

오픈소스 기초프로젝트 중간고사

2024. 5. 1(수).

< 안내사항 >

1. 문항은 총 **80문항**입니다.
2. 한 문제당 **점수는 0.25점씩, 80문항을 맞추면 총 20점**입니다.
부분점수는 없습니다.
3. 빈칸을 채우는 문제 중 [] 라고 표현된 부분 중 괄호는 무시 바랍니다.
4. 평가방법에 따른 점수 반영비율은 아래 표와 같습니다.

중간고사	기말고사	출석	퀴즈	과제	합계
20%(20점)	50%(50점)	10%(10점)	10%(10점)	10%(10점)	100%(100점)

1. 다음 괄호안에 들어갈 용어를 보기에서 고르시오.

보기 : 프로그래밍 언어, 소프트웨어, 앱, 프로그래머, 사람의 언어

컴퓨터가 이해하는 말을 (1)라고 한다. (1)는 컴퓨터에서 작동하는 (2)를 만드는 도구 같은 것으로, 요즘에는 스마트폰에서 작동하는 앱도 (1)로 만든다. 그리고 (1)를 사용해 (2)나 앱을 만드는 사람을 (3)라고 한다.

2. 프로그래밍 언어가 아닌 것을 모두 고르시오.

- ① Java ② Excel ③ C++ ④ JavaScript ⑤ Chrome

3. 다음 설명 중 옳은 것을 모두 고르시오.

- ① 초보자가 배우기 쉬운 언어임
② 개발자 빌게이츠가 C언어로 제작함
③ 무료로 사용할 수 있지만, 소스코드는 공개되어 있지 않음
④ 라즈베리파이 기반의 사물인터넷과 연동이 잘 됨
⑤ 외부 라이브러리가 좀 부족한 것이 단점임
⑥ 인스타그램도 파이썬 웹 프레임워크로 제작되었음

4. 다음 중 "안녕, 파이썬"이 출력되는 코드를 고르시오.

- ① print("안녕, 파이썬")
② printf("안녕, 파이썬")
③ write("안녕, 파이썬")
④ put("안녕, 파이썬")

5. 파이썬을 실행하는 통합 환경을 부르는 용어를 고르시오.

- ① Prompt
② IDLE Shell
③ Interpreter
④ Compiler

6. 다음 코드의 출력 결과를 작성하시오.

- ① 100 + 200 : (1)
② 100 - 200 : (2)
③ 100 x 200 : (3)
④ 100 / 200 : (4)

7. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :
Hello world!

8. 다음 코드의 출력 결과를 작성하시오.

- ① `print("11+22-33")` : (1)
- ② `print(안녕? 파이썬)` : (2)

9. 다음 코드의 실행 결과가 다른 것을 고르시오.

- ① `print("100")`
- ② `print(100)`
- ③ `print(50+50)`
- ④ `print("50+50")`

10. 다음 코드의 실행 결과를 고르시오.

```
print('%d / %d = %d' % ( 5, 10, 5/10 ))
```

- ① `5 / 10 = 0.5`
- ② `5 / 10 = 0.50000`
- ③ `5 / 10 = 0`
- ④ `5 / 10 = 0.00000`

11. 다음 코드의 출력 결과를 작성하시오.

- ① `print("%04d" % 876)` : (1)
- ② `print("%5s" % "Software")` : (2)
- ③ `print("%.1f" % 123.45)` : (3)

12. 다음 코드의 실행 결과를 고르시오.

```
print("{2:d}, {0:d}, {1:d}".format(111, 222, 333))
```

- ① 111, 222, 333
- ② 333, 222, 111
- ③ 333, 111, 222
- ④ 222, 333, 111

13. 다음 중 보기에서 설명하는 이스케이프 문자를 작성하시오

보기 : \n, \t, \b, \\, \', \"

- ① 다음 탭으로 이동 : (1)
- ② 뒤로 한 칸 이동 : (2)
- ③ \를 출력 : (3)

14. 파이썬에서 제공되는 데이터 형식 중 거리가 먼 것을 고르시오.

