객체지향프로그래밍
Week6

학과:컴퓨터공학과
학번: 2022112701
이름:안소희
제출일자:2023.10.15
담당 교수님:한인 교수님

실습 문제 1

1. 문제 분석

Try catch블록을 사용하여 숫자가 아닌 문자를 받았을 때의 예외를 처리하고, 반복문을 통해 54를 더한 값을 출력하는 것을 수행하고 조건문을 통해 q를 입력했을 때 루프에서 빠져나오도록 코드를 작성했다.

2. 프로그램 설계 및 알고리즘

먼저 문자열을 통해 사용자에게 입력을 받고 사용자가 q를 입력했을 때 루프에서 빠져나오고 그게 아니라면 문자열을 부동소수점형으로 변환하도록 한다. 변환 후 문자열에 정수가 입력되지 않았다면 즉, 부동소수점형이라면 정수 입력 메세지를 출력한다. 정수라면 54를 더한 뒤 결과를 출력한다. 만약 숫자가 아니라면 부동소수점형으로 변환하지 못하므로 catch블록으로 넘어가 NumberFormatException 예외를 처리한다.

3. 소스코드 및 주석

```
package week6_1;
import java.util.Scanner;
public class Add_54 {

	public static void main(String[] args) {

		double number;
		number = 0;
		String sInput; //입력을 저장하는 문자열 변수 선언
		Scanner oInDev; //입력을 읽는 Scanner 객체 생성

		oInDev = new Scanner(System.in); // Scanner 객체 초기화

		while (true) {//무한 루프
		System.out.print("숫자 입력 종료(q): ");
		  sInput = oInDev.next();// 사용자 입력을 읽어 문자열 변수 sInput 에 저장

		  if (sInput.equals("q")) {// 만약 사용자가 "q"를 입력했다면
			System.out.println("main 메소드 종료");
			oInDev.close();//Scanner 객체를 닫음
			break;//무한 루프 종료
		    }

		  try {
			number = Double.parseDouble(sInput);// 문자열을 숫자로 변환하여 number 변수에 저장
			if(number%1!=0) {// 정수가 아닐 때(부동소수점형)
```

```
                        System.out.println("정수를 입력하세요 ");//정수 입력
조건 충족
                    }
                    else {;// 정수일 때

                        number = number + 54.0;// 입력된 숫자에 54 를 더하고
결과를 result 변수에 저장
                        System.out.println("숫자 + 54 : " + number);
                    }
                }
                catch (NumberFormatException e) {// 만약 변환이 실패하면
NumberFormatException 예외를 처리
                    System.out.println("문자가 아닌 숫자 입력할 것 ");// 예외
메시지 출력
                }

            }

        }


}
```

4. 결과 및 결과 분석

```
숫자 입력 종료(q): 4
숫자 + 54 : 58.0
숫자 입력 종료(q): 4.5
정수를 입력하세요
숫자 입력 종료(q): 6
숫자 + 54 : 60.0
숫자 입력 종료(q): a
문자가 아닌 숫자 입력할 것
숫자 입력 종료(q): adc
문자가 아닌 숫자 입력할 것
숫자 입력 종료(q): q
main 메소드 종료
```

4 입력 시 q가 아니므로 try문에 들어가 문자열을 숫자로 바꾼 뒤 %1을 시행하여 정수임을 판단하고 정수이므로 54를 더한 값을 출력한다. 4,5 입력 시 q가 아니므로 try문에 들어가 문자열을 숫자로 바꾼 뒤 %1을 시행하여 정수임을 판단하고 정수가 아니므로 정수를 입력하라는 문구가 뜬가 6 입력 시 4 입력할 때와 똑같다 a 입력 시 q가 아니므로 try블록에 들어간다. 이때 문자열을 숫자로 변환하지 못하므로 예외처리가 발생해 문자가 아닌 숫자를 입력하라는 문구가 뜬다. adc를 입력했을 때에도 이전과 같이 예외처리가 발생해 문자가 아닌 숫자를 입력하라는 문구가 뜬다. q 입력 시 sInput.equals("q")가 true이므로 Scanner 객체를 닫은 뒤 무한 루프에서 빠져나가 종료된다.

5. 소감

예외처리블럭(try-catch)을 어떻게 사용할지 if else문과 무엇이 다를지 생각하게 되는 과제였다. If else과 try-catch블록의 다른 점은 첫 번째로 구문, 블록의 차이이고 if else문은 에러가 발생한 객체에 대해 수명이 유지되고 try catch블록은 예외를 발생시킴과 동시에 try 안의 모든 객체는 수명이 유지되지 않아 try블록 안에서 예외가 발생했을 가능성이 있는 모든 객체의 참조를 막아 더욱 안전한 예외처리를 할 수 있다는 점이다.

실습 문제 2

1. 문제 분석

먼저 기본 생성자로 삼각형을 생성하고 메서드를 통해 밑변과 높이를 설정해야겠다고 생각했다. 조건문을 통해 넓이 비교를 한 뒤 결과를 입출력했다.

