



# 정 보 처 리 기 사 실 기

기출풀이집 - 프로그래밍 언어편



# 목 차

1. Phyton ..... 1
2. Java ..... 6
3. C언어 ..... 25

**정처기 실기 (프로그래밍 언어편) 전범위 자료가**

**궁금하시다면?**

**여기**를 클릭해주세요!





안녕하세요! 꿈꾸는라이언입니다.

먼저 정처기 필기 합격을 진심으로 축하드려요! 🎉

이제 실기를 준비하시는 분들을 위해 [프로그래밍 언어편 요약 노트](#)를 준비했습니다.

요즘 정처기 실기 시험의 프로그래밍 언어 비중이 매우 높아졌어요. 전체 20문항 중 8~10문제가 프로그래밍 언어 관련 문제라니 절반에 가까운 수준이죠. 😱 따라서 프로그래밍 언어를 제대로 준비하지 않으면 아무리 이론을 잘 알아도 합격이 쉽지 않은게 현실입니다.

실기에서 다루는 프로그래밍 언어는 C언어, 자바, 파이썬인데요, **출제 비중은 C언어(3~4문제) > 자바(2~3문제) > 파이썬(1~2문제)** 순이에요. 그래서 점수의 비중으로는 **C언어가 가장 중요하고, 그 다음으로 자바와 파이썬 순이에요!** 하지만 이 노트에서는 오히려 **정말 유형도 적고 이해하기 쉬운 파이썬부터 시작해서 자바, C언어 순서로 정리했어요.** 처음 프로그래밍 언어를 접하시는 분들도 차근차근 따라오실 수 있게요! 😊

- 프로그래밍 언어에서는 기출 유형이 반복돼요!** 그래서 20년부터 23년도까지의 모든 프로그래밍 언어 기출문제를 분석하여 자주 출제되는 유형을 요약했어요. 단순 암기식 문제부터 응용력을 요구하는 문제까지 다양한 레벨의 문제가 있지만, 결국 **같은 유형이 반복되고 있었어요.** 이 유형만 확실히 잡으시면 프로그래밍 언어 문제는 충분히 고득점 받으실 수 있을 거예요! 👍
  - 모든 코드를 이해할 필요는 없어요!** 코드 한줄 한줄 이해를 하고 넘어가면 정말 좋지만, **문제 풀이에 있어서는 출제 유형을 빠르게 파악하는 것이 관건입니다.** 실제로 많은 문제들이 숫자만 바뀌어서 동일한 유형으로 출제되거든요. 예를 들어 완전수를 찾는 문제가 매번 나오는데, 사실 완전수는 100이하에서는 6과 28뿐이에요. 그런데 문제의 코드를 전부 해석하려 하면 손으로 100번까지 계산해야 할 수도 있어요. 😱 하지만 완전수의 특징만 알고 있다면 30초 안에 답을 찾을 수 있죠. 그러니 모든 코드를 이해하는데 집중하기보다는 먼저 어떤 유형의 문제인지를 캐치하는 것이 효과적인 전략이 될 거예요. 이를 위해 제가 정리한 분석을 통해 반복적으로 출제되는 유형을 정확히 파악하시는 걸 추천드려요!
- 추가로 각 기출 문제마다 아래와 같이 **난이도**를 표시했어요! 낮은 난이도부터 우선해서 학습해주세요!
- ★☆☆ : 반드시 맞춰야 하는 쉬운 문제
  - ★★☆ : 처음에는 다소 어렵지만 합격을 위해서는 도전해야 하는 문제
  - ★★★ : 실제 시험에서는 풀지 말아야할 정도로 어려운 문제

프로그래밍의 실행 순서가 복잡한 문제의 경우, 작은 숫자(①②...)로 코드 진행 순서를 각 코드 앞에 표기해놨어요!

- 유튜브에 풀이가 있어요!** 모든 문제에 대한 효율적인 풀이 및 코드에 대한 한 줄 한 줄 설명을 적었지만, 이해가 어려운 경우에는 모든 문제에 기출 연도와 회차를 적어두었으니 유튜브에 검색해서 자세한 설명을 보시는 것도 가능해요! 그리고, 프로그래밍 기출 문제는 사람들이 각자 복원한 문제이기에 **정리마다 조금씩 코드가 다를 수는 있어요.** 하지만, 문제의 유형은 동일하니 걱정하지 않으셔도 됩니다.

이 <정처기 실기 요약노트 - 프로그래밍 언어편>으로 공부하시면서 추가 궁금하신 사항이나 의견 있으실 분들을 위해 꿈꾸는라이언 공식 카페에 '**정처기 실기 질의응답 게시판**'을 별도 만들었어요! 카페 가입하시고 질문 남겨주시면 본 요약노트 제작에 많은 도움을 준 협업 개발자 분께서 바로는 아니더라도 틈틈이 답변해주실꺼에요! 😊👍 문의주신 모든 분들께 최대한 가능한 선에서 성심껏 빠르게 답변해드린다고 하니 참고해주세요!

⭐ **꿈꾸는라이언 네이버 카페** : <https://cafe.naver.com/dreamingryan>

그럼 모두들 정처기 합격을 시작으로 각자 계획하셨던 목표와 소중한 꿈을 이루시길 진심으로 응원하겠습니다.

다들 포기하지 마시고, 끝까지 함께 달려봐요!

합격 미리 축하드립니다!! ❤️



## 이것만 이해해도 문제 풀이에 정말 큰 도움이 돼요

## ★ 집합(Set)

집합은 순서가 없고 중복을 허용하지 않는 자료형이에요.

아래와 같이 사용할 수 있어요.

## # 집합 생성

```
fruit_basket = {"apple", "banana", "grape"}
```

## # 집합에 요소 추가

```
fruit_basket.add("orange")
```

```
print(fruit_basket) → 출력: {"apple", "banana", "grape", "orange"}
```

## # 집합에서 요소 제거

```
fruit_basket.remove("banana")
```

```
print(fruit_basket) → 출력: {"apple", "grape", "orange"}
```

## # 집합 업데이트

```
more_fruits = ["kiwi", "pear", "apple"]
```

```
fruit_basket.update(more_fruits)
```

```
print(fruit_basket) → 출력: {"apple", "grape", "orange", "kiwi", "pear"}
```

## ★ 비트 연산자(Bitwise Operator)

2진수 형태로 데이터를 다룰 때 사용해요.

비트 연산자를 이해하기 쉬운 예시로는 "전등 스위치"를 들 수 있어요.

전등 스위치는 켜짐(1)과 꺼짐(0), 두 가지 상태만 있어요.

\*여기서 켜짐은 True, 꺼짐은 False를 의미해요.

AND 연산(&)은 두 개의 스위치가 모두 켜져 있을 때만 전등이 켜지는 거예요.

OR 연산(|)은 두 개의 스위치 중 하나라도 켜져 있으면 전등이 켜지는 거예요.

XOR 연산(^)은 두 개의 스위치 중 하나만 켜져 있을 때 전등이 켜지는 거예요.

(서로의 결과가 달라야 전등이 켜진다는 의미에요)

아래와 같이 사용할 수 있어요. 내가 연산자를 보고 계산할 수 있으면 돼요.

## # AND 연산

```
result = 5 & 3 → 101 & 011 = 001
```

```
print(result) → 출력: 1
```

## # OR 연산

```
result = 5 | 3 → 101 | 011 = 111
```

```
print(result) → 출력: 7
```

## # XOR 연산

```
result = 5 ^ 3 → 101 ^ 011 = 110
```

```
print(result) → 출력: 6
```

## # 비트 시프트 연산

왼쪽으로 한칸 이동은 맨 오른쪽에 0하나를 추가한다는 의미에요.

```
result = 5 << 1 → 101 << 1 = 1010
```

```
print(result) → 출력: 10
```

오른쪽으로 한칸 이동은 맨 오른쪽 값을 지운다는 의미에요.

여기서 2진수 010과 10은 같은 2를 뜻해요.

```
result = 5 >> 1 → 101 >> 1 = 010
```

```
print(result) → 출력: 2
```

## ★ split()

문자열을 지정된 구분자를 기준으로 분할하여 리스트로 반환합니다.

기본 구분자는 공백입니다.

## # 공백을 기준으로 문자열 분할

```
sentence = "Hello world! How are you?"
```

```
words = sentence.split() → sentence.split(" ") 과 동일한 표현
```

```
print(words) → 출력: ['Hello', 'world!', 'How', 'are', 'you?']
```

## ★ 슬라이싱(Slicing)

리스트나 문자열에서 일부분을 잘라내는 것을 말해요.

Index의 시작점은 0부터 시작해요.

끝점의 index는 자를 때 포함하지 않아요.

[0:5]은 첫 번째 문자부터 시작해서 0, 1, 2, 3, 4까지의 문자를 뜻해요.

처음부터 자르고 싶으면 시작점을 생략할 수 있어요.

[:]

끝까지 자르고 싶으면 끝점을 생략할 수 있어요.

