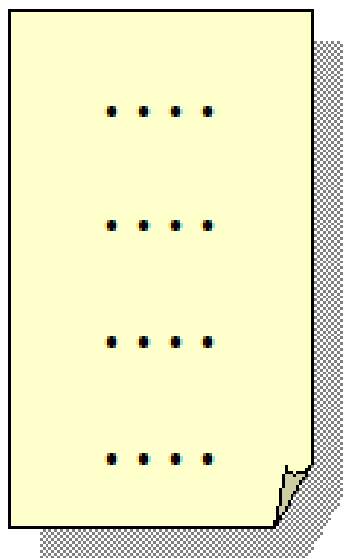


# 예외 처리 Exceptions

2023

국민대학교 소프트웨어학부

# 프로그램의 오류 처리



실행 오류 발생



예외 객체 발생



예외 처리기

# 예외 처리기

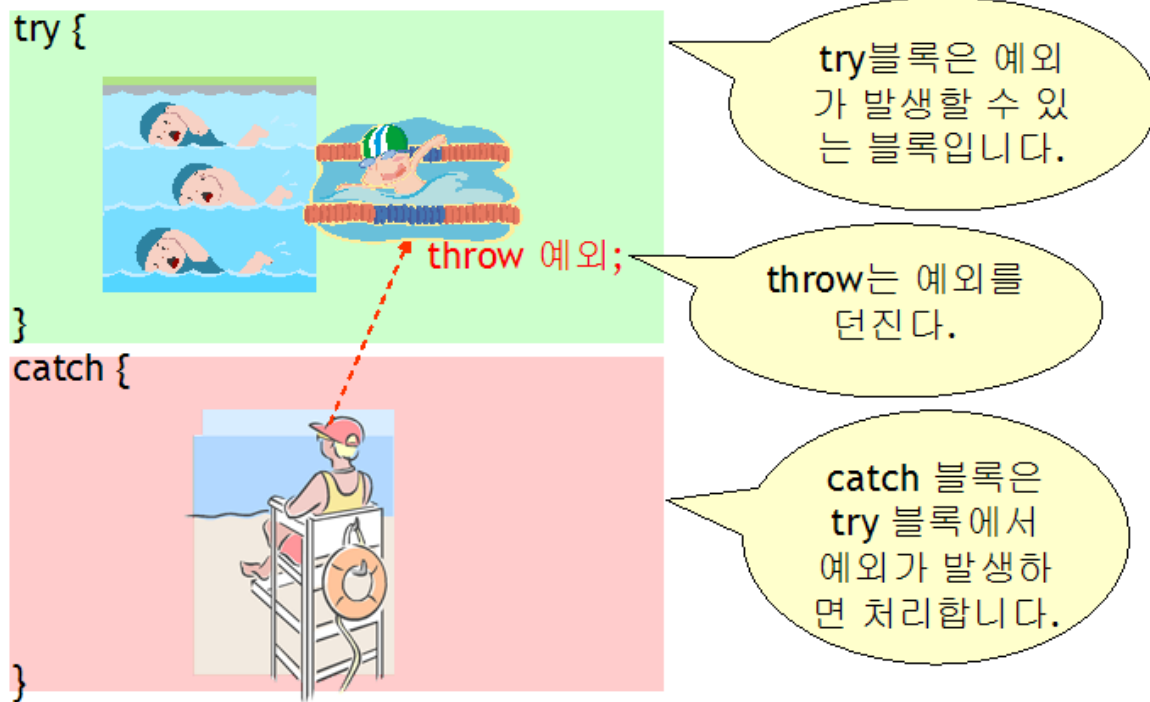



그림 15.2 try블록은 예외가 발생할 수 있는 위험한 코드이다. catch 블록은 예외를 처리하는 코드이다.

