메서드 : 하나의 특정 작업을 수행하는 일련의 문장들을 하나로 묶은 것

ex) 랜덤한 수를 입력, 콘솔에 입력 받기(Scanner), 콘솔에 출력하기(println,print)

- 장점 : 효율 좋게 잘 만든 메소드는 재사용성이 높다(print, Scanner 등)

메서드 : 자판기

입력 : 메뉴, 금액

출력 : 음료수

기능 : 두 정수의 합을 알려주는 메서드

↓

입력 : 이 기능을 실행할 때 필요한 정보, 두 정수 = int num1, int num2 / 정수만 사용

- 생략가능

ex) nextint(),

↓

계산 : 입력에 함축돼 있는 계산식으로 계산

↓

출력 : 이 기능을 실행하고 나오는 결과(계산 과정은 생략)

- 출력 결과가 불필요할 시 생략 가능 / 자료형만 사용

메서드 선언 방법

출력 이름(입력){

구현;

}

리턴타입 메서드명(매개변수){

구현;

}

int sum(int num,1, int num2){

int res = num1+num2;

return res;

}

메서드 : 하나의 특정 기능을 하도록 묶어 놓은 것

매개변수 : 기능을 실행하는데 있어서 필요한 정보

리턴타입 : 기능이 시행되고 나서 알려주는 정보

메서드명 : 기능을 잘 표현할 수 있는 이름으로 선언한다

순서

1. 메서드의 기능을 풀어 쓴다

2. 풀어 쓴 기능에서 매개변수를 정한다

3. 풀어 쓴 기능에서 리턴타입을 정한다

4. 메서드명을 정한다

5. 2, 3, 4에서 정한 내용을 토대로 메서드를 선언한다

6. 메서드를 구현한다

일반변수 vs 참조변수

일반변수 : 값을 저장하는 변수 - int num1; char op; double num2;

참조변수 : 주소를 저장하는 변수 - int []arr1;

매개변수의 원본이 바뀌려면 매개변수가 참조변수여야 한다

로또번호 생성 메서드

1. random : 최소값과 최대값 사이의 랜덤한 수를 생성해서 알려주는 메서드

2. search : 숫자가 배열의 cnt 개수에서 이

3. createRandArr : 배열의 최소값과 최대값 사이의 랜덤한 수로 채우고 성공여부를 알려주는 메서드

매개변수 : int min, int max, int[] arr

리턴타입 : boolean

메서드 - 하나의 특정 기능을 하도록 작성된 코드

매개변수(입력) - 기능을 실행하기 위한 정보

- 정해진 약속대로 입력

- 단, 자동형변환이 가능한 경우는 융통성을 발휘한다(문자 a, b등은 유니코드 값이 있기 때문에 정수로 변환하여 계산 가능)

- 매개변수는 최소한으로 정확히 입력한다(사용되지 않는 매개변수라도 출력값을 입력하지 않으면 에러가 발생한다)

리턴타입

- 기능이 종료되고 알려주는 정보

- 기능을 상세히 썼을 때 알려준다, 전달한다 등의 단어가 나오면 리턴타입이 필요

- 동작만 하고 부연 설명의 요구가 없으면 리턴타입이 필요없다

메서드의 특징

- 재사용이 용이하다

- 해당 코드를 반복 사용할 시 코드의 중복을 줄일 수 있다

- 프로그램의 구조화가 가능하다

기능 : ~를 더하세요

매개변수 : 두 정수

기능 : ~를 계산하세요

매개변수 : 두 정수, 연산자

기능 : 학생에게 두 정수의 합을 물어보는 기능

리턴타입 : 정수의 합을 저장할 자료형 int

기능 : 학생에게 두 정수의 합을 칠판에 적는 기능(콘솔에 출력)

- 답을 확인할 필요 없이 바로 콘솔에 출력하면 되기 때문에 리턴타입이 필요없다(void)

객체지향언어

- 객체에 초점을 맞춤

-> 복사를 하고 싶다 -> 복사기 설계도(클래스)를 이용해서 복사기(객체)를 만들고, 복사 기능(메서드)을 이용하여 복사해라

절차지향언어

- 순서에 초점을 맞춤

-> 복사를 하고 싶다 -> 복사하는 메서드를 만들어서 사용해라

재귀 메서드

- 자기 자신을 호출하는 메서드

- statckOverFlowError가 발성하면 재귀호출을 확인하라

오버로딩

- 한 클래스 안에서 메서드명이 동일한 메서드가 여러개일 경우

조건

- 매개변수의 개수가 다른 경우

- 매개변수의 개수가 같지만 자료형이 다른 경우

클래스 - 멤버변수+멤버메서드

접근 제한자 - public, default, private, protected(상속)

클래스 = 설계도

객체 = 제품

기능 = 메서드

클래스 객체명 = new 클래스();

객체명. 멤버변수;

객체명. 멤버메서드;

일반적으로

멤버변수는 private로 설정

멤버메서드는 public으로 설정