[8:]

## # 문자열 슬라이싱

```
message = "Hello, World!"
```

```
print(message[0:5]) → 출력: "Hello"
```

```
print(message[:5]) → 출력: "Hello"
```

```
print(message[7:]) → 출력: "World!"
```

```
print(message[7:12]) → 출력: "World"
```

```
print(message[::-2]) → 출력: "Hlo ol!" → 홀수번째 문자열 추출!
```

(2 page 2번 문제 참고)

## # 리스트 슬라이싱

```
numbers = ["a", "b", "c", "d", "e"]
```

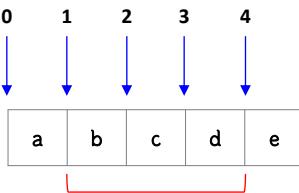
```
print(numbers[1:4]) → 출력: ["b", "c", "d"]
```

```
print(numbers[:3]) → 출력: ["a", "b", "c"]
```

```
print(numbers[3:]) → 출력: ["d", "e"]
```

```
print(numbers[::-2]) → 출력: ["a", "c", "e"]
```

→ 홀수번째 문자열 추출!  
(2 page 2번 문제 참고)



numbers[1:4] 슬라이싱 시,

인덱스 1에서 시작해서

인덱스 4 앞까지 추출

(인덱스 3까지만 포함)

## ★ 문자열 포맷팅(String Formatting)

문자열 내에 특정 값을 삽입하는 방법이에요.

% 연산자를 사용하여 문자열 포맷팅을 할 수 있어요.

## # % 연산자

```
name = "Alice"
```

```
age = 25
```

```
print("My name is %. I'm %d years old." % (name, age))
```

My name is Alice. I'm 25 years old.

## ★ map()

리스트나 튜플 등의 이터러블 객체의 각 요소에 함수를 적용하여 새로운 이터러블 객체를 반환합니다.

map()은 원본 데이터를 변경하지 않고 새로운 데이터를 생성합니다.

```
numbers = [1, 2, 3, 4, 5]
```

# lambda 함수를 사용하여 각 요소에 10을 더하기

```
numbers = [1, 2, 3, 4, 5]
```

```
add_ten = map(lambda x: x + 10, numbers)
```

```
print(list(add_ten)) → 출력: [11, 12, 13, 14, 15]
```



▶ 정답 split

## 1. [기출 예제] 23년 3회 실기 (난이도: ★★★)

다음 Python 프로그램과 그 &lt;실행결과&gt;를 분석하여 팔호에 들어갈 알맞은 예약어를 쓰시오.

첫 번째 라인의 '5 10'은 입력 받은 값에 해당한다.

&lt;실행결과&gt;

x, y의 값을 공백으로 구분하여 입력: 5 10

x의 값: 5

y의 값: 10

정답은 split 이에요

split() 메서드는 문자열을 특정 구분자를 기준으로 나누어 리스트로 반환해요.  
기본적으로는 공백을 구분자로 사용해요.

split() 메서드는 사용자 입력을 받아 분리할 때 자주 사용돼요.

```
1 | x, y = input("x, y의 값을 공백으로 구분하여 입력: ") .( 빈 칸 ) (" ") → input() 함수는 사용자로부터 입력을 받는 기능을 해요.
2 | print("x의 값: ", x)
3 | print("y의 값: ", y)
```

## ★ key point

사용자가 5 10 입력한 것을 실행결과에서 알려주고 있어요.

"5 10"이라는 값을 x, y로 공백 기준으로 값을 할당하기 위해서는 공백 기준으로 문자열을 나눠야해요.

문자열을 특정 구분자로 나누는 키워드는 split이에요.

그래서 정답이 split이 돼요.

\*구분자를 넘겨주지 않는 split()은 split(" ")과 동일해요. 즉, 공백 문자열이 기본값이에요.

## 2. [기출 예제] 23년 2회 실기 (난이도: ★★★)

다음 파이썬 코드에서 알맞는 출력값을 작성하시오.

▶ 정답 engneing

```
1 | a = "engineer information processing"
2 | b = a[:3]
3 | c = a[4:6]
4 | d = a[28:]
5 | e=b+c+d
6 | print(e)
```

## ★ 슬라이싱 (자주 쓰이니 외우면 좋아요)

sequence[start:end:step]의 형태로 사용돼요.

start는 슬라이스가 시작되는 인덱스이고,  
end는 슬라이스가 끝나는 인덱스이며,

(해당 인덱스는 문자열에서 제외)

step은 슬라이스가 진행되는 간격이에요.

start와 end는 생략할 수 있고, 생략하면 각각  
시퀀스의 시작과 끝이 기본값으로 사용돼요.

step도 생략할 수 있고,

생략하면 기본값 1이 사용돼요.

변수 a에 문자열 "engineer information processing"를 할당해요.

a 문자열의 처음부터 3번째 문자까지 가져와 b에 저장합니다.  
여기서 a[:3]은 a의 인덱스 0부터 인덱스 2까지를 의미해요.

따라서 b에는 "eng"가 저장됩니다.

a 문자열에서 5번째부터 6번째 문자까지 가져와 c에 저장합니다.  
(인덱스 4부터 인덱스 5까지)

여기서 a[4:6]은 "ne"을 의미해요.

변수 d에 a의 29번째 문자부터 끝까지의 슬라이스를 할당해요.

슬라이싱에서 두 번째 인덱스를 생략하면 문자열의 끝까지 슬라이스 해요.  
a의 29번째 문자부터 끝까지는 "ing" 부분 문자열 이에요.

따라서 d에는 "ing"이 할당돼요.

변수 e에 b, c, d를 연결한 문자열을 할당해요.  
b는 "eng", c는 "ne", d는 "ing"이므로,

e에는 "engneing"이 할당돼요.

[6행] 변수 e의 값인 "engneing"을 출력해요.

## ★ 집합 (자주 쓰이니 외우면 좋아요)

중복되지 않는 요소들의 모음이에요.

같은 값을 여러 번 추가해도 집합에는 한 번만  
저장돼요.

순서가 없어요.

집합의 요소들은 특정한 순서를 가지지 않아요.

## 3. [기출 예제] 23년 1회 실기 (난이도: ★★★)

다음 파이썬 코드에서 알맞는 출력값을 작성하시오.

▶ 정답 {'한국', '중국', '베트남', '홍콩', '태국'}

```
1 | a = {'한국', '중국', '일본'}
2 | a.add('베트남')
3 | a.add('중국')
4 | a.remove('일본')
5 | a.update({'홍콩', '한국', '태국'})
6 | print(a)
```

최종적인 a 집합을 출력해요.

정답은 {'한국', '중국', '베트남', '홍콩', '태국'}이에요

a 집합에 다른 집합의 요소들을 추가해요.

update() 메서드는 인자로 주어진 집합의 모든 요소를 a에 추가해요.

'한국'은 이미 a에 존재하므로 중복 추가되지 않아요.

a라는 이름의 집합을 생성하고 초기값으로 '한국', '중국', '일본'을 갖게 해요.

a 집합에 '베트남'을 추가해요.  
집합에 새로운 요소를 추가할 때는 add() 메서드를 사용해요.  
이 연산 후에 a는 {'한국', '중국', '일본', '베트남'}이 돼요.

a 집합에 '중국'을 추가하려고 해요.

하지만 '중국'은 이미 a에 존재하는 요소예요.

집합은 중복된 요소를 허용하지 않기 때문에

이 연산은 a를 변화시키지 않아요.

이 연산 후에도 a는 여전히 {'한국', '중국', '일본', '베트남'}이에요.

a 집합에서 '일본'을 제거해요.

집합에서 특정 요소를 제거할 때는 remove() 메서드를 사용해요.

이 연산 후에 a는 {'한국', '중국', '베트남'}이 돼요.



## 4. [기출 예제] 22년 3회 실기 (난이도: ★★★)

▶ 정답 [101,102,103,104,105]

다음 파이썬 코드에 대한 출력값을 작성하시오.

```
1 TestList = [1,2,3,4,5] → TestList라는 이름의 리스트를 생성하고 초기값으로 [1,2,3,4,5]를 할당해요.
2 TestList = list(map(lambda num : num + 100, TestList)) → 결과적으로, TestList의 각 요소에 100을 더한 새로운 리스트를 생성해요.
3
4 print(TestList) →
```

map() 함수는 첫 번째 인자로 함수를, 두 번째 인자로 이터러블(리스트 등)을 받아요.  
map() 함수는 이터러블의 각 요소에 함수를 적용한 결과를 반환해요.

여기 코드에서는 람다 표현식 `lambda num : num + 100`이 `map()` 함수의 첫 번째 인자로 사용되었어요.  
이 람다 표현식은 인자 `num`을 받아 `num + 100`을 반환하는 함수예요.  
`map()`의 두 번째 인자는 `TestList`가 전달되었어요.

따라서, `map()`은 `TestList`의 각 요소에 람다 함수를 적용해요.

1 + 100 = 101  
2 + 100 = 102  
3 + 100 = 103  
4 + 100 = 104  
5 + 100 = 105

`map()`의 결과는 맵 객체(map object)로 리스트로 변환하기 위해 `list()` 함수를 사용했어요.  
최종적으로 `TestList`에는 [101, 102, 103, 104, 105]가 할당돼요.

→ 정답은 [101, 102, 103, 104, 105] 이 출력돼요.