# exception 의 catch

- try 문 안에서 throw 문을 수행하면 try 문의 나머지 부분은 수행되지 않는다.
- catch block 이 수행되면 그 다음에는 정상 수행
- if x == true, M2→M3
- if x == false, M1→M3

```
int main(){
    try{
        if (x) throw 1;
        M1
    }
    catch (int e){
        M2
    }
    M3
}
```

매개변수



인자의 타입이 일치되는 예외만 처리한다.

```
try {  
    int person = 0;  
    ...  
    if (persons == 0)  
        throw persons;  
    ...  
}  
catch(char e)  
{  
    cout << "사람이 " << e << " 명 입니다." << endl;  
}
```

타입이 일치하지 않음

# multiple catch blocks

- 매개 변수 type 이 다른 여러 개의 catch blocks 가 있을 수 있다.
  - 매개 변수 type 이 일치하는 **한 개의 catch block 만** 수행된다.
  - catch block 이 수행되면 그 다음에는 정상 수행
- 
- if x == true, M2→M4
  - if x == false && y== true M3→M4
  - if x == false && y== false M1→M4

```
int main(){
    try{
        if (x) throw 1;
        if (y) throw 'a';
        M1
    }
    catch (int e){
        M2
    }
    catch (char e){
        M3
    }
    M4
}
```

# 모든 타입의 예외를 잡고 싶으면

```
catch(...)  
{  
    // 모든 예외를 처리할 수 있다.  
}
```

## multiple catch blocks (2)

- 매개 변수 type 이 일치하는 한 개의 catch block 을 찾는 순서는 앞에서부터
- catch block 이 수행되면 그 다음에는 정상 수행
- if x == true, M2 → M4
- if x == false && y == true M2 → M4
- if x == false && y == false M1 → M4

```
int main(){  
    try{  
        if (x) throw 1;  
        if (y) throw 'a';  
        M1  
    }  
    catch (...){  
        M2  
    }  
    catch (char e){  
        M3  
    }  
    M4  
}
```



## 예외 클래스

- throw 문장은 클래스 타입의 객체도 던질 수 있다.
- 예외 클래스는 보통 아주 단순하다.

```
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : 10
10
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : 0
Zero
0
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : -2
Negative Value :-2
-2
```

```
4 class ZeroException{};
5 class NegativeException{
6 public:
7     int v;
8     NegativeException(int v): v(v){}
9 };
10 int main(){
11     int v;
12     try {
13         cout << " enter a value : ";
14         cin >> v;
15         if (v<0) throw NegativeException(v);
16         if (v==0) throw ZeroException();
17     }
18     catch (ZeroException& e){
19         cout << "Zero \n";
20     }
21     catch (NegativeException& e){
22         cout << "Negative Value :" << e.v << endl;
23     }
24     cout << v << endl;
25     return 0;
26 }
```

## 예외 클래스의 상속

```
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : 10
10
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : 0
Zero
0
ejim@ejim-VirtualBox:~/C2020$ ./exception
enter a value : -2
Negative Value :-2
-2
```

```
4  class NonPositiveException{
5  public:
6      int v;
7      NonPositiveException(int v=0): v(v){}
8  };
9  class ZeroException : public NonPositiveException{};
10 int main(){
11     int v;
12     try {
13         cout << " enter a value : ";
14         cin >> v;
15         if (v<0) throw NonPositiveException(v);
16         if (v==0) throw ZeroException();
17     }
18     catch (ZeroException& e){
19         cout << "Zero \n";
20     }
21     catch (NonPositiveException& e){
22         cout << "NonPositive Value :" << e.v << endl;
23     }
24     cout << v << endl;
25     return 0;
26 }
```

## 예외 클래스의 상속

### catch block 순서가 잘못된 예

```
ejim@ejim-VirtualBox:~/C2020$ ./exception2
enter a value : 10
10
ejim@ejim-VirtualBox:~/C2020$ ./exception2
enter a value : 0
NonPositive Value :0
0
ejim@ejim-VirtualBox:~/C2020$ ./exception2
enter a value : -1
NonPositive Value :-1
-1
```

