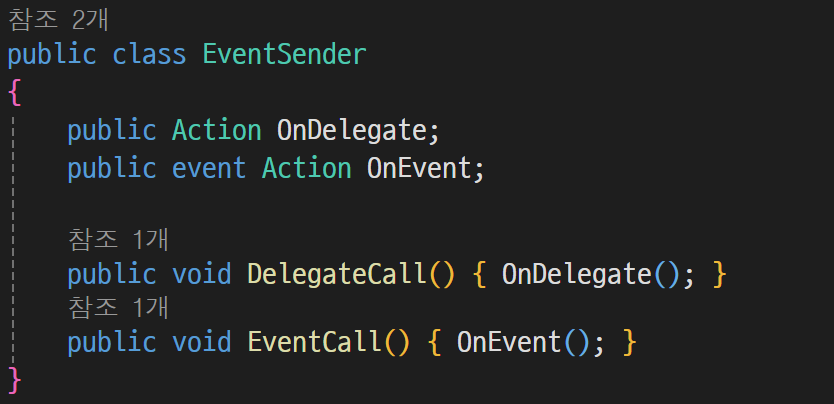
f

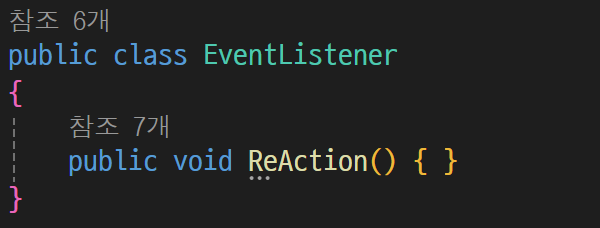
1. 다음 중 델리게이트에 대한 설명으로 옳지 않은 것을 고르시오.
   1. 델리게이트는 특정 매개 변수 목록 및 반환 형식이 있는 함수에 대한 참조이다.
   2. 델리게이트 인스턴스를 통해 함수를 참조하고 참조한 함수를 호출할 수 있다.
   3. 델리게이트 인스턴스는 오직 하나의 함수만 참조가 가능하며 대입연산자만 지원한다.
   4. 델리게이트는 지정자로 사용하여 미완성 상태의 함수를 매개변수로 전달하여 호출할 수 있다.
   5. 델리게이트 인스턴스는 무명메서드와 람다식 또한 참조할 수 있다.
2. 아래의 코드 중 출력이 틀린 곳을 고르시오.

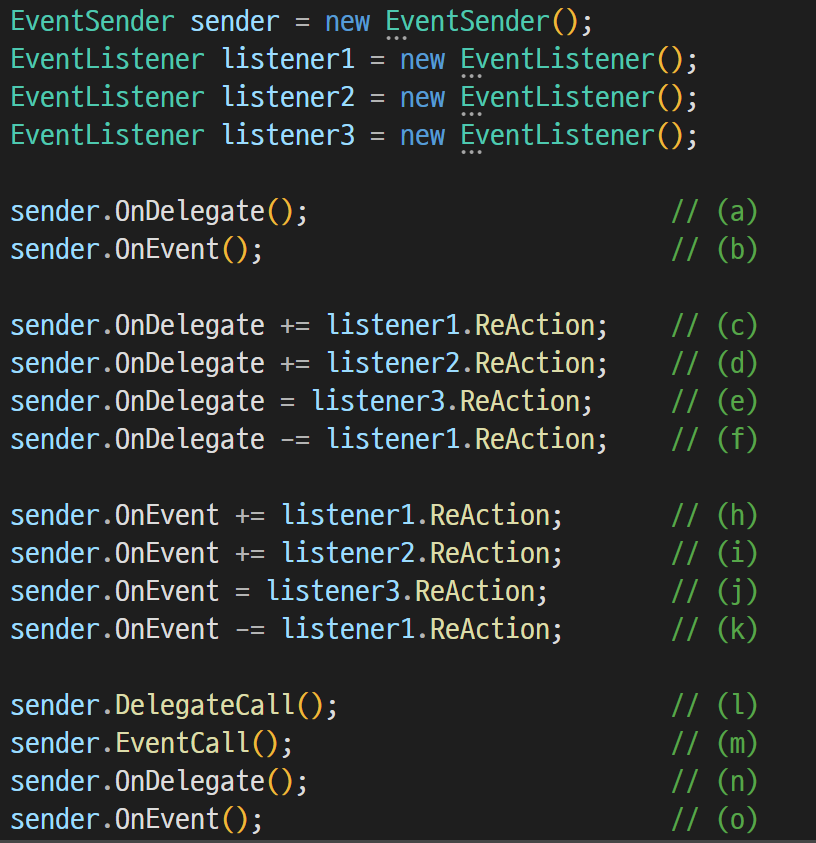


c

1. 다음 중 이벤트에 대한 설명으로 옳지 않은 것을 고르시오.
   1. 이벤트는 일련의 사건이 발생했다는 사실을 다른 객체에게 전달하는 용도로 사용한다.
   2. 이벤트는 델리게이트의 일종이며 델리게이트의 일부 기능을 제한하여 이벤트로 사용한다.
   3. 이벤트는 호출 순서를 보장해주지는 않으며, 사건 발생에 반응하는데 목적이 있다.
   4. 이벤트는 += 과 -= 을 통한 추가 할당과 할당 제거만이 가능하며 = 대입연산자는 제한되어 있다.
   5. 이벤트는 클래스 외부에서 발생이 가능하다.
2. 다음 중 오류가 발생하는 코드를 모두 고르시오. (런타임 오류도 포함)







a,b,j a,b,j,o

1. (서술형) 값형식과 참조형식에 대해 아는대로 설명하세요. (최대한 자세하게, 간략하게 작성하지 말것)