## 5. [기출 예제] 22년 2회 실기 (난이도: ★★★)

▶ 정답 REMEMBER AND STR

다음 파이썬 코드에 대한 출력값을 작성하시오.

```
1 a = "REMEMBER NOVEMBER" → a라는 변수에 문자열 "REMEMBER NOVEMBER"를 할당해요.
2 b = a[:3] + a[12:16]; → 문자열 슬라이싱을 사용하여 새로운 문자열을 만들어 b에 할당하는 부분이에요.
3 c = "R AND %s" % "STR"; → a[3]은 a의 처음부터 3번째 문자까지의 슬라이스로, "REM"이에요.  

   (인덱스 0부터 인덱스 2까지)  

   a[12:16]은 a의 13번째 문자부터 16번째 문자까지의 슬라이스로, "EMBE"예요.  

   (인덱스 12부터 인덱스 15까지)
```

두 슬라이스를 + 연산자로 연결하면 "REMEMBE"가 돼요.  
따라서 b에는 "REMEMBE"가 할당돼요.  
→ 문자열 포맷팅을 사용하여 새로운 문자열을 만들어 c에 할당하는 부분이에요.  
"R AND %s"는 포맷 문자열로, %s 부분에 다른 문자열이 들어갈 자리예요.  
% 연산자 뒤에 오는 "STR"이 %s 자리에 들어가요.  
따라서 c에는 "R AND STR"이 할당돼요.

→ b와 c를 연결한 문자열을 출력하는 부분이에요.  
b는 "REMEMBE", c는 "R AND STR"이에요.  
두 문자열을 + 연산자로 연결하면 "REMEMBER AND STR"이 돼요.

정답은 REMEMBER AND STR 이에요.

## 6. [기출 예제] 22년 1회 실기 (난이도: ★★★)

▶ 정답 a= 20 b= 2

다음 파이썬 코드에 대한 출력값을 작성하시오.

```
1 def exam(num1, num2=2): → 이 코드의 특징은 함수의 매개변수에 기본값을 지정하는 방법을 보여줘요.
2     print('a=', num1, 'b=', num2)
3 exam(20) → 함수 정의에서 num2=2와 같이 매개변수에 =로 값을 할당하면, 그 값이 해당 매개변수의 기본값이 돼요.  

   함수를 호출할 때 해당 매개변수에 값을 전달하지 않으면, 기본값이 사용돼요.  

   여기서는, exam(20)은 num1에 20을 전달하고 num2에는 아무것도 전달하지 않았어요.  

   따라서, num1은 20, num2는 기본값 2가 사용돼요.
```

그러므로 이 코드의 출력 결과는 a= 20 b= 2 에요.



▶ 정답 False

## 7. [기출 예제] 21년 3회 실기 (난이도: ★★★)

다음 파이썬 코드에 대한 출력값을 작성하시오.

```
1 a, b = 100, 200 → a에 100을, b에 200을 할당하는 거예요.  
2 print(a==b) → 이렇게 하면 a와 b에 각각 다른 값이 저장돼요.
```

`==`은 두 값이 같은지 비교하는 연산자예요.  
만약 a와 b가 같다면 `True`를,  
다르다면 `False`를 결과로 돌려줘요.

a는 100, b는 200이니까  
당연히 둘은 같지 않아요.  
그래서 `a==b`의 결과는 `False`가 되는 거죠.

## 8. [기출 예제] 21년 2회 실기 (난이도: ★★★)

다음 파이썬 코드에 대한 출력값을 작성하시오.

```
1 a = 100 → a에 100을 할당하고,  
2 result = 0 → result에 0을 할당했어요.  
3 for i in range(1,3): → 이렇게 하면 a는 100, result는 0으로 초기화되는 거예요.  
4     result = a >> i → for 루프를 사용해서 i를 1부터 2까지 반복해요.  
5     result = result + 1 → range(시작 숫자, 끝 숫자) 형태를 사용하는데, 이때 끝 숫자는 포함되지 않습니다.  
6 print(result) → 슬라이스와 비슷하게 인덱스 시작 숫자 이상 인덱스 끝 숫자 - 1 이하 까지를 뜻해요.
```

▶ 정답 26

루프 안에서는 아래 로직을 수행해요.

result = a &gt;&gt; i는 a를 오른쪽으로 1번 시프트한 결과를 result에 할당해요.

&gt;&gt;는 비트 오른쪽 시프트 연산자예요.

이 연산자는 a의 비트를 오른쪽으로 1번 이동시켜요.

초조 어려울 수 있어요.

첫 번째 반복(`i=1`)에서 `a >> 1`은 a의 비트를 오른쪽으로 1번 이동시키는 거예요.2진수로 표현하면 `1100100`(기준 100)에서 `0110010`(= `110010`)(비트 이동)이 되죠.이것은 10진수로 `50`이에요.여기에 1을 더하면 `51`이 돼요.두 번째 반복(`i=2`)에서는 `a >> 2`로 a의 비트를 오른쪽으로 2번 이동시켜요.2진수로 `1100100`(`50`)에서 `0011001`(= `11001`)이 되는데, 이것은 10진수로 `25`예요.여기에 1을 더하면 `26`이 돼요.루프가 끝나면 `print(result)`로 최종 result의 값을 출력해요.정답은 `26`이에요.

## 9. [기출 예제] 21년 1회 실기 (난이도: ★★★)

다음 파이썬 코드에 대한 출력값을 작성하시오.

▶ 정답 skiddp

```
1 class good : good이라는 클래스를 정의했어요.  
2     li = ["seoul", "kyeonggi", "inchon", "daejeon", "daegu", "pusan"]  
3  
4 g = good() → good 클래스의 인스턴스를 생성해서 g라는 변수에 할당하는 거예요.  
5 str01 = '' → 이렇게 하면 g를 통해 good 클래스의 속성과 메서드에 접근할 수 있게 돼요.  
6 for i in g.li: → 빈 문자열을 str01이라는 변수에 할당하는 거예요.  
7     str01 = str01 + i[0] → 이 변수는 앞으로 도시 이름의 첫 글자를 모으는 데 사용될 거예요.  
8  
9 print(str01) → for 루프를 사용해서 g.li의 각 요소를 i라는 변수에 하나씩 할당하면서 반복해요.  
→ 여기서는 순서대로, seoul, kyeonggi … 이 값들이 i에 할당돼요.
```

`str01 = str01 + i[0]`은 현재 `str01`의 값에 `i`의 첫 번째 문자(`[0]`)를 더하는 거예요.  
문자열에서 배열처럼 index로 문자 하나씩 가져올 수 있어요.

첫 번째 반복에서는 "seoul"의 첫 글자 "s"가 `str01`에 더해져요.두 번째 반복에서는 "kyeonggi"의 첫 글자 "k"가 `str01`에 더해져요.이런 식으로 계속 반복되면서 `str01`에는 각 도시 이름의 첫 글자가 모이게 돼요.

루프가 끝나면 `print(str01)`로 최종 `str01`의 값을 출력해요.  
정답은 `skiddp`이에요.

## ★ 연산자 종류

관계	<code>==</code>	같다	>	크다	<	작다
	<code>!=</code>	같지 않다	<code>&gt;=</code>	크거나 같다	<code>&lt;=</code>	작거나 같다
비트	<code>&amp;</code>	and		모든 비트가 1이면 1		
	<code>^</code>	xor		모든 비트가 같으면 0, 하나라도 다르면 1		
	<code> </code>	or		모든 비트 중 하나라도 1이면 1		
	<code>~</code>	not		각 비트의 부정 (0이면 1 산출)		
	<code>&lt;&lt;</code>	-		비트를 왼쪽으로 이동		
	<code>&gt;&gt;</code>	-		비트를 오른쪽으로 이동		
논리	<code>!</code>	not	<code>&amp;&amp;</code>	and	<code>  </code>	or



▶ 정답  
[1, 2, 3]  
7  
123  
45  
6789

## 10. [기출 예제] 20년 4회 실기 (난이도: ★★★)

다음 파이썬 코드에 대한 출력값을 작성하시오.

```

1  lol = [[1,2,3],[4,5],[6,7,8,9]]           → 리스트들을 요소로 가지고 있는 2차원 리스트예요.
2  print(lol[0])                            → lol의 첫 번째 요소를 출력하는 거예요. lol[0]은 [1,2,3]이므로, 이 코드는 [1,2,3]을 출력해요.
3  print(lol[2][1])                          → 세 번째 요소의 두 번째 요소를 출력하는 거예요.
4  for sub in lol:                         → lol[2]는 [6,7,8,9]이고, lol[2][1]은 7이므로, 이 코드는 7을 출력해요.
5    for item in sub:                      → 중첩 for 루프를 사용하고 있어요.
6      print(item, end = ' ')            → 바깥쪽 루프에서는 lol의 각 요소(리스트)를 sub라는 변수에 하나씩 할당하면서 반복해요.
7    print()                                → 안쪽 루프에서는 각 sub의 요소를 item이라는 변수에 하나씩 할당하면서 반복해요.

안쪽 루프가 끝나면 print()를 사용해서 줄바꿈을 해요.
이렇게 하면 각 리스트의 요소들이 한 줄씩 출력되는 거죠.
item 기준으로는 한줄로 출력하고,
Sub 기준으로는 한줄씩 내려서 출력돼요.

item에 이라는 파이썬 키워드가 있는 건 아니에요.
lol, sub과 동일한 변수명이에요!

