

응용 프로그래밍

(1) **[생성형 AI 활용]** 제시된 프로그램에 대하여 AI를 활용하여 문제를 해결하세요. (2, 3번)

```
public class Test {  
    public static void main(String[] args) {  
        sub();  
    }  
    public static void sub() {  
        int[] array = new int[10];  
        int i = array[10];  
    }  
}
```

- 1) 위의 프로그램은 컴파일 시에 오류가 발생한다. 어떤 오류가 발생하나요?
- 2) try/catch 블록을 사용하여 예외를 처리하여 보세요.
- 3) throws 선언을 이용하여 예외를 처리하여 보세요.

생 성 형 AI 플 랫폼	Perplexity
질 문	<pre>public class Test {     public static void main(String[] args) {         sub();     }     public static void sub() {         int[] array = new int[10];         int i = array[10];     } }</pre> <p>try/catch 블록을 사용하여 예외를 처리하여 보세요. throws 선언을 이용하여 예외를 처리하여 보세요. 해결해줘</p>
답 변	1.

```
실행 Test x
/Users/hataeyeong/Library/Java/JavaVirtualMachines/temurin-17.0.14/Contents/Home/bin/java -javaagent:/Applications/
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Create breakpoint : Index 10 out of bounds for leng
    at Answer1.Test.sub(Test.java:9)
    at Answer1.Test.main(Test.java:5)
종료 코드 1(으)로 완료된 프로세스
```

2.

try/catch 블록을 사용하여 예외 처리

java

```
public class Test {
    public static void main(String[] args) {
        try {
            sub();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("예외 발생: " + e.getMessage());
        }
    }

    public static void sub() {
        int[] array = new int[10];
        int i = array[10]; // 배열의 인덱스 범위를 벗어나 예외 발생
    }
}
```

- sub() 메소드에서 배열의 범위를 벗어난 인덱스(10)에 접근하므로 `ArrayIndexOutOfBoundsException` 이 발생합니다.
- main()에서 try/catch 로 예외를 받아 메시지를 출력합니다

3.

throws 선언을 이용하여 예외 처리

java

```
public class Test {
    public static void main(String[] args) {
        try {
            sub();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("예외 발생: " + e.getMessage());
        }
    }

    public static void sub() throws ArrayIndexOutOfBoundsException {
```

<pre>int[] array = new int[10]; int i = array[10]; // 배열의 인덱스 범위를 벗어나 예외 발생 } }</pre>
<ul style="list-style-type: none"><li>sub() 메소드 선언부에 throws <code>ArrayIndexOutOfBoundsException</code> 를 추가합니다.</li><li>예외 처리는 여전히 main()에서 try/catch 로 처리해야 프로그램이 비정상 종료되지 않습니다</li></ul>
<b>정리</b> <ul style="list-style-type: none"><li><code>ArrayIndexOutOfBoundsException</code> 은 런타임 예외(Checked Exception 이 아님)이기 때문에 throws 선언 없이도 자동으로 전파됩니다. 하지만 명시적으로 throws 를 선언할 수 있습니다</li></ul> 예외가 발생할 가능성이 있는 코드는 try/catch로 감싸서 프로그램의 비정상 종료를 방지할 수 있습니다

(2) **[생성형 AI 활용]** 제시된 프로그램에서 오류가 발생하는 원인과 해결방법을 제시하세요.

```
public static void main(String[] args) {  
    try {  
        int x = 0;  
        int y = 5 / x;  
    }  
    catch (Exception e) {  
        System.out.println("Exception");  
    }  
    catch (ArithmeticException ae) {  
        System.out.println(" Arithmetic Exception");  
    }  
    System.out.println("finished");  
}
```

생성형 AI 플랫폼	Perplexity
질문	<p>제시된 프로그램에서 오류가 발생하는 원인과 해결방법을 제시하세요.</p> <pre>public static void main(String[] args) {     try {         int x = 0;         int y = 5 / x;     }     catch (Exception e) {         System.out.println("Exception");     }     catch (ArithmeticException ae) {         System.out.println(" Arithmetic Exception");     }     System.out.println("finished"); }</pre> <p>해결해줘</p>
답변	오류 발생 원인

제시된 프로그램에서 발생하는 오류의 원인은 **정수 0으로 나누기**입니다.