값형식는 실제 데이터가 가지고 있는 값을 말하며, 참조형식은 실제값이 들어있는 메모리 주소를 말한다. 값형식의 경우 깊은 복사를 통해 복사를 할 수 있으며, 참조형식은 얕은 복사로 메모리 주소를 복사하고 실제값이 들어있는 장소를 가리킨다. 값형식을 호출했을때는 값만 불러고 실제 값에대한 변환을 할 수 없는것에 비해 참조형식은 실제값이 들어있는 주소로 들어가 실제값을 수정할 수 있게 만들어 준다.

값형식 : 구조체, 깊은복사, 실제값을 저장

참조형식 : 클래스, 얕은복사, 주소값을 저장

1. (서술형) 객체지향 프로그래밍에 대해 아는대로 설명하세요. (최대한 자세하게, 간략하게 작성하지 말것)

객체지향은 순차지향과 같이 나열된 코드를 순서대로 읽고 처리하는 방식이 아닌 캡슐화된 객체들을 통하여 서로 유기적으로 코드를 읽어와 처리하는 방식을 말한다.

객체지향은 4가지 특징을 가지고 있는데 이는 캡슐화, 다형성, 상속, 추상화이다. 캡슐화는 외부에서의 접속을 관리하며 내부의 정보를 보호하는 것을 말한다. 다형성은 다양한 기능을 한곳에 모아두는 것이 아닌 다양한 곳에 만들어 두고 관리를 할 수 있는 기능을 말한다. 상속은 부모 구조체에서 자식 구조체로 변수 또는 함수를 상속을 할 수 있는 기능을 말한다. 추상화는 부모 구조체에서 당장 안만들거나 추상화를 시킴으로서 자식 구조체에서 이를 상속받아 정의 할 수 있는 기능을 말한다.

1. 왜 등장했는지. 절차지향 vs 객체지향
2. 객체란? 클래스란? 인스턴스란?
3. 장단점 : 소스의 재사용, 디버깅 용이, 대규모 프로젝트 적합 / 설계에 신중해야 함
4. 4특징 : 상속, 추상화, 캡슐화, 다형성
5. A+) 실제로 포트폴리오에서 객체지향 프로그래밍을 잘 구사한 부분을 보여주세요.
6. (서술형) 메모리 구조에 대해 아는대로 설명하세요. (최대한 자세하게, 간략하게 작성하지 말것)

메모리는 주로 코드 영역, 데이터 영역, 힙(heap)영역, 스텍(stack)영역으로 나누어져 있다.

코드 영역은 실행할 코드 전체의 내용을 저장해 두는 곳으로 처음 코드를 실행했을 때 모든 코드 내용이 여기에 저장된다.

데이터 영역은 정적으로 정의된 변수 밑 함수가 저장되는 곳으로 실행하는데 필요한 데이터 밑 코드를 저장하는 곳이며 모든 코드가 다 실행되었을 때 해제되는 구역이다.

힙 영역은 유동적인 크기를 가지는 데이터들을 주로 힙형식으로 관리 밑 저장을 하는 구역으로, 참조 형식으로 접근할 수 있으며, 더 이상 참조되지 않을 때 가비지 콜렉터에 의해서 주기적으로 삭제가 이루어지며, 정리가 되며 주로 new로 만들어진 string등 구초체에 해당되는 이들이 이에 저장되어 관리된다..

스텍 영역은 스택 형식, 즉 마지막에 들어온 순서대로 다시 빼내는 방식으로 작동되며 함수를 불러올 때마다 이전 함수 위에 쌓이고 이가 실행이 완료될 때마다 순차적을 삭제되는 방식으로 관리가 되며 주로 매개변수, 지역변수 등이 들어간다.

만약 스텍 영역이 힙 영역을 넘어갈경우 오버플로우라고 하고, 힙영역이 스텍 영역을 넘어갈 경우 언더플로우 라고 한다.

1. 코드영역, 데이터영역, 스택영역, 힙영역
2. 각각의 영역을 분리해서 사용하는 이유 -> 접근범위, 생존범위
3. 각각의 영역에 저장되는 데이터들 -> 소스코드, 정적변수, 지역변수, 매개변수, 인스턴스
4. 코드영역에 대한 설명
5. 데이터영역에 대한 설명
6. 스택영역에 대한 설명 : 함수 호출 스택
7. 힙영역에 대한 설명 : 가비지 컬렉터