```

▶ 정답 [1, 2, 3]  
7  
123  
45  
6789

첫번째 배열인 [1,2,3]는 123  
두번째 배열인 [4,5]는 45  
세번째 배열인 [6,7,8,9]는 6789  
이렇게 출력돼요.

## 11. [기출 예제] 20년 2회 실기 (난이도: ★★★)

다음 파이썬 코드에 대한 출력값을 작성하시오.

▶ 정답 {'중국', '한국', '베트남', '홍콩', '태국'}

```

1  a={'일본', '중국', '한국'}           → 집합(set) 자료형을 다루고 있어요. 집합은 순서가 없고, 중복된 요소를 허용하지 않는 자료형이에요.
2  a.add('베트남')                     → 자주 나오고 있어요. 중복되지 않는다는 사실 알고 있으면 좋아요.
3  a.add('중국')                      → 중괄호 {}를 사용하여 생성할 수 있고, 빈 집합을 생성할 때는 set()을 사용해요.
4  a.remove('일본')                   → a 집합에 '베트남'을 추가하는 거예요. 이 연산 후에 a는 {'일본', '중국', '한국', '베트남'}이 돼요.
5  a.update(['홍콩', '한국', '태국'])   → a 집합에 '중국'을 추가하려고 하는 거예요. 하지만 '중국'은 이미 a에 존재하는 요소예요.
6  print(a)                           → 집합은 중복된 요소를 허용하지 않기 때문에, 이 연산은 a를 변화시키지 않아요.

a 집합에서 '일본'을 제거하는 거예요. 이 연산 후에 a는 {'중국', '한국', '베트남'}이 돼요.

a 집합에 '홍콩', '한국', '태국'을 추가하는 거예요.
update() 메서드는 다른 집합이나 리스트 등의 이터러블 객체를 인자로 받아서,
그 안의 모든 요소를 집합에 추가해요. 이 때, '한국'은 이미 a에 존재하므로 중복 추가되지 않아요.
이 연산 후에 a는 {'중국', '한국', '베트남', '홍콩', '태국'}이 돼요.

▶ 최종적인 a 집합을 출력하는 거예요.
{'중국', '한국', '베트남', '홍콩', '태국'}이 출력돼요.

```



## [문제 유형] equals

## ★ == 연산자

`==`는 기본 데이터 타입(예: int, float, char 등)의 값을 비교할 때 자주 사용돼요. 두 값이 같은지 확인하려면 `==`를 사용하면 되죠.

참조 데이터 타입(예: 객체)에 대해서 `==`를 사용하면, 그것들이 같은 메모리 주소를 가리키고 있는지 확인합니다. 즉, 두 변수가 같은 객체를 참조하고 있는지를 봐요.

```
String text1 = new String("hello");
String text2 = new String("hello");
System.out.println(text1 == text2); // false, 왜냐하면 메모리 주소가 다르니까요.
```

여기서 `text1`과 `text2`는 같은 내용의 문자열을 가지고 있지만, 각기 다른 객체이므로 `==` 연산자는 `false`를 반환합니다.

## ★ equals 메서드

`equals` 메서드는 객체의 내용이 같은지를 확인하는 데 사용되요.

예를 들어, `String` 클래스에서 `equals` 메서드는 문자열의 내용이 같은지를 비교합니다.

```
String text1 = new String("hello");
String text2 = new String("hello");
System.out.println(text1.equals(text2)); // true, 내용이 같기 때문이에요.
```

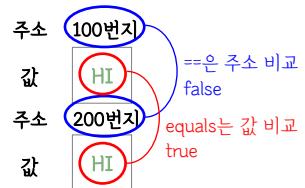
여기서 `text1`과 `text2`의 내용은 같기 때문에 `equals` 메서드는 `true`를 반환해요.

간단히 말하면, `==`는 두 객체가 실제로 같은 객체인지, 즉 메모리 상에서 같은 위치를 참조하는지를 확인하는 반면, `equals` 메서드는 두 객체가 '논리적으로' 동등한지를 비교하는 데 사용돼요.

## 1. [기출 예제] 23년 2회 실기 (난이도: ★★★)

다음은 자바에 대한 문제이다. 알맞은 출력 값을 작성 하시오.

```
1 public class Main {
2     public static void main(String[] args) {
3
4         String str1 = 'Programming';
5         String str2 = 'Programming';
6         String str3 = new String('Programming'); → [4행] [5행] 이 들은 하나의 객체가 생성되고 자바에서 같은 객체를 할당해줍니다.
7
8         println(str1==str2) → str1와 str2의 객체 주소는 동일해요
9         println(str1==str3) → str1와 str3의 객체 주소는 달라요. new를 통해 새롭게 생성했기 때문이에요.
10        println(str1.equals(str3)) → str1과 str3의 값을 비교해요. 둘의 문자열은 동일해요
11        print(str2.equals(str3)) → str2과 str3의 값을 비교해요. 둘의 문자열은 동일해요
12    }
13 }
14 }
```



▶ 정답  
true  
false  
true  
true



## [문제 유형] 단순 for문

## ★ for 문

for 문은 일정한 규칙에 따라 반복해서 명령을 실행할 때 사용해요.

for 문은 보통 세 부분으로 구성되어 있어요.

**초기화:** 반복을 시작하기 전에 처음에 설정해야 하는 부분이에요. 여기서 반복할 때 사용할 변수를 설정합니다.

**조건:** 반복을 계속할지 말지를 결정하는 조건이에요. 이 조건이 참이면 반복을 계속하고, 거짓이면 반복을 멈춰요.

**증감:** 반복할 때마다 실행되는 부분으로, 보통 반복하는 변수를 증가시키거나 감소시키는데 사용돼요.

```
for(int i = 0; i < 5; i++) {
    System.out.println("반복 중: " + i);
}

// 반복 중: 0
// 반복 중: 1
// 반복 중: 2
// 반복 중: 3
// 반복 중: 4
```

**int i = 0;** - 반복을 시작하기 전에 변수 i를 0으로 초기화해요.

**i < 5;** - i가 5보다 작은 동안에는 반복을 계속해요. i가 5에 도달하면 반복을 멈추죠.

**i++** - 매 반복마다 i의 값을 1씩 증가시켜요.

결과적으로 이 for 문은 "반복 중: 0"부터 "반복 중: 4"까지 총 5번의 메시지를 출력하게 됩니다.

▶ 정답 0123

## 2. [기출 예제] 20년 1회 실기 (난이도: ★★★)

다음은 자바 소스코드이다. 출력 결과를 쓰시오.

```
1 public class Main {
2     public static void main(String[] args) {
3         int i;
4         int a[] = {0,1,2,3}; → 4 크기인 배열 a를 생성했어요.
5         for(i=0; i<4; i++) { → i가 0,1,2,3 까지 반복해요. (4보다 작을 때까지)
6             System.out.print(a[i] + " "); → a 배열의 0번째, 1번째, 2번째, 3번째 값을 출력해요.
7         }
8     }
9 }
```

차례대로 0 1 2 3이 출력돼요.

▶ 정답 0123

## 3. [기출 예제] 22년 3회 실기 (난이도: ★★★)

아래 자바 코드에서 출력되는 값을 작성 하시오.

```
1 class Test {
2     static int[] mkarr() {
3         int[] tmpArr = new int[4]; → 크기가 4인 정수형 배열 tmpArr를 선언하고 생성해요.
4         for (int i = 0; i < tmpArr.length; i++) { → 0부터 크기가 4인 정수형 배열의 길이보다 작을 때(4미만)까지 반복
5             tmpArr[i] = i; → 0번째 배열에는 0
6         }
7         return tmpArr; → 0 1 2 3 배열 반환!
8     }
9
10
11
12     public static void main(String[] args) {
13         int[] arr; → 배열 객체를 선언만 하고, 할당은 [14행]에서 해요.
14         arr = mkarr(); → 위 {0, 1, 2, 3} 배열을 할당
15         for (int i = 0; i < arr.length; i++) {
16             System.out.print(arr[i]); → [15행]을 보면, 0부터 배열을 출력하는데
17         }
18     }
19 }
```

정답은 0123 이에요.



▶ 정답 61

## 4. [기출 예제] 22년 2회 실기 (난이도: ★★★)

아래 자바 코드에서 출력되는 값을 작성하시오.