ZeroException z;  
NonPositiveException& e = z; // implicit upcasting

```
4  class NonPositiveException{
5  public:
6      int v;
7      NonPositiveException(int v=0): v(v){}
8  };
9  class ZeroException : public NonPositiveException{};
10 int main(){
11     int v;
12     try {
13         cout << " enter a value : ";
14         cin >> v;
15         if (v<0) throw NonPositiveException(v);
16         if (v==0) throw ZeroException();
17     }
18     catch (NonPositiveException& e){
19         cout << "NonPositive Value :" << e.v << endl;
20     }
21     catch (ZeroException& e){
22         cout << "Zero \n";
23     }
24     cout << v << endl;
25     return 0;
26 }
```

## 함수 호출의 예외 전달

- 함수가 catch 하지 않는 경우

```
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 1
foo
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 0
Zero
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : -1
NonPositive Value :-1
main
```

```
4  class NonPositiveException{
5  public:
6      int v;
7      NonPositiveException(int v=0): v(v){}
8  };
9  class ZeroException : public NonPositiveException{};
10 void foo(){
11     int v;
12     cout << "enter a value : ";
13     cin >> v;
14     if (v<0) throw NonPositiveException(v);
15     if (v==0) throw ZeroException();
16     cout << "foo\n";
17 }
18 int main(){
19     try {
20         foo();
21     }
22     catch (ZeroException& e){
23         cout << "Zero \n";
24     }
25     catch (NonPositiveException& e){
26         cout << "NonPositive Value :" << e.v << endl;
27     }
28     cout << "main" << endl;
29     return 0;
30 }
```

# 함수 호출의 예외 전달

## - 함수 내에서 catch

```
4 class NonPositiveException{
5 public:
6     int v;
7     NonPositiveException(int v=0): v(v){}
8 };
9 class ZeroException : public NonPositiveException{};
10 void foo(){
11     try {
12         int v;
13         cout << "enter a value : ";
14         cin >> v;
15         if (v<0) throw NonPositiveException(v);
16         if (v==0) throw ZeroException();
17         cout << "foo\n";
18     }
19     catch(...){
20         cout << "foo catches exception\n";
21     }
22 }
```

```
int main(){
    try {
        foo();
    }
    catch (ZeroException& e){
        cout << "Zero \n";
    }
    catch (NonPositiveException& e){
        cout << "NonPositive Value : " << e.v << endl;
    }
    cout << "main" << endl;
    return 0;
}
```

```
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 1
foo
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 0
foo catches exception
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : -1
foo catches exception
main
```



## 함수 호출의 예외 전달

- 함수 내에서 catch 후에 caller 함수에서 다시 catch 하려면

```
4 class NonPositiveException{
5 public:
6     int v;
7     NonPositiveException(int v=0): v(v){}
8 };
9 class ZeroException : public NonPositiveException{};
10 void foo(){
11     try {
12         int v;
13         cout << "enter a value : ";
14         cin >> v;
15         if (v<0) throw NonPositiveException(v);
16         if (v==0) throw ZeroException();
17         cout << "foo\n";
18     }
19     catch(...){
20         cout << "foo catches exception\n";
21         throw;
22     }
23 }
```

```
int main(){
    try {
        foo();
    }
    catch (ZeroException& e){
        cout << "Zero \n";
    }
    catch (NonPositiveException& e){
        cout << "NonPositive Value :" << e.v << endl;
    }
    cout << "main" << endl;
    return 0;
}
```

```
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 1
foo
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : 0
foo catches exception
Zero
main
ejim@ejim-VirtualBox:~/C2020$ ./exception3
enter a value : -1
foo catches exception
NonPositive Value :-1
main
```

예외를 throw 했는데 처리되지 않으면 ?

