

# Report\_Chap09

Date: 2022. 05. 17

Name: 송예지

Student ID: 22100396

(9 장) Page 777. #1.

- Write a program that converts a time from 24-hour notation to 12- hour notation.  
The following is a sample interaction between the user and the program:
- Define an exception class called `TimeFormatException`. If the user enters an illegal time, like 10:65, or even gibberish, like 8&\*68, your program `should throw and handle a TimeFormatException`.

시간을 24 시간 표기법에서 12 시간 표기법으로 변환하는 프로그램을 작성한다. 다음은 사용자와 프로그램 간의 상호 작용 예제입니다.

- `TimeFormatException` 이라는 예외 클래스를 정의합니다. 사용자가 10:65 와 같이 잘못된 시간을 입력하거나 8&\*68 과 같이 횡설수설하는 경우 프로그램에서 `TimeFormatException` 을 던져 처리해야 합니다.
- 시간을 24 시간 표기법에서 12 시간 표기법으로 변환하는 프로그램을 작성한다. 다음은 사용자와 프로그램 간의 상호 작용 예제입니다.
- `TimeFormatException` 이라는 예외 클래스를 정의합니다. 사용자가 10:65 와 같이 잘못된 시간을 입력하거나 8&\*68 과 같이 횡설수설하는 경우 프로그램에서 `TimeFormatException` 을 던져 처리해야 합니다.

---

## 기본 문제 해결

⇒ TimeConvert\_main class

- 시간 받을 변수 생성
- 프로그램을 한 번 돌릴지에 관한 변수 생성
- 받은 시간을 시간과 분으로 나눠줄 string array 생성
- Am 인지 Pm 인지 구분해줄 변수 생성 (when)
- While 문으로 코드 구현
- 시간을 입력 받고 ":"를 기준으로 시간과 분을 구분
- 구분해준 값을 int 값으로 typecast
- 받은 시간이 이상한 값이면 throw
- 받은 시간이 정상적인 값이면 조건에 따라 값 출력
  - Hour 이 12 보다 많으면 12 를 빼주고
  - When 변수에 pm 넣기
  - Hour 이 12 보다 같거나 작으면
  - When 변수에 am 넣기
  - -----
  - Minute 가 10 보다 작은 숫자이면
  - 출력 전에 0 출력해주기 (ex) 09
- 프로그램 또 돌릴건지 묻고 답변 받기
  - Y 면 더 돌리고
  - N 이면 프로그램 종료

⇒ TimeFormatException class

- Main에서 받은 시간 값이 이상한 값이면
  - Try 문에서 exception 을 던짐
  - Catch 문에서 메세지를 출력하고 프로그램 다시 돌림

---

## # 실습 코드

```
# TimeConvert_main.java
import java.util.Scanner;

public class TimeConvert_main {
    public static void main(String [] args) {
        Scanner s = new Scanner(System.in);
        String sentence;
        String answer;
        String[] result;
        String when;

        while(true) {
            System.out.println("Enter time in 24-hour notation:");
            sentence = s.nextLine();
            result = sentence.split(":");
            // System.out.println(result[0]); // 시간(시) 확인용
            // System.out.println(result[1]); // 시간(분) 확인용

            int hour = Integer.parseInt(result[0]);
            int minute = Integer.parseInt(result[1]);
            TimeFormatException time = new TimeFormatException(hour, minute);

            if((hour>24||hour<=0)||minute>60||minute<0)) {
                time.errorTime();
            }
            else {
                if(hour>12) {
                    hour = hour-12;
                    when = "PM";
                }
                else when = "AM";

                System.out.println("That is the same as");
                if(minute<10)
                    System.out.println(hour+":0"+minute+" "+when);
                else
                    System.out.println(hour+":"+minute+" "+when);
                System.out.println("Again? (y/n)");
            }
        }
    }
}
```

```
        answer = s.nextLine();
        if(answer.equalsIgnoreCase("N")) {
            System.out.println("End of program");
            break;
        }
    }

    s.close();
}
}
```

```
# TimeFormatException.java
public class TimeFormatException {
    private int h;
    private int m;

    public TimeFormatException(int h1, int m1) {
        h = h1;
        m = m1;
    }

    public TimeFormatException() {
        h = 0;
        m = 0;
    }

    public void errorTime() {
        try {
```

```

        throw new Exception("There is no such time as "+h+":"+m);
    }
    catch(Exception e){
        System.out.println(e.getMessage());
        System.out.println("Try Again:");
    }
}

