

5. 제어 구조

1. 제어 구조의 이해

- 제품 생산 공정
- 구조적 프로그래밍 패러다임
 - 절차적 프로그래밍 패러다임의 하위 개념
 - goto 문을 사용하지 않고 프로그램을 3가지 제어 구조만으로 구성하는 프로그래밍 패러다임
 - 순차(sequence) 구조
 - 선택(selection) 구조
 - 반복(iteration) 구조
 - 프로그램 실행 흐름이 간결하고 작은 규모로 조직화하기 쉬움
- 순차 구조
 - 실행의 흐름을 주어지는 명령의 위치적 흐름에 따라 수행하는 구조
 - 명령 라인 위에서 아래로 흐르는 가장 직관적인 구조
- 선택 구조
 - 특정 영역 내의 명령문에 대한 실행 여부를 프로그램 실행 과정 중 결정하는 구조
 - 실행 여부는 조건에 따라 결정
- 반복 구조
 - 특정 영역의 명령문을 여러 번 재실행하는 구조
 - 반복 횟수를 조건에 따라 결정

2. 순차 구조

- 순차 구조
 - 실행의 흐름을 주어지는 명령의 위치적 흐름에 따라 수행하는 구조
 - 명령 라인 위에서 아래로 흐르는 가장 직관적인 구조
 - 일단 첫 단계를 시작하면 마지막 단계까지 수행
- 삼각형 출력하기
 - 다음과 같이 삼각형 모양으로 *을 출력하는 프로그램을 작성하시오.
- 원뿔 계산 프로그램 개선
 - 사용자로부터 반지름과 높이를 입력받고 계산
- 사용자 입력
 - input: 사용자로부터 데이터를 입력받는 함수
 - 입력 데이터를 문자 데이터 타입으로 반환
 - 함수의 파라미터는 입력 안내문의 목적으로 사용
 - rad = input() rad = input("반지름을 입력하세요:")
- 원뿔 계산 프로그램 개선
 - 사용자로부터 반지름과 높이를 입력받고 계산
- 원뿔 계산 프로그램 개선

2. 순차 구조 (2)

- 프로그래밍 에러
 - 설계 미숙, 결함 또는 문법 오류로 프로그램이 의도한 대로 결과를 생성하지 못하는 문제
 - 구문 오류(syntax error):
 - 문법 체계에 적합하지 않는 명령문 입력 시 발생
 - 실행 오류(runtime error)
 - 논리적으로 실행 불가능한 명령문 작성 시 발생
 - [실행 오류] TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'int'
 - 의미 오류(semantic error)
 - 의미적으로 잘못 해석되는 명령문 작성 시 발생
- 데이터 타입 변환
 - 데이터 타입을 다른 타입으로 전환
 - 문자열 타입으로 변환
 - str 함수

- 정수 타입으로 변환

- └ **int** 함수

- 소수 타입으로 변환

- └ **float** 함수

- **print** 함수의 확장

- 여러 개의 데이터를 단일 함수로 출력 가능

- 콤마(,)로 파라미터를 구분하여 입력

- 데이터 사이에 공백(기본값)이 자동으로 추가

- **sep** 옵션을 변경하여 공백 변경 가능

- 키워드 인자의 활용

- 원뿔 계산 프로그램 개선

- └ **print** 함수는 여러 개의 데이터를 출력 가능