아래 코드의 `int y = 5 / x;` 부분에서 `x`가 0 이므로, 자바에서는  
이때 `ArithmeticException`이 발생합니다

```
int x = 0;
```

```
int y = 5 / x; // 여기서 ArithmeticException 발생
```

해결 방법

**catch 문의 순서가 잘못되어 컴파일 오류가 발생합니다.**

자바에서는 예외를 처리할 때, **\*\*더 구체적인 예외(자식 클래스)\*\***를 먼저, **\*\*더 일반적인 예외(부모 클래스)\*\***를 나중에 catch 해야 합니다.

현재 코드는 `Exception(부모)`을 먼저, `ArithmeticException(자식)`을 나중에 catch 하고  
있어 컴파일 에러가 발생합니다

올바른 예외 처리 순서

아래와 같이 **구체적인 예외를 먼저**, **일반적인 예외를 나중에** catch 하도록 순서를 바꿔야  
합니다.

```
public static void main(String[] args) {  
    try {  
        int x = 0;  
        int y = 5 / x;  
    }  
    catch (ArithmeticException ae) {  
        System.out.println("Arithmetic Exception");  
    }  
    catch (Exception e) {  
        System.out.println("Exception");  
    }  
    System.out.println("finished");  
}
```

요약

- **오류 원인:** 0으로 나누는 연산(`5 / 0`)에서 `ArithmeticException`이 발생함.
- **해결 방법:** catch 문의 순서를 `ArithmeticException` → `Exception` 순으로  
바꿔야 함.  
(구체적인 예외를 먼저, 일반적인 예외를 나중에!)

이렇게 수정하면 프로그램은 "Arithmetic Exception"과 "finished"를 정상적으로  
출력합니다.

(3) 다음은 사용자 예외를 정의하고 사용하는 간단한 예이다. `checkNegative()` 메소드는 음수가 전달되면 사용자  
예외를 발생한다. 빈칸을 채워 컴파일하고 실행하세요.

```

class MyException extends _____ {
    public MyException(String message) { super(message); }
}

public class MyExceptionTest {
    public static void checkNegative(int number) throws MyException {
        if (number < 0) {
            _____(new MyException("음수는 안됩니다."));
        }
        System.out.println("다행히 음수가 아니군요");
    }
    public static void main(String[] args) {
        try {
            checkNegative(1);
            checkNegative(-1);
        } catch (MyException ex) { ex.printStackTrace(); }
    }
}

```

---

#### [프로그램 소스]

```

package answer3;

class MyException extends Exception {
    public MyException(String message) { super(message); }
}

```

```

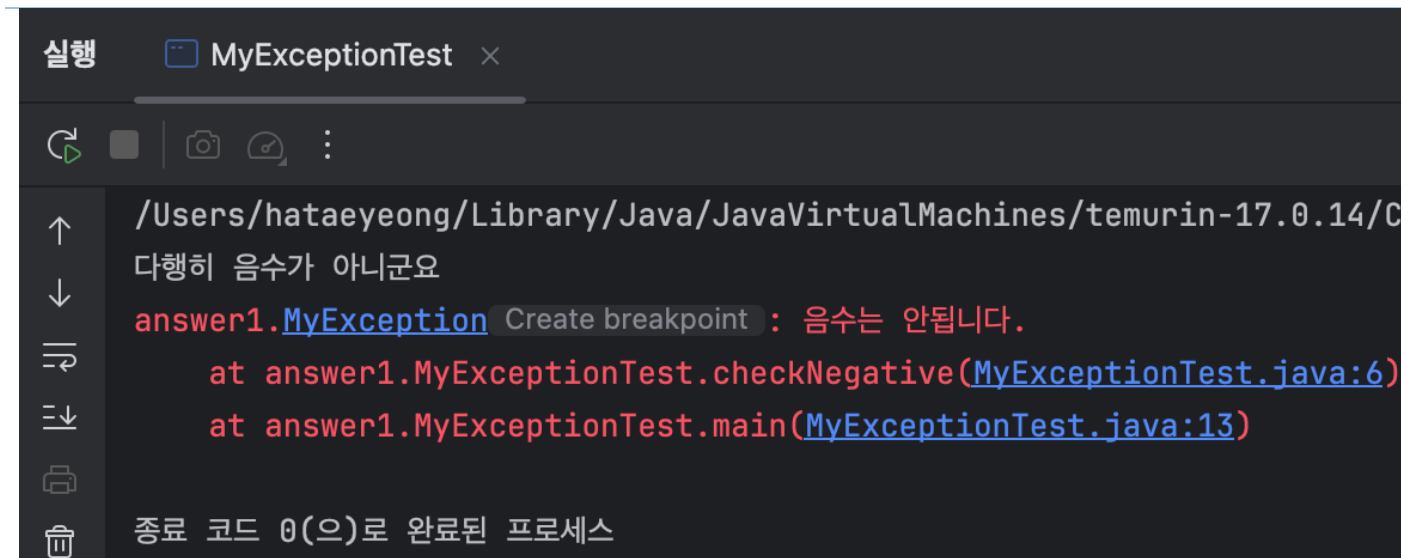
package answer3;

public class MyExceptionTest {
    public static void checkNegative(int number) throws MyException {
        if (number < 0) {
            throw new MyException("음수는 안됩니다.");
        }
        System.out.println("다행히 음수가 아니군요");
    }
    public static void main(String[] args) {
        try {
            checkNegative(1);
            checkNegative(-1);
        } catch (MyException ex) { ex.printStackTrace(); }
    }
}