실제 실행코드	<pre> 1 class Main { 2     public static void main(String[] args) { 3         ①Cond obj = new Cond(3); 4         ④obj.a=5; 5         ⑤int b = obj.func(); 6         ⑦System.out.print(obj.a + b); 7     } 8 }</pre>	<p>Cond 객체 생성 및 생성자에 3을 넘겨주고 있어요. Cond 안의 멤버변수 a에 3이 할당돼요.      하지만, [4행]에서 바로 a의 값을 5로 덮어쓰고 있어요.      [16행]이 실행돼요. 56이 반환돼요. (로직은 아래에서 설명할게요)      obj.a + b는 5 + 56이므로, 정답은 61이에요.</p>
Cond 클래스	<pre> 10 class Cond { 11     int a; 12     ② public Cond(int a) { 13         ③ this.a = a; 14     } 15 16     ⑥ public int func() { 17         int b = 1; 18         for (int i = 1; i &lt; a; i++) { 19             b += a * i; 20         } 21         return a + b; 22     } 23 }</pre>	<p>매개변수 a를 받아서 Cond 멤버변수 a에 할당해요. 여기서는 [3행]에 의해 3이 할당되었다가 [4행]에 의해 5가 되었어요.      b의 시작값 1이에요.      1부터 a 미만까지 +1 증가하면서 반복해요.      여기서 a는 5여서, 1~4까지 반복해요.      a * i는 (5 * 1) 한 값을 b에 더해줘요.      순서대로 를 반복해서 b에 더해줘요.      즉 b의 시작값 1 + (5 * 1) + (5 * 2) + (5 * 3) + (5 * 4) 이 b에 할당돼요.      b는 51이에요.      func() 메소드의 반환은 a + b이니, 5 + 51을 반환하면 56가 반환되요.</p>

## 5. [기출 예제] 22년 3회 실기 (난이도: ★★★)

아래 자바 코드에서 출력되는 값을 작성하시오.

실제 실행코드	<pre> 1 class Exam{ 2     public static void main(String[] args){ 3         int a = 0; → 정수형 변수 a를 선언하고 0을 할당합니다. 4 5         for(int i = 1; i &lt; 999; i++) { → 1부터 998까지 (999미만) +1씩 증가하면서 반복해요. 6             if(i % 3 == 0 &amp;&amp; i % 2 == 0) → %는 나머지를 뜻해요. 7                 a = i; 8         } 9 10        System.out.print(a); 11    } 12 }</pre>	<p>즉, i를 3으로 나누고 남은 나머지가 0이면서, i를 2로 나누고 남은 나머지가 0인 경우에 a에 i를 할당해라 라는 의미에요.      이 말만 보면 3의 배수이면서 2의 배수인 6의 배수를 찾으라는 말인데요!      6의 배수마다 a에 계속해서 덮어쓰는 코드이니, 사실상 출력되는 값은 1000에서 가장 가까운 6의 배수가 되겠네요!</p>
------------	--	---

1000은 3의 배수가 아니고, 999는 2의 배수가 아니고, 998은 3의 배수가 아니고,  
 997은 2의 배수가 아닙니다. 996는 3의 배수 면서 2의 배수입니다!  
 그러므로, 정답은 996이 됩니다!



▶ 정답 24513

## 6. [기출 예제] 22년 3회 실기 (난이도: ★★★)

아래 자바 코드에서 출력되는 값을 작성하시오.

```

1 public class Test {
2     public static void main(String[] args) {
3         int result[] = new int[5]; →
4
5         int arr[] = { 77, 32, 10, 99, 50 }; → 크기가 5인 정수형 배열 arr을 선언하고, 초기값으로 77, 32, 10, 99, 50을 저장합니다.
6
7         for (int i = 0; i < 5; i++) {
8             result[i] = 1; →
9                 for (int j = 0; j < 5; j++) { j 가 0 부터 4까지
10                if(arr[i] < arr[j]) { + 1씩 증가하면서
11                    result[i]++; 반복해요.
12                }
13            }
14
15            for (int k = 0; k < 5; k++) {
16                System.out.print(result[k]);
17            }
18        }
19    }
20 }
21 }
```

실제 실행코드

n[0] n[1] n[2] n[3] n[4]

비어 있는 배열 5칸이 생성돼요.

크기가 5인 정수형 배열 result를 선언하고 생성해요.

배열은 같은 타입의 데이터를 연속적으로 저장하는 자료구조입니다.

배열의 크기는 생성 시에 결정되며, 이 경우 result 배열은 5개의 정수를 저장할 수 있어요.

n[0] n[1] n[2] n[3] n[4]

77 32 10 99 50

배열 선언과 동시에 할당해요.  
이걸 초기화라고 불러요.

n[0] n[1] n[2] n[3] n[4]

i=0 1



n[0] n[1] n[2] n[3] n[4]

i=1 1 1

이중 반복문으로  
i 가 0일 때, j 가 0 ~ 4까지 반복하고,  
i 가 1일 때, j 가 0 ~ 4까지 반복하고,  
...  
i 가 4일 때까지 반복하는 거에요.

그 안에 조건문 if 는 다음과 같아요.

arr의 i번째의 값이 arr의 j번째의 값보다 작으면, result[i]의 값을 +1 증가시켜요.  
(result[i]의 값은 [8행]에서 넣어준 1 이 시작 값이에요.)arr[0] 인 77보다 큰 값은 99 하나에요. 즉, 1을 더해줘요.  $1 + 1 = 2$ arr[1] 인 32보다 큰 값은 77, 99, 50 세개에요. 즉, 3을 더해줘요.  $1 + 3 = 4$ arr[2] 인 10보다 큰 값은 77, 32, 99, 50 네개에요. 즉, 4를 더해줘요.  $1 + 4 = 5$ 

arr[3] 인 99보다 큰 값은 없어요. 즉, 그냥 시작값 1이에요.

arr[4] 인 50보다 큰 값은 77, 99 두개에요. 즉, 2를 더해줘요.  $1 + 2 = 3$ 

정답은 24513 이에요.

(한 줄 내리지 않아야 해요. print 는 줄 내림 없이 출력하고, println 은 출력 후 한 줄 내려요)

## ★ key point

이중으로 for문이 동작하는 것뿐, 어렵지 않아요!

코드를 해석하는 게 어렵다면, i=0 을 한번 손으로 직접

계산 해보세요. 규칙이 보일거에요.

## 7. [기출 예제] 21년 1회 실기 (난이도: ★★★)

클래스 내에서 객체 생성 없이 사용할 수 있는 메소드로써 출력 결과를 작성하시오.

```

1 public class Main {
2     public static void main(String[] args) {
3         int i, j;
4         for(j=0, i=0; i<=5; i++) { → j와 i에 0을 할당하고 (시작값), i가 5보다 작거나 같을 때까지 +1 증가하면서 반복해요
5             j+=i; → += 연산자는 j = j + i 로 표현할 수 있어요. 즉, 자기자신과 i를 계속 더해줘요.
6             System.out.print(i); → i가 0, 1, 2, 3, 4, 5까지 순서대로 출력돼요. println이 아닌 print므로 줄내림하지 않아요.
7             if(i==5) { → i가 5일 때는 + 대신 = 를 출력하기 위한 조건문이에요.
8                 System.out.print("=");
9                 System.out.print(j); → i가 5일 때, = 를 출력하고 여태 i를 계속 더했던 j를 출력해요.
10            } else {
11                System.out.print("+");
12            }
13        }
14    }
15 }
```

실제 실행코드

문제가 크게 어렵지는 않았지만, 조심해야 할 포인트가 몇 가지 있어요!

1. println이 아닌 print라 줄내림없이 출력할 것!

2. 더하기 결과값인 15만 출력하는 것이 아닌,  $0+1+\dots+15$  까지 전부 출력할 것!

실수하지 않도록 조심해야해요!

▶ 정답 0+1+2+3+4+5=15

그냥 15만 적으면 틀려요!



## [문제 유형] 단순 while문

## ★ while

특정 조건이 참인 동안 계속해서 코드 블록을 실행하게 해요. 마치 "아직 배가 고프다면 계속 음식을 더 먹어"라고 말하는 것과 비슷해요.  
조건이 거짓이 되는 순간, 반복이 멈춰요.

```
int i = 0; // 'i'라는 변수를 0으로 시작해요.
while (i < 5) { // 'i'가 5보다 작은 동안 계속 반복해요.
    System.out.println("i의 값은 " + i + "입니다."); // 현재 'i'의 값을 출력해요.
    i++; // 그리고 'i'를 1씩 증가시켜요.
}
```

## for 문과의 차이

while 문은 조건이 참인 동안 계속 반복되는데, 주로 반복 횟수가 명확하지 않을 때 사용해요.  
for 문은 반복 횟수가 정해져 있거나, 반복을 위한 변수의 시작과 끝이 명확할 때 주로 사용돼요.

## ★ continue

continue를 만나면, 반복문의 현재 반복을 즉시 종료하고 다음 반복으로 넘어가요.  
마치 "이번 단계는 여기까지 하고, 다음 단계로 바로 넘어가자"라는 뜻이에요.  
반복문의 나머지 부분을 건너뛰고, 반복문의 시작 부분으로 돌아가 새로운 반복을 시작해요.