2. 프로그램 설계 및 알고리즘

먼저 Tri class를 생성하고 기본생성자, 밑변과 높이 설정 메서드, 넓이 구하는 메서드를 생성한다. 그리고 main함수가 포함된 Area class에서 삼각형 2개의 객체를 생성 후 밑변과 높이를 입력을 받고 밑변과 높이 설정한 뒤 각 객체의 getArea() 메서드를 실행하여 넓이를 할당 받은 뒤 넓이 비교 후 결과를 출력하도록 코드를 작성했다.

3. 소스코드 및 주석

Tri class

```
package week6_2;

public class Tri { // Tri 클래스 정의
    private double width; // 밑변 필드 같은 클래스 내에서만 사용하므로 private 접근 지정자 사용
    private double height; // 높이 필드 같은 클래스 내에서만 사용하므로 private 접근 지정자 사용

    // 기본 생성자: 밑변과 높이를 0.0 으로 초기화 Area class 가 사용하므로 디폴트로 접근지정자 설정
    Tri() {
        this.width = 0.0;
        this.height = 0.0;
    }

    // 밑변, 높이 설정 메서드 Area class 가 사용하므로 디폴트로 접근지정자 설정

    void setlength(double width, double height) {
        this.width = width;
        this.height = height;
    }
```

```
    // 삼각형의 넓이를 계산하는 메서드 Area class 가 사용하므로 디폴트로 접근지정자 설정

    double getArea() {
        return 0.5 * width * height;
    }
}
```

Area class(main)

```
package week6_2;
import java.util.Scanner;

public class Area {

       public static void main(String[] args) {
        Scanner oInDev;//사용자의 입력을 받기 위해 Scanner 객체 선언
        double height1, height2;//높이 선언
        double width1, width2;//밑변 선언
        double area1,area2;//넓이 선언
        oInDev = new Scanner(System.in);//Scanner 객체 초기화

        Tri triangle1 = new Tri(); // Tri 클래스 기본 생성자 사용하여 생성
        System.out.print("삼각형 1 밑변: ");
        width1 = oInDev.nextDouble();//밑변 사용자에게 double 형으로 입력 받음
        System.out.print("삼각형 1 높이: ");
        height1 = oInDev.nextDouble();//높이 사용자에게 double 형으로 입력 받음

        triangle1.setlength(width1,height1);//밑변, 높이 setWidth 메서드를
이용해 초기화

        Tri triangle2 = new Tri(); // Tri 클래스 기본 생성자 사용하여 생성, 위와
같음
        System.out.print("삼각형 2 밑변: ");
        width2 = oInDev.nextDouble();
        System.out.print("삼각형 2 높이: ");
        height2 = oInDev.nextDouble();
        triangle2.setlength(width2, height2);

        oInDev.close();//Scanner 객체 닫음

        area1 = triangle1.getArea();//getArea 메서드로 넓이 할당받음
        area2 = triangle2.getArea();

        System.out.print("삼각형 1: " + area1 + " 삼각형 2: " + area2 + "로
");
        if (area1 > area2) {//넓이 비교
            System.out.println("삼각형 1 이 더 넓습니다!");
        } else if (area2 > area1) {
```

```
            System.out.println("삼각형 2 가 더 넓습니다!");
        } else {
            System.out.println("두 삼각형의 넓이가 같습니다.");
        }
    }
}
```

4. 결과 및 결과 분석

```
삼각형1 밑변: 5
삼각형1 높이: 7
삼각형2 밑변: 8
삼각형2 높이: 12
삼각형1: 17.5 삼각형2: 48.0로 삼각형2가 더 넓습니다!
```

먼저 triangle1 의 객체가 생성되고 밑변 입력 문구가 출력된다 5 를 입력한 뒤 높이에 7 을 입력하면 triangle1.setlength 함수가 밑변과 높이의 길이를 할당해준다. triangle2 도 똑같이 할당해준 뒤 getArea()메서드로 넓이를 area1, area2 에 할당한다. area1 에는 0.5*7*7 인 17.5 가 할당되고 area2 에는 0.5*8*12 인 48.0 이 할당된다. 조건문으로 넓이 비교 후 삼각형 2 가 더 넓다는 결과를 출력한다.

5. 소감

처음에는 기본생성자라는 조건을 보지 못하고 파라미터가 있는 생성자를 선언 후 입력 받은 값을 바로 할당했는데 기본 생성자를 사용하고 메서드를 통해 할당하는 방법을 사용하니 원래 잘 사용하지 않던 기본 생성자를 어떻게 활용해야 할지 알게 되었다. 또한 getArea 메서드를 통해 넓이를 구해 메서드를 어떻게 활용해야 효율적일지 알게되었다. 마지막으로 클래스를 두 개로 나누어 코드를 작성했기 때문에 접근 지정자에 대해 더욱 고민해보는 시간이었다.