→ 시스템 라이브러리 함수 abort() 가 호출되어 프로그램이 종료됨

```
10 void foo(){
11     try {
12         int v;
13         cout << "enter a value : ";
14         cin >> v;
15         if (v<0) throw -1; int
16         if (v==0) throw ZeroException();
17         cout << "foo\n";
18     }
19     catch(...){
20         cout << "foo catches exception\n";
21         throw;
22     }
23 }
```

```
int main(){
    try {
        foo();
    }
    catch (ZeroException& e){
        cout << "Zero \n";
    }
    catch (NonPositiveException& e){
        cout << "NonPositive Value :" << e.v << endl;
    }
    cout << "main" << endl;
    return 0;
}
```

```
ejim@ejim-VirtualBox:~/C2020$ ./exception4
enter a value : -1
foo catches exception
terminate called after throwing an instance of 'int'
Aborted (core dumped)
```

# (1) what if x == true?

```
int main(){
  try{
    foo();
    M1
  }
  catch (ChildException& e){
    M2
  }
  catch (ParentException& e){
    M3
  }
  catch (...){
    M4
  }
  M5
}
```

```
void foo(){
  try{
    if (x) throw ChildException();
    F1
  }
  catch (...){
    F2
    throw;
  }
  F3
}
```

F2->M2->M5



## (2) what if x == true?

```
int main(){
  try{
    foo();
    M1
  }
  catch (ChildException& e){
    M2
  }
  catch (ParentException& e){
    M3
  }
  catch (...){
    M4
  }
  M5
}
```

```
void foo(){
  try{
    if (x) throw ChildException();
    F1
  }
  catch (...){
    F2
    // throw;
  }
  F3
}
```

F2->F3->M1->M5

### (3) what if x == true?

```
int main(){  
    try{  
        foo();  
        M1  
    }  
    catch (ChildException& e){  
        M2  
    }  
    catch (ParentException& e){  
        M3  
    }  
    catch (...){  
        M4  
    }  
    M5  
}
```

```
void foo(){  
    try{  
        if (x) throw 1;  
        F1  
    }  
    catch (...){  
        F2  
        throw;  
    }  
    F3  
}
```

F2->M4->M5

## (4) what if x == false?

```
int main(){
    try{
        foo();
        M1
    }
    catch (ChildException& e){
        M2
    }
    catch (ParentException& e){
        M3
    }
    catch (...){
        M4
    }
    M5
}
```

```
void foo(){
    try{
        if (x) throw ChildException();
        F1
    }
    catch (...){
        F2
        throw;
    }
    F3
}
```

F1->F3->M1->M5

# 실습

- BitSet class 의 operator+() 함수에서 두 operands 의 최대 원소 갯수 (N) 이 다르면 IncompatibleException 을 발생하도록 변경하라.
- IncompatibleException 은 std::exception class 를 상속하여 만들어라.
- main() 함수에서 IncompatibleException 을 catch 하여 다음 화면과 같이 출력하도록 변경하라.

```
1 // BitSet.h by ejim@kookmin.ac.kr
2 #include <iostream>
3 #include <exception>
4 #include "Bvector.h"
```

멤버 변수와 생성자를 구현하라.

```
5 > class IncompatibleException : public std::exception{=};
```

```
10
11 class BitSet : public Bvector{
12     public:
13         BitSet(int sz=32);
14         void insert(int v);
15         friend BitSet operator+(const BitSet& v1, const BitSet& v2);
16         friend std::ostream& operator<<(std::ostream& os, const BitSet& s);
17     };

```

IncompatibleException 를 발생하도록 BitSet.cpp에서 함수 구현을 변경하라.

```

24 int main(int argc, char *argv[]){
25     BitSet b1(132), b2(131);
26     b1.insert(3); b1.insert(5); b1.insert(8);
27     b2.insert(4); b2.insert(5); b2.insert(8); b2.insert(130);
28     b1.print(); b2.print();
29     cout << "b1= " << b1 << endl;
30     cout << "b2= " << b2 << endl;
31     cout << "b1+b2= " << b1+b2 << endl;

```

IncompatibleException 발생

[illegible]

main() 을 수정하여 다음과 같은 출력이 되게 하라.

[illegible]