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        continue; // i가 5일 때, 여기서 나머지 반복문을 건너뛰고 다음 반복으로 넘어가요.
    }
    if (i == 8) {
        break; // i가 8이 되면, for 문을 완전히 끝내요.
    }
}
```

## ★ break

break는 반복문을 완전히 끝내는 역할을 해요.  
마치 "여기까지만 하고, 반복문을 완전히 벗어나자"라고 할 때 쓰는 것과 같아요.  
break를 만나면, 그 즉시 반복문이 종료되고 반복문 바깥의 다음 코드로 넘어가요.

## 8. [기출 예제] 20년 3회 실기 (난이도: ★★★)

▶ 정답 30

다음은 자바 코드이다. 출력 결과를 쓰시오.

실제  
실행코드

```
1 public class Main {
2     public static void main(String[] args) {
3         int i=0; → i가 0으로 선언 및 할당 되었어요.
4         int sum=0; → sum도 0으로 선언 및 할당 되었어요.
5         while (i<10) { → i가 10 미만일 때까지 반복해요. 여기서는 0,1,2,3,4,5,6,7,8,9 까지 반복해요
6             i++; → i의 값을 +1 증가시켜요.
7             if(i%2 ==1) → i를 2로 나눈 나머지가 1일 때, (의미상 홀수)를 뜻해요.
8                 continue; → continue해요. 이 키워드는 다음 반복문으로 넘어간다는 의미에요.
9                 sum += i;
10            }
11            System.out.println(sum);
12        }
13    }
```

i가 10 미만일 때까지 반복해요. 여기서는 0,1,2,3,4,5,6,7,8,9 까지 반복해요  
i를 2로 나눈 나머지가 1일 때, (의미상 홀수)를 뜻해요.  
continue해요. 이 키워드는 다음 반복문으로 넘어간다는 의미에요.  
여기서는 [5행]으로 올라가요. (즉, [9행]을 수행하지 않기 위함이에요)  
[7행] 조건문에 걸리지 않았을 경우, (짝수일 때)  
sum 변수에 i를 더해요.  
즉, 0,1,2,3,4,5,6,7,8,9 중에서 짝수만 더한 결과값을 출력하겠다는 코드에요

정답은  $2+4+6+8+10=30$ 이에요.



## [문제 유형] 상속

## ★ 상속 (Inheritance)

상속은 부모 클래스가 가진 특성(변수)과 행동(메서드)을 자식 클래스가 물려받는 것을 말해요. 마치 부모님의 유전자를 물려받아 닮은 것처럼, 자바에서는 클래스도 다른 클래스의 특성을 물려받을 수 있어요.

```
class Parent {
    void sayHello() {
        System.out.println("Hello, I'm a parent!");
    }
}

class Child extends Parent {
    // Child 클래스는 Parent 클래스의 sayHello 메서드를 상속받았어요.
}
```

Child 클래스는 Parent 클래스로부터 sayHello 메서드를 상속받으니, Child 객체를 만들어 sayHello를 호출하면 "Hello, I'm a parent!"라는 메시지가 출력될 거예요.

## ★ 오버라이딩 (Overriding)

오버라이딩은 상속받은 메서드를 자식 클래스에서 재정의하는 것을 말해요.  
자식 클래스가 상속받은 메서드를 마음에 들게 바꾸고 싶을 때 사용해요.

```
class Child extends Parent {
    void sayHello() {
        System.out.println("Hello, I'm a child!");
    }
}
```

여기서 Child 클래스는 Parent의 sayHello 메서드를 오버라이딩해서 "Hello, I'm a child!"라는 새 메시지를 출력하도록 변경했어요.

## ★ 오버로딩 (Overloading)

오버로딩은 같은 이름의 메서드를 여러 개 가지되, 매개변수의 유형이나 개수를 다르게 하는 것을 말해요.  
이렇게 하면 같은 일을 하는 메서드라도 조금씩 다른 상황에서 사용할 수 있어요.

```
class MathOperations {
    int sum(int a, int b) {
        return a + b;
    }

    // 오버로딩된 sum 메서드, 매개변수의 개수가 다름.
    int sum(int a, int b, int c) {
        return a + b + c;
    }
}
```

MathOperations 클래스에는 sum이라는 메서드가 두 개 있어요. 하나는 두 숫자를 더하고, 다른 하나는 세 숫자를 더해요. 매개변수가 다르기 때문에 두 메서드는 서로 다른 버전으로 존재할 수 있어요.

**의 세계의 개념은 자바에서 정말 중요해요!** 먼저, 이해하시면 좋아요

## 9. [기출 예제] 23년 3회 실기 (난이도: ★★★)

▶ 정답 “BDCDD”

다음은 Java 코드이다. 올바른 출력 결과를 작성하시오.

실행코드	<pre>1 public class main{ 2     public static void main(String[] args) { 3         ① A b = new B(); 4         ② b.paint();           → 자식 클래스 B의 paint() 실행 : [22행] → [14행] 실행 → [15행] “B” 출력 → [16행] → [26행] 실행 → [27행] “D” 출력 5         ④ b.draw(); 6     }                   → 자식 클래스 B의 draw() 실행 7 }                   : [26행] 실행 → [27행] “D” 출력 (끝)  9 class A { 10     public void ③paint(){          → 클래스 A의 메소드와 클래스 B의 메소드 이름이 동일 11         System.out.print("A"); 12         draw(); 13     } 14     ⑤ public void ④draw(){       (= 자식 클래스인 B 메소드 실행) 15         ⑥ System.out.print("B"); 16         ⑦ draw();             → 자식 클래스 draw() = [26행] 17     } 18 }</pre> <p>(오버라이딩 : 동일한 이름의 메소드일 때 자식 클래스 메소드만 실행)</p>
부모 클래스 (A)	<p>★ super.~ : 부모의 메소드를 호출  ★ this.~ : 현재 실행 중인 나의 메소드 호출 / 생략도 가능 → draw() = this.draw()  super와 this에 대한 자세한 설명은 다음장에 있어요.</p>
자식 클래스 (B)	<pre>20 class B extends A {           → 자식 클래스 B는 부모 클래스 A로부터 상속! 21     ③ public void ⑤paint(){   ④ super.draw();           → 부모 클래스 A의 draw() 호출 = [14행] 22         ⑩ System.out.print("C"); 23         ⑪ this.draw();        → 자식 클래스 B의 draw() 호출 = [26행] 24     } 25 } 26 ⑧ ⑨ ⑩ public void ⑥draw(){  ⑪ ⑫ ⑬ System.out.print("D"); 27     ⑭ ⑮ ⑯ }</pre>



### ★ this 키워드

this는 현재 객체, 즉 메서드나 생성자를 호출하는 객체를 가리키는 데 사용돼요.  
같은 이름의 클래스 변수와 메서드 매개변수가 충돌할 때 이를 구별하기 위해 사용되기도 해요.  
클래스 내의 다른 생성자를 호출할 때도 this()를 사용할 수 있어요.

```
class Flower {
    private String name;

    Flower(String name) {
        this.name = name; // 여기서 'this.name'은 클래스 변수를, 'name'은 생성자의 매개변수를 가리켜요.
    }

    void printName() {
        System.out.println(this.name); // 'this'로 클래스 변수 'name'에 접근해요.
    }
}
```

여기서 this.name은 클래스의 name 변수를 참조하고, 단순히 name은 생성자로 전달된 매개변수를 참조해요.

### ★ super 키워드

super는 부모 클래스를 가리키는데 사용되며, 부모 클래스의 변수나 메서드, 생성자에 접근할 때 사용해요.  
자식 클래스에서 오버라이드한 메서드가 부모 클래스의 메서드를 호출하고 싶을 때 super를 사용해요.

```
class Plant {
    String type = "Plant";

    void printType() {
        System.out.println(this.type); // 'this'로 자신의 type 변수에 접근해요.
    }
}

class Flower extends Plant {
    String type = "Flower";

    void printType() {
        super.printType(); // 'super'로 부모 클래스의 printType 메서드를 호출해요.
        System.out.println(this.type); // 'this'로 자신의 type 변수에 접근해요.
    }
}
```

Flower 클래스에서 super.printType()은 Plant 클래스의 printType 메서드를 호출하고, this.type은 Flower 클래스 자신의 type 변수를 참조해요.

▶ 정답 a=10

### 10. [기출 예제] 20년 2회 실기 (난이도: ★★★)

다음은 자바 코드이다. 출력 결과를 쓰시오.