```

---

#### [실행 결과]



- (4) 로그인 기능을 LoginExample 클래스의 login() 메소드에서 구현하려고 합니다. 존재하지 않은 ID를 입력했을 경우 NotExistIDException을 발생시키고, 잘못된 패스워드를 입력했을 경우 WrongPasswordException을 발생시키려고 합니다. 제시된 클래스의 구조를 보고 적절하게 변경 구현하여 프로그램을 테스트 하세요.

```
public class NotExistIDException {
    //~~~~~
}

public class WrongPasswordException {
    //~~~~~
}

public class LoginExample {
    public static void main(String[] args) {
        try {
            login("white", 12345);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        try {
            login("blue", 54321);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }

    public static void login(String id, int password) {
        //id가 blue가 아니라면 NotExistIDException 발생시킴
        //~~~~~

        //password가 12345가 아니라면 WrongPasswordException 발생시킴
        //~~~~~
    }
}
```

```
}  
}
```

---

#### [프로그램 소스]

```
package answer4;  
  
public class NotExistIDException extends Exception {  
    public NotExistIDException() {}  
    public NotExistIDException(String message) {  
        super(message);  
    }  
}
```

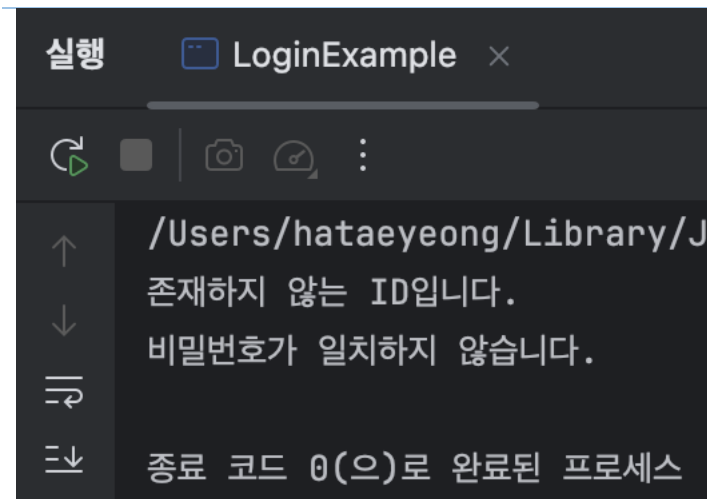
```
package answer4;  
  
public class WrongPasswordException extends Exception {  
    public WrongPasswordException() {}  
    public WrongPasswordException(String message) {  
        super(message);  
    }  
}
```

```
package answer4;  
  
public class LoginExample {  
    public static void main(String[] args) {  
        try {  
            login("white", 12345);  
        } catch (Exception e) {  
            System.out.println(e.getMessage()); // message 를 문자열로 출력  
        }  
  
        try {  
            login("blue", 54321);  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
    public static void login(String id, int password) throws NotExistIDException,  
WrongPasswordException {  
        //id 가 blue 가 아니면 NotExistIDException 발생시킴  
        if(!id.equals("blue")) {  
            throw new NotExistIDException("존재하지 않는 ID 입니다."); // 예외도 객체이므로  
new 로 생성해야한다.  
        }  
  
        //password 가 12345 가 아니면 WrongPasswordException 발생시킴  
        if(password != 12345) {  
            throw new WrongPasswordException("비밀번호가 일치하지 않습니다.");  
        }  
    }  
}
```

---

#### [실행 결과]

---



(5) 필요한 클래스를 구현하고 프로그램을 테스트 하세요.

