



G-CAMP WORKSHOP

BUILD THE RIGHT THING
BUILD THING RIGHT
SHIP FAST

Lecturer Kim Myoung-Ho
Nickname 블스
blogstudy@naver.com

19강_예외처리

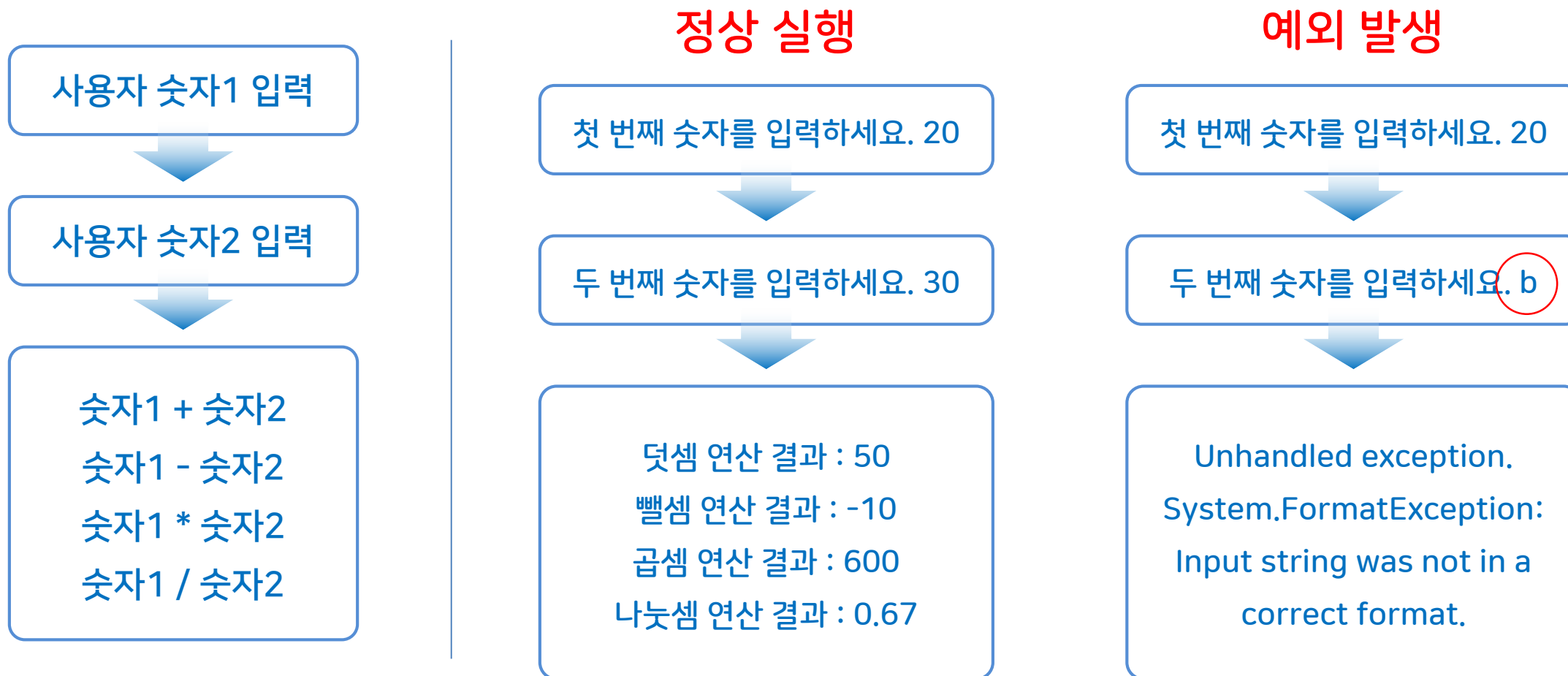
- 예외(Exception)란?
- 예외 처리(try ~ catch)
- 예외 던지기(throw) 및 반듯이 실행(finally)
- 사용자 예외(Custom Exception)



19-1. 예외(Exception)란?

ex) lec19/exceptionPjt

- Exception : 코드에는 문제가 없으나 외부 요인으로 발생한 오류





19-2. 예외 처리(try ~ catch)

- 예외 발생이 예상되는 코드를 try처리하고 catch에서 잡는다.

```
try  
{  
    예외 발생이 예상되는 코드  
}  
catch (Exception e)  
{  
    예외 처리  
}
```

```
try  
{  
    코드 1 - 정상 실행  
    코드 2 - 예외 발생  
    코드 3 - 실행 하지 않음  
}  
catch(Exception e)  
{  
    예외 처리  
}
```



19-3. 예외 던지기(throw) 및 반듯이 실행(finally)

- CLR이 예외를 발생하듯이 사용자가 예외를 발생한다.

```
databaseTask.connectDatabase();
databaseTask.queryDatabase();
databaseTask.closeDatabase();
```

```
public void queryDatabase()
{
    Console.WriteLine("=== queryDatabase() Start ===");
    throw new NullReferenceException("Reference Exception");
}
예외 발생
```

```
try
{
    databaseTask.connectDatabase();
    databaseTask.queryDatabase();
}
catch (NullReferenceException e2)
{
    Console.WriteLine($"e.Message : {e2.Message}");
}
finally
{
    databaseTask.closeDatabase();
}
```

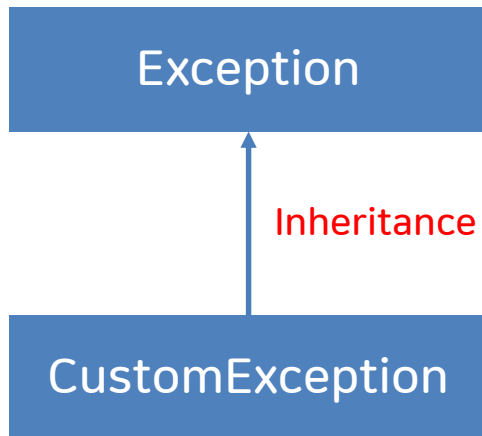
예외가 발생해도 반듯이 실행한다.



19-4. 사용자 예외(Custom Exception)

ex) lec19/exceptionPjt

- Exception을 상속해서 사용자 예외 클래스를 만든다.



```
class CustomException : Exception
{
    public CustomException()
    {
        Console.WriteLine("=== CustomException CONSTRUCTOR ===");
    }

    public CustomException(string message) : base(message)
    {
        Console.WriteLine("=== CustomException(string message) CONSTRUCTOR ===");
    }
}
```