```
1 class A {
2     private int a;
3     ④public A(int a){
4         this.a = a; → a의 멤버 변수를 10을 할당해요.
5     }
6     ⑥public void display(){
7         System.out.println("a=" + a);
8     }
9 }
10
11 class B extends A {
12     ②public B(int a){
13         ③super(a);
14         ⑤super.display(); → super()는 부모 클래스의 생성자를 호출하는 키워드에요.
15     }
16 }
17
18
19 public class Main {
20     public static void main(String[] args){
21         ①B obj = new B(10); → [3행]이 수행돼요.
22     }
23 }
```

A  
부모 클래스

B  
자식 클래스

실제 실행코드

▶ 정답 a=10

설명:  
 1. A 클래스의 생성자에서 this.a = a;은 a의 멤버 변수를 10을 할당해요.  
 2. B 클래스의 생성자에서 super(a);은 부모 클래스의 생성자를 호출하는 키워드에요.  
 3. B 클래스의 생성자에서 super.display();은 부모의 [6행] display()를 호출해요.  
 4. 즉, a에 10을 할당한 뒤, a를 출력하는 코드에요.  
 정답은 a=10이에요.

설명:  
 1. B의 생성자에서 ①B obj = new B(10);은 B의 생성자에 10을 넣어 B 객체를 생성합니다.  
 2. [12행]이 수행돼요.



▶ 정답 11

## 11. [기출 예제] 21년 2회 실기 (난이도: ★★★)

다음은 JAVA 관련 문제이다. 알맞는 출력값을 작성하시오.

**실제 실행코드**

```

1  public class Ovr1 { Ovr1 부모 클래스
2      public static void main(String[] args) {
3          ①Ovr1 a1 = new Ovr1(); Ovr1 타입의 객체 a1을 생성하고 있어요.
4          ②Ovr2 a2 = new Ovr2(); Ovr2 타입의 객체 a2를 생성하고 있어요. Ovr2는 Ovr1을 상속받았기 때문에 Ovr1의 메서드를 가지고 있어요.
5          ③System.out.println(a1.sun(3,2) + a2.sun(3,2)); → Ovr1의 sun은 3+2 = 5이고,
6                                         Ovr2의 sun은 3-2+5 = 6이기 때문에 정답은 11이 출력돼요.
7      }
8
9      ④int sun(int x, int y) {
10         return x + y;
11     }
12
13     class Ovr2 extends Ovr1 {
14
15         ⑤int sun(int x, int y) { → sun 메서드를 재정의(오버라이드)하고 있어요.
16             return x - y + super.sun(x,y);
17         }
18     }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

**Ovr2 자식 클래스**

④int sun(int x, int y) { → Ovr1 클래스에 sun이라는 메서드를 정의하고 있어요.  
두 정수를 받아서 그 합을 반환합니다.

⑤int sun(int x, int y) { → sun 메서드를 재정의(오버라이드)하고 있어요.  
이 메서드는 인자로 받은 두 정수의 차에 부모 클래스의 sun 메서드의 결과를 더해 반환합니다.  
super.sun(x,y)는 부모 클래스인 Ovr1의 sun 메서드를 호출하는 방법이에요.

## ★extends

class 자식 extends 부모 의 형태로 사용해요.

부모를 상속 받은 자식은 부모의 변수 및 메소드를 사용할 수 있고,  
오버라이드를 통해 내 코드로 수정해서 사용할 수 있어요.

여기서는 [15행]이 부모의 sun을 오버라이드해서 사용하고 있어요.

이럴 경우, 자식 코드에서 부모의 메소드를 호출 하고 싶을 때는

super.메소드 명으로 사용할 수 있어요.

## 12. [기출 예제] 23년 1회 실기 (난이도: ★★★)

▶ 정답 5000

아래 자바 코드에서 출력되는 값을 작성 하시오.

**부모 클래스**

```

1  class Parent {
2      int x = 1000;
3
4      Parent() {
5          this(3000); ← 기본 생성자 [4행]은 [8행] 생성자를 호출해서, X에 3000을 할당해요.
6      }
7
8      Parent(int x) {
9          this.x = x;
10     }
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

**자식 클래스**

```

1  class Child extends Parent {
2      int x = 4000;
3
4      ②Child() {
5          ③this(5000); ← 기본 생성자 [16행]은 [20행] 생성자를 호출해서, X에 5000을 할당해요.
6      }
7
8      ④Child(int x) {
9          ⑤this.x = x;
10     }
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

```

**실제 실행코드**

⑦int getX() { → 만약, getX()가 (Child에 없고) Parent에만 구현되어 있었다면, 정답은 3000이 되었을거에요.  
코드를 보면, getX()는 자기 클래스의 x를 반환하니까요.

①Child obj = new Child(); → Child를 기본 생성자로 생성했어요. [16행] → [17행] → [20행] → [21행] 의 순서에 의해 Child의 x는 5000이 되었어요.  
⑥System.out.println(obj.getX()); → Child의 getX() 메소드는 [24행] Child의 x를 반환하기 때문에 정답은 5000이 돼요.



▶ 정답 2

## 13. [기출 예제] 23년 3회 실기 (난이도: ★★★)

다음은 Java 코드이다. 올바른 출력 결과를 작성하시오.

부모 클래스 (Parent)

```

1 class Parent {
2     int compute(int num) { → 이 문제에서는 호출되지 않아요.
3         if(num <= 1) [19행]에서 Parent obj = new Parent(); 로 할당되었다면 해당 메소드가 호출되었을 것.
4             return num;
5         return compute(num-1) + compute(num-2);
6     }
7 }

```

자식 클래스 (Child)

```

9 class Child extends Parent { → 실제 호출되는 메소드.
10    int compute(int num) { → 자기 자신을 계속 호출하고 있어요.
11        if(num <= 1) [20행]에서 7을 넣어줬으니, compute(6) + compute(4) 가 호출돼요.
12            return num; → * compute(n) 에서 n이 1이하일 때, n을 반환해요.
13            return compute(num-1) + compute(num-3); → 즉, compute(6) 의 값과 compute(4) 의 값을 구해서 더해주면 정답이에요.
14    }
15 }
16

```

실제 실행코드

```

17 public class main {
18     public static void main(String args[]) {
19         Parent obj = new Child(); → 1 + 0 = 1
20         System.out.print(obj.compute(7)); → 1 + 0 = 1
21     }
22 }
23

```

자기 자신을 계속 호출하고 있어요.  
[20행]에서 7을 넣어줬으니, **compute(6) + compute(4)** 가 호출돼요.  
\* **compute(n)** 에서 **n이 1이하일 때, n을 반환해요.**  
즉, **compute(6)** 의 값과 **compute(4)** 의 값을 구해서 더해주면 정답이에요.  
**compute(6)**은 **compute(5) + compute(3)**의 합  
**compute(5)**는 **compute(4) + compute(2)**의 합  
**compute(4)**는 **compute(3) + compute(1)**의 합  
**compute(3)**은 **compute(2) + compute(0)**의 합  
**compute(2)**는 **compute(1) + compute(-1)**의 합  
**compute(1) + compute(-1)**부터 거꾸로 계산하면서 올라가면서 풀어야요.  
즉, **compute(6)**은 1, **compute(4)**는 1이므로, **정답은 2**

</



▶ 정답 new

## 15. [기출 예제] 20년 2회 실기 (난이도: ★★★)

다음은 자바 코드이다. 다음 밑줄에 들어갈 키워드를 쓰시오.

```

Parent 부모 클래스
1 class Parent {
2     public void show() {
3         system.out.println("Parent");
4     }
5 }

Child 자식 클래스
6
7 class Child extends Parent {
8     ③public void show() {
9         ④system.out.println("Child");
10    }
11 }