- 은행 예금을 나타내는 클래스 BankAccount는 잔액(private int balance)을 필드로 가지며, 입금을 나타내는 deposit() 메소드와 출금을 나타내는 withdraw() 메소드를 가진다.
- deposit() 메소드는 매개변수로 전달된 금액(amount)을 잔액에 추가한 후 추가된 잔액을 반환한다.
- withdraw() 메소드는 전달된 인출 금액(amount)이 현재의 잔액(balance)보다 작거나 같다면 현재의 잔액에서 인출 금액을 빼고 잔액을 반환 한다. 그러나 인출 금액(amount)이 현재의 잔액(balance)보다 크면 NegativeBalanceException("잔고가 음수입니다.")을 발생한다. NegativeBalanceException이 발생하면 현재의 잔액은 변경 없이 그대로 리턴 한다.
- BankAccountTest 클래스에서 BankAccount 객체를 생성하여 deposit(), withdraw()를 호출하여 다음 그림처럼 수행되도록 한다.

```
<terminated> OurClassTest [Java]
저축 100, 잔고 = 100
저축 500, 잔고 = 600
출금 800, 잔고가 음수입니다.
잔고 = 600
```

#### [프로그램 소스]

```
package answer5;

public class NegativeBalanceException extends Exception {
    public NegativeBalanceException() {}
    public NegativeBalanceException(String message) {
        super(message);
    }
}
```

```
package answer5;

public class BankAccount {
    private int balance = 0; // 잔액

    public BankAccount() {
        this.balance = balance;
    }
}
```



```
// 입금
public int deposit(int amount) {
    balance += amount;
    return balance;
}

// 출금
public int withdraw(int amount) {
    try {
        if(balance >= amount) {
            balance -= amount;
        } else {
            throw new NegativeBalanceException("잔고가 음수입니다.");
        }
    } catch (NegativeBalanceException e) {
        System.out.println(e.getMessage());
        System.out.println("잔고 = " + balance);
    }
    return balance;
}
}
```

```
package answer5;

public class BankAccountTest {
    public static void main(String[] args) {
        BankAccount bankAccount = new BankAccount();

        System.out.println("저축 100, 잔고 = " + bankAccount.deposit(100));
        System.out.println("저축 500, 잔고 = " + bankAccount.deposit(500));

        // try-catch 없이 바로 호출
        System.out.print("출금 800, ");
        bankAccount.withdraw(800); // 잔고 확인
    }
}
```

#### [실행 결과]

```
실행 BankAccountTest x
/Users/hataeyeong/Library/...
저축 100, 잔고 = 100
저축 500, 잔고 = 600
출금 800, 잔고가 음수입니다.
잔고 = 600
종료 코드 0(으)로 완료된 프로세스
```

(6) 필요한 클래스를 구현하고 프로그램을 테스트 하세요.

- ① 숫자를 저장하고 있는 배열과 특정 숫자를 받아서 특정 숫자를 찾는 `searchArray()` 정적 메소드를 작성한다.
- ② `searchArray()`는 배열 안에 원하는 숫자가 있으면 배열에서 그 숫자가 존재하는 위치를 반환한다. 만약 원하는 숫자가 없으면 `NotFoundException("배열에서 찾을 수 없습니다.")` 예외를 발생한다.
- ③ `searchArray()` 메소드에서 발생하는 예외는 이 메소드를 호출한 곳에서 `try/catch`를 이용하여 처리 한다.
- ④ `NotFoundException` 클래스는 사용자 정의 예외로서 `Exception` 클래스를 상속받아서 정의한다.

---

#### [프로그램 소스]

##### 찾은 경우

```
package answer6;

public class NotFoundException extends Exception{
    public NotFoundException() {}
    public NotFoundException(String message) {
        super(message);
    }
}
```

```
package answer6;
public class Main {
    public static int searchArray(int[] arr, int target) throws NotFoundException {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                return i;
            }
        }
        throw new NotFoundException("배열에서 찾을 수 없습니다.");
    }
    public static void main(String[] args) {
        int [] arr = {1, 2, 3, 4, 5};
        try {
            int target = 3;
            int idx = searchArray(arr, target);
            System.out.println("찾은 인덱스 : " + idx);
        } catch (NotFoundException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

##### 못 찾은 경우

```
package answer6;

public class NotFoundException extends Exception{
    public NotFoundException() {}
    public NotFoundException(String message) {
        super(message);
    }
}
```

