>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 10

예외 처리

예외

Exception

방어 프로그래밍
Defensive Programming

안전 코딩
Secure Coding

방어 프로그래밍
Defensive Programming

안전 코딩
Secure Coding

예외 처리
Exception Handling

CHAPTER 10

예외 처리

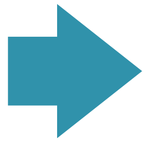
CHAPTER 10

예외 처리

내장 예외

Built-in Exception

예외 타입	발생 상황
SyntaxError	문법이 틀린 경우
TypeError	피연산자 또는 함수 인수의 타입이 틀린 경우
ValueError	피연산자 또는 함수 인수의 값이 틀린 경우
NameError	지정한 적이 없는 모르는 이름이 나타난 경우
IndexError	없는 인덱스를 사용한 경우
KeyError	없는 키를 사용한 경우
ZeroDivisionError	0으로 나누려 하는 경우
IOError	없는 파일을 열려고 하는 경우, 열지 않고 파일을 읽거나 쓰려고 하는 경우 등



>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

p.463

실습 10.1 예외 발생시켜 보기

CHAPTER 10

예외 처리

예외 처리

Exception Handling

code : 10-1.py

```
1 x = int(input("Enter a number: "))
2 reciprocal = 1 / x
3 print("The reciprocal of", x, "is", reciprocal)
```

code : 10-2.py

```
1 try:
2     x = int(input("Enter a number: "))
3     reciprocal = 1 / x
4     print("The reciprocal of", x, "is", reciprocal)
5 except ValueError:
6     print("Not a number.")
7 except ZeroDivisionError:
8     print("The reciprocal of 0 does not exist.")
```

예외 처리

Exception Handling

code : 10-1.py

```
1  x = int(input("Enter a number: "))
2  reciprocal = 1 / x
3  print("The reciprocal of", x, "is", reciprocal)
```

code : 10-3.py

```
1  while True:
2      try:
3          x = int(input("Enter a number: "))
4          reciprocal = 1 / x
5          print("The reciprocal of", x, "is", reciprocal)
6          break
7      except ValueError:
8          print("Not a number.")
9      except ZeroDivisionError:
10         print("The reciprocal of 0 does not exist.")
11         break
```

구문

구문 10.1 예외 처리문

```
try:
    ⟨블록⟩
except ⟨예외이름⟩1:
    ⟨블록⟩1
except ⟨예외이름⟩2:
    ⟨블록⟩2
…
except ⟨예외이름⟩n:
    ⟨블록⟩n
```

의미

의미 10.1 예외 처리문

try 블록인 ⟨블록⟩을 실행한다.

- 예외 상황이 발생하지 않고 ⟨블록⟩의 실행을 종료하면, except 블록은 모두 무시한다.
- ⟨블록⟩의 실행도중 예외 상황이 발생하면 try 블록의 남은 부분은 무시하고, ⟨예외이름⟩$_1$, ⟨예외이름⟩$_2$, …, ⟨예외이름⟩$_n$ 중에서 발생한 예외와 같은 이름(종류)을 위에서부터 나타나는 순서대로 찾아서 해당 블록을 실행한다. 여러 개의 예외처리 블록 중에서 가장 먼저 일치한 예외처리 블록 하나만 선택하여 실행한다.
- 발생한 예외와 같은 이름이 ⟨예외이름⟩$_1$, ⟨예외이름⟩$_2$, …, ⟨예외이름⟩$_n$ 중에 없으면 발생한 오류메시지를 내주면서 실행을 멈춘다.

예외 처리

Exception Handling

code : 10-4.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
        print("The reciprocal of", x, "is", reciprocal)
        break
    except ValueError:
        print("Not a number.")
    except ZeroDivisionError:
        print("The reciprocal of 0 does not exist.")
        break
    except:
        print("Unexpected exception occurred.")
        break
```

예외 처리

Exception Handling

code : 10-5.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
        print("The reciprocal of", x, "is", reciprocal)
        break
    except ValueError as message:
        print(message)
    except ZeroDivisionError as message:
        print(message)
        break
```

code : 10-3.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
        print("The reciprocal of", x, "is", reciprocal)
        break
    except ValueError:
        print("Not a number.")
    except ZeroDivisionError:
        print("The reciprocal of
        break
```

code : 10-6.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
    except ValueError:
        print("Not a number.")
    except ZeroDivisionError:
        print("The reciprocal of 0 does not exist.")
        break
    else:
        print("The reciprocal of", x, "is", reciprocal)
        break
```

code : 10-3.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
        print("The reciprocal of
        break
    except ValueError:
        print("Not a number.")
    except ZeroDivisionError:
        print("The reciprocal of
        break
```

code : 10-7.py

```
while True:
    try:
        x = int(input("Enter a number: "))
        reciprocal = 1 / x
    except ValueError:
        print("Not a number.")
    except ZeroDivisionError:
        print("The reciprocal of 0 does not exist.")
        break
    else:
        print("The reciprocal of", x, "is", reciprocal)
        break
    finally:
        print(":-)")
```

>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

p.471

실습 10.2 실수 입력 확인

CHAPTER 10

예외 처리

구문 / 의미

구문 10.2 assert 문

assert 〈논리식〉

의미 10.2 assert 문

〈논리식〉의 계산 결과가 True이면 그냥 통과하고, False이면 AssertionError라는 이름의 예외를 발생시킨다.

code : 10-8.py

```
1  def fac(n):
2      ans = 1
3      while n > 1:
4          ans = n * ans
5          n = n - 1
6      return ans
```

code : 10-9.py

```
1  def factorial():
2      n = int(input("Enter a number: "))
3      print("factorial(", n, ") = ", fac(n), sep='')
```

code : 10-10.py

```
1  def factorial():
2      n = int(input("Enter a number: "))
3      assert n >= 1
4      print("factorial(", n, ") = ", fac(n), sep='')
```

code : 10-10.py

```
1  def factorial():
2      n = int(input("Enter a number: "))
3      assert n >= 1
4      print("factorial(", n, ") = ", fac(n), sep='')
```

code : 10-11.py

```
1  def factorial():
2      while True:
3          try:
4              n = int(input("Enter a number: "))
5              assert n >= 1
6          except ValueError:
7              print("Not a number.")
8          except AssertionError:
9              print("Not a natural number.")
10         else:
11             print("factorial(", n, ") = ", fac(n), sep='')
12             print("Goodbye!")
13             break
```

>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

pp.474~476

실습 10.3 조합 계산 서비스 구현

CHAPTER 10

예외 처리

사용자 정의 예외

User-defined Exception

code : 10-12.py

예외 정의 →

예외 발생 →

예외 처리 →

```
1  class NonPositive(Exception): pass
2
3  def factorial():
4      while True:
5          try:
6              n = int(input("Enter a number: "))
7              if n < 1:
8                  raise NonPositive
9          except ValueError:
10             print("Not a number.")
11         except NonPositive:
12             print("Not a natural number.")
13         else:
14             print("factorial(", n, ") = ", fac(n), sep='')
15             print("Goodbye!")
16             break
```

>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

p.477

실습 10.4 실수 입력 확인 (범위 제한)

>>>>>>>>>>> 제어 구조의 설계 원리를 중심으로 배우는 >>>>>>>>>>>

프로그래밍의 정석

파이썬

도경구 지음

생능출판

CHAPTER 10

예외 처리