실제 실행코드
12
13 public class Main {
14     public static void main(String[] args) {
15         ①Parent pa = _____ Child(); →
16         ②pa.show(); → [8행]이 호출돼요.
17     }
18 }

```

Parent 부모 클래스의 show()가 Child 자식 클래스 show로 오버라이드 되었어요.  
즉, [16행] 호출 시, [8행]의 show가 호출돼요.  
물론, 이 문제와는 상관없어요!

## ★ key point

이제는 아마, 나올 일 없는

가장 기본적인 자바의 객체 생성에 대한 문법 문제이에요.

자바에서 생성자를 만드는 기본 문법은 A a = new A();입니다  
A는 클래스명이에요.  
여기서는 Parent pa = new Child(); 이 부분이에요.  
그래서 정답은 new가 돼요

가장 기본적인 문법이에요.

# 꿈꾸는라이언 정처기 요약노트 (실기 – 프로그래밍 언어편)

## 누적 1,500건 이상의 만족도 100% 생생한 후기

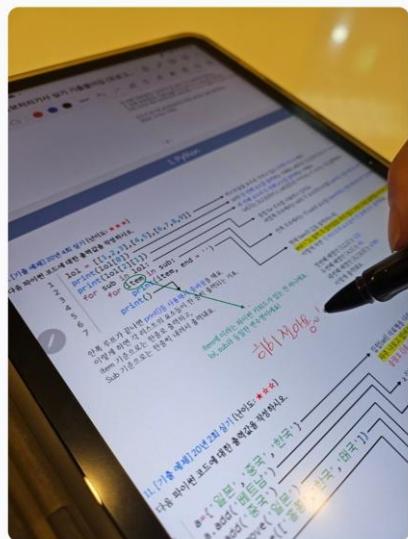


★★★★★ 5

jooh\*\*\*\*\* · 24.04.11. · 신고

**BEST** 친구가 이거랑 이론편으로 공부하는 거보고 저도 따라 샀어요!!! 언어별, 유형별로 나누어져 있어 깔끔하고 보기좋아요. 비전공자로 맨땅에 헤딩하는 기분이었는데 코드 라인 한줄한줄 엄청 자세하고 친절하게 설명 되어있어서 이게 하나로 그냥 프로그래밍 언어 대비하면 될 거 같아요!!!  
톡톡으로 문의드렸더니 바로바로 답장해주시고 주변에 계속 추천하고 다니고 있어요ㅋㅋ 진짜 대만족이에요 bb  
전교1등 노트 훔쳐보는 느낌이랄까

접기 ^



★★★★★ 5

cacy\*\*\* · 24.04.07. · 신고

책으로 프로그래밍 부분을 공부하는데, 문제에 .. 한 풀이가 자세하지 않아서 공부하기가 어렵더라고요..! 그러던 중 꿈꾸는 라이언님의 실기 요약본이 생각나 바로 구매했습니다! 주제별로 다양한 문제에 대한 해설이 정리되어 있어서 보기 좋은 것 같아요! 좋은 자료 만들어주셔서 감사합니다! 😊



★★★★★ 5

khso\*\*\* · 24.04.10. · 신고

올해 정보처리기사 필기 준비하면서 블로그에서 나눔 해주신 무료버전이 굉장히 많은 도움이 됐었습니다!  
저를 일주일만에 합격시켜준, 💫 이번 실기도 오늘부터 열심히 하려구 무료버전 내용 확인 후 바로 이론과 프로그래밍 언어 모두 결제 하겠습니다!  
역시 내용도 잘 정리 되어있고, 어떻게 공부해야할지 막막했는데 어느정도 감이 잡히는 것 같아요!!  
프로그래밍언어 포기하지 않고 열심히 공부하겠습니다!



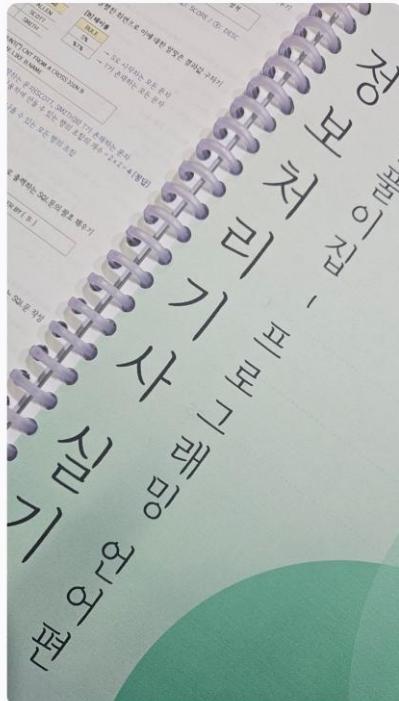
★★★★★ 5

rlaw\*\*\*\*\* · 24.04.11. · 신고

컴활, 한국사까지 모두 여기서 구매해서 항상 만족하는 급수를 받았습니다!! 정처기 필기도 구매했었는데 다행히 안정권으로 합격했었습니다. 이번에 실기 요약본을 찾던 도중 찾게 되어 너무 반가웠습니다!! 이번에도 이론편과 프로그래밍 기출풀이집까지 구매해서 열심히 따보겠습니다!! 진짜 만들어주셔서 항상 감사합니다! 👍



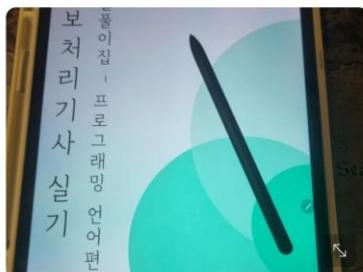
접기 ^



★★★★★ 5

hush\*\*\*\*\* · 24.04.08. · 신고

실기의 경우 펜을 쓰면서 풀어보는게 가장 중요하다고 생각했는데 책이 아닌 태블릿을 이용해 풀어보고 지우고를 반복해볼 수 있는게 너무 좋습니다



★★★★★ 5

jijo\*\*\*\*\* · 24.04.12. · 신고

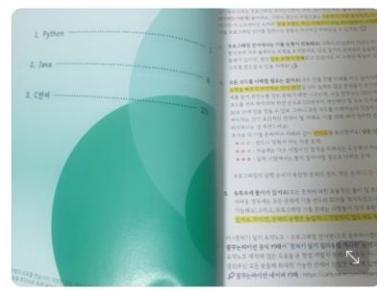
세상에.....혼자 독학하기 넘 좋아요 문제 해설 너무 완벽하고 친절하고꼼꼼하고..기출인강이 필요 없을 정도...유료강의랑 교재 모두 구입해서 들었었는데 유료강의의 책, 요약본, 문제 자료들보다 더 좋아요!!!



★★★★★ 5

dudw\*\*\*\*\* · 24.04.15. · 신고

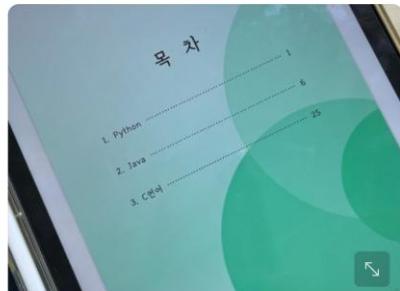
정보처리기사 실기 프로그래밍 공부를 어떻게 해야 할지 고민하고 있었는데, 해당 요약집으로 공부하니 이해도 빠르고 공부하기도 편해져서 좋아요!! 잘 활용해서 정처기 실기 빨리 합격하도록 하겠습니다ㅎㅎ



★★★★★ 5

rlaw\*\*\*\*\* · 24.04.11. · 신고

프로그래밍 언어편 엄청 기다렸는데! 드디어 나왔네요 😊 여기가 요점정리 원탑이에요👍👍 열심히 공부해서 정처기 한 번에 합격하고 싶어요~



★★★★★ 5

dlwn\*\*\*\*\* · 24.04.15. · 신고

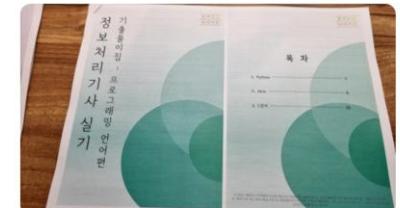
인터넷 서치하면서 요약본 찾고 있었는데 제 맘에 들게 되었는데 없더라고요ㅠㅠ 무료 요약본은 무료인데 이유가 있고 다른 것은 책 사는것만큼 돈이 들어서 고민이었는데 딱 좋은 요약집 찾아서 너무 맘에 들니다!! 내용이 너어무 많아서 보기 부담스럽지도 않고 최근 3개년에 출제된 개념은 다 정리되어 있어서 너무 만족합니다!!



★★★★★ 5

gon\*\*\*\* · 24.04.14. · 신고

프로그래밍 언어 중심으로 공부할 수 있는 최고의 자료에요!! 디자인도 너무 이쁘고 열심히해서 꼭 격할게요!!♥



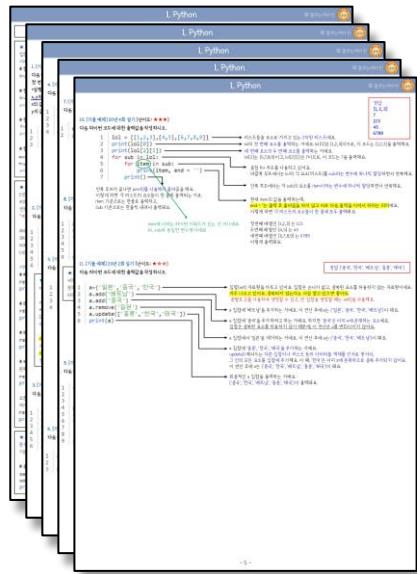
# 정처기 실기 (프로그래밍 언어편) 전범위 자료가

궁금하시다면?

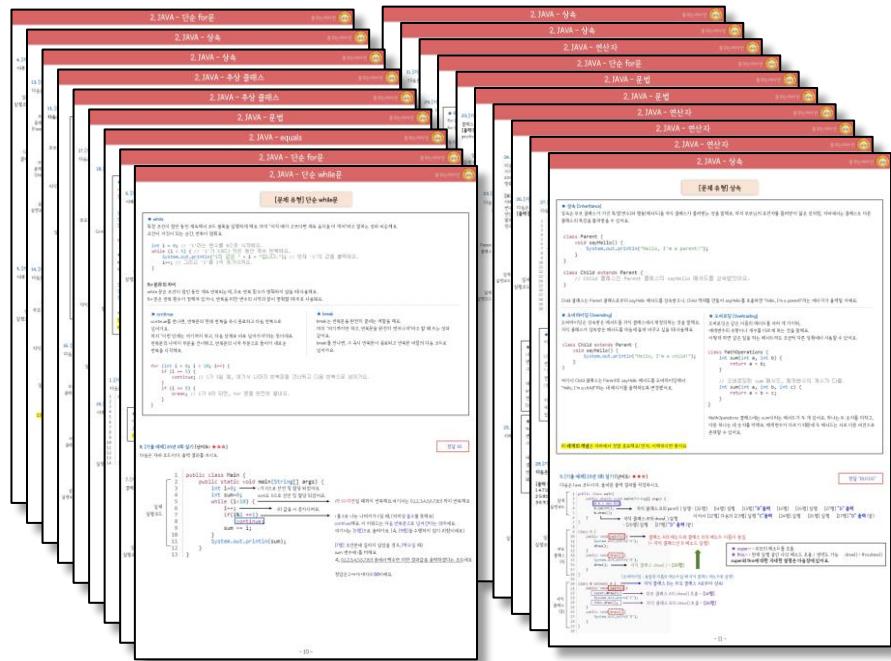
여기 를 클릭해주세요!

(20년도 ~ 최신 기출문제 풀이 수록)

## 1. Python



## 2. Java



## 3. C언어