```
package answer6;
public class Main {
    public static int searchArray(int[] arr, int target) throws NotFoundException {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                return i;
            }
        }
        throw new NotFoundException("배열에서 찾을 수 없습니다.");
    }
}
```

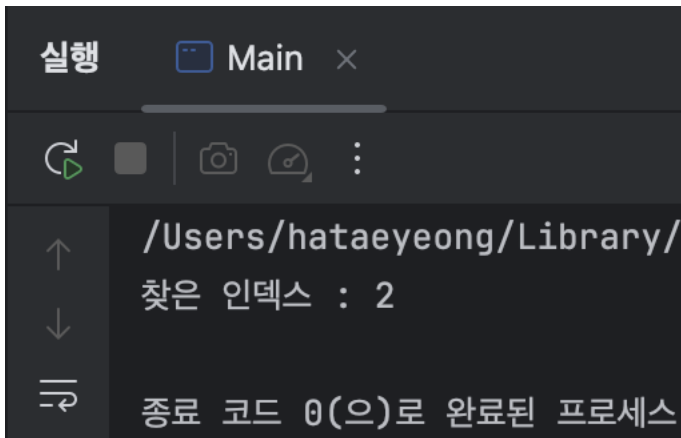
```

    }
    public static void main(String[] args) {
        int [] arr = {1, 2, 3, 4, 5};
        try {
            int target = 7;
            int idx = searchArray(arr, target);
            System.out.println("찾은 인덱스 : " + idx);
        } catch (NotFoundException e) {
            System.out.println(e.getMessage());
        }
    }
}

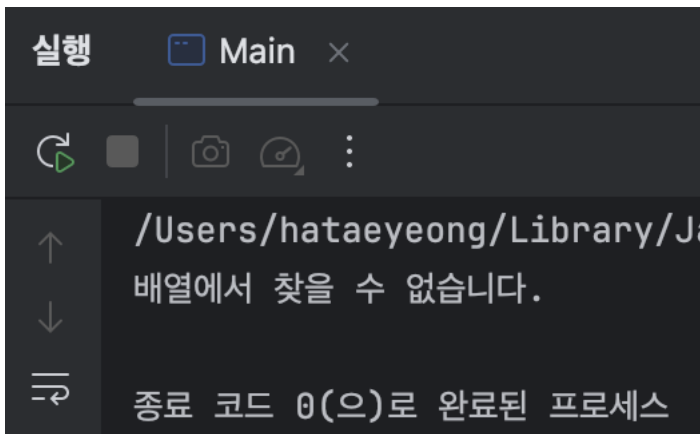
```

[실행 결과]

찾은 경우



못 찾은 경우



- (7) 키보드로 입력 받은 문자열에 대하여 길이가 8 미만이거나 첫 글자가 영문자가 아니면 NotUseId 사용자 예외를 발생시켜 "사용할 수 없는 아이디"를 출력하고, 길이가 8 이상이고 첫 글자가 영문자 이면 "사용 가능한 아이디 입니다." 를 출력한다.

아이디를 입력하세요

345fshsty3

345fshsty3는 사용할 수 없는 아이디

아이디를 입력하세요

a485skfdg

a485skfdg는 사용 가능한 아이디 입니다.

[프로그램 소스]

```

package answer7;

public class NotUseId extends Exception {
    public NotUseId() {}
    public NotUseId(String message) {
        super(message);
    }
}

```

```

    }
}

package answer7;

import java.util.Scanner;

public class Main {
    public static void checkId(String id) throws NotUseId{
        char ch = id.charAt(0);
        if (id.length() < 8 ||
            !((ch >= 'a' && ch <= 'z') ||
              (ch >= 'A' && ch <= 'Z'))){
            throw new NotUseId(id + "는 사용할 수 없는 아이디 입니다.");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("아이디를 입력하세요");
        String id = sc.next();
        try {
            checkId(id);
            System.out.println(id + "는 사용 가능한 아이디 입니다.");
        } catch (NotUseId e){
            System.out.println(e.getMessage());
        }
    }
}

```

[실행 결과]

```

실행 Main x
[Run] [Stop] [Screenshot] [Copy] [Paste] [Find] [More]
/Users/hataeyeong/Library/Jav
아이디를 입력하세요
1231
1231는 사용할 수 없는 아이디 입니다.
종료 코드 0(으)로 완료된 프로세스

```

실행

Main ×



/Users/hataeyeong/Library/Java/J

아이디를 입력하세요

a123123

a123123는 사용할 수 없는 아이디입니다.

종료 코드 0(으)로 완료된 프로세스

실행

Main ×



/Users/hataeyeong/Library/Java/Java

아이디를 입력하세요

a12312312313

a12312312313는 사용 가능한 아이디입니다.

종료 코드 0(으)로 완료된 프로세스