```

## # 실습 코드 설명

```

# TimeConvert_main.java
import java.util.Scanner;

public class TimeConvert_main {
    public static void main(String [] args) {
        Scanner s = new Scanner(System.in);
        → 값을 입력 받기 위해 Scanner class의 Object 생성
        String sentence;
        → 시간 입력 받을 변수
        String answer;
        → 프로그램을 다시 돌릴지에 대한 여부
        String[] result;
        → 입력 받은 시간을 hour와 minute로 나눠주기 위한 string array 생성
        String when;
        → 시간이 am인지 pm인지 저장하기 위한 변수

        while(true) {
        → Break가 나오기 전까지 계속 Loop문 반복
            System.out.println("Enter time in 24-hour notation:");
        → 시간을 입력하라는 문구
            sentence = s.nextLine();
        → 시간 입력받기
            result = sentence.split(":");

```

- String array인 result[]를 기준으로 sentence를 나눠줌

```
// System.out.println(result[0]); // 시간(시) 확인용
// System.out.println(result[1]); // 시간(분) 확인용
```

```
int hour = Integer.parseInt(result[0]);
```

- Result 0번째 값을 int형식으로 바꿔서 hour에 넣어줌

```
int minute = Integer.parseInt(result[1]);
```

- Result 1번째 값을 int형식으로 바꿔서 minute에 넣어줌

```
TimeFormatException time = new TimeFormatException(hour, minute);
```

- TimeFormatException class의 object 선언

```
if((hour>24||hour<=0)||minute>60||minute<0)) {
```

```
    time.errorTime();
```

- 이상한 시간이 입력되었으면 errorTime()을 통해 throw

```
}
```

```
else {
```

- 제대로된 시간이 입력되었을 때

```
if(hour>12) {
```

- Hour가 13 이상이면

```
    hour = hour-12;
```

- Hour 값에서 12를 빼주고

```
    when = "PM";
```

- Am 인지 pm 인지 알려주는 변수에 pm 값 저장

```
}
```

```
else when = "AM";
```

- Hour가 12 이하의 값이 들어오면, Am 인지 pm 인지 알려주는 변수에 am 값 저장

```
System.out.println("That is the same as");
```

- "That is the same as "라는 문자열 출력

```
if(minute<10)
```

- Minute가 10보다 작은 수가 들어오면

```
    System.out.println(hour+":0"+minute+" "+when);
```

- Minute 출력 전에 '0'출력

```
Else
```

- Minute가 10보다 큰 수가 들어오면

```
    System.out.println(hour":"+minute+" "+when);
```

- 그대로 출력

```
System.out.println("Again? (y/n)");
```

- 프로그램을 더 돌릴건지에 대한 여부를 묻고

```
answer = s.nextLine();
```

```

➔ 입력 받았을 때
    if(answer.equalsIgnoreCase("N")) {
➔ 대소문자 구분 무시하고 'n'이 들어오면
        System.out.println("End of program");
        break;
➔ "End of program" 출력해주고 while문 종료
    }
}

s.close();
}
}

```

```

# TimeFormatException.java
public class TimeFormatException {
    private int h;
    private int m;
    ➔ Instance variable 선언
    ➔ H-> hour
    ➔ M-> minute

    public TimeFormatException(int h1, int m1) {
        h = h1;
        m = m1;
    }
    ➔ Constructor 생성
}

```

```
public TimeFormatException() {  
    h = 0;  
    m = 0;  
}  
→ Default constructor 생성  
  
public void errorTime() {  
    try {  
        throw new Exception("There is no such time as "+h+":"+m);  
    } → Error throw 함  
    catch(Exception e){  
        System.out.println(e.getMessage());  
    } → Error message 출력  
    → 해결책 제시  
    }  
}  
}
```