- ① 불형 : bool
- ② 정수 : int
- ③ 실수 : float
- ④ 문자 : character
- ⑤ 문자열 : str

15. 다음 코드의 실행 결과 중 오류가 발생하는 것을 모두 고르시오.

- ① num1 = 100
- ② 100 = num1
- ③ num1 = num2 = 100
- ④ num1 = 100 = num2 = 100
- ⑤ num1 = num2 = num3 = 100

16. 각 진수를 10진수로 변환하시오.

- ① 2진수 0011 : (1)
- ② 2진수 1010 : (2)
- ③ 16진수 11 : (3)
- ④ 8진수 17 : (4)

17. 각 진수를 10진수로 변환하시오.

- ① 1011(2진수) : (1)
- ② 1A (16진수) : (2)

18. 다음 괄호에 들어갈 함수 이름을 쓰시오.

【 1 】 함수는 숫자를 16진수로, 【 2 】 함수는 숫자를 8진수로, 【 3 】 함수는 숫자를 2진수로 출력합니다. 그리고, 16진수로 숫자를 선언하고자 할 때는 숫자 앞에 【 4 】를, 8진수로 선언하고자 할 때는 숫자 앞에 【 5 】를, 2진수로 숫자를 선언하고자 할 때는 숫자 앞에 【 6 】를 붙여야 합니다.

19. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

10진수 : 12345678

16진수 : 0xbc614e

8진수 : 0o57060516

2진수 : 0b101111000110000101001110

num = 12345678

print("10진수 : ", 【 1 】)

print("16진수 : ", 【 2 】)

print(" 8진수 : ", 【 3 】)

print(" 2진수 : ", 【 4 】)

20. 다음 각 산술연산자의 설명을 보기에서 고르시오.

보기 : =, +, -, *, ^, ~, /, //, %, **, ***, &, #, \$, @, ?, ==

① 대입 연산자 : 【 1 】

② 나누기(몫) : 【 2 】

③ 나머지 값 : 【 3 】

④ 제곱 : 【 4 】

21. 다음 코드의 출력 결과를 작성하시오.

n1, n2, n3 = 3, 2, 1

① $n1 + n2 * n3$: 【 1 】

② $n1 * n2 + n3$: 【 2 】

③ $n1 - n2 * n3$: 【 3 】

④ $n1 * n2 - n3$: 【 4 】

22. 다음 괄호에 들어갈 함수 이름을 쓰시오.

문자열을 정수로 변환하는 함수는 (1), 실수로 변환하는 함수는 (2) 입니다.
그리고 숫자를 문자열로 변경하는 함수는 (3) 입니다.

23. 다음 코드의 출력 결과를 작성하시오.

```
num = 100;  
num += 1;  
num -= 1;  
num *= 1;  
num /= 1;  
num = int(num);  
print(num)
```

24. 다음 각 관계 연산자의 결과를 True 또는 False로 고르시오.

num1, num2 = -100, 100

- ① num1 == num2 : (1)
- ② num1 >= num2 : (2)
- ③ num1 <= num2 : (3)
- ④ num1 != num2 : (4)

25. 다음 각 논리 연산자의 결과를 True 또는 False로 고르시오.

num1, num2 = -100, 100

- ① (num1 == num2) and (num1 != num2)
- ② (num1 == num2) or (num1 != num2)
- ③ (num1 >= num2) and (num1 <= num2)
- ④ (num1 >= num2) or (num1 <= num2)

26. 다음 중 num이 100인 경우 실행되는 코드를 고르시오.

- ① if num = 100 :
- ② if num equal 100 :
- ③ if num == 100 :
- ④ if num != 100 :

27. 다음 코드의 실행 결과를 고르시오.

```
num = 0
if num > 0 :
    print("케이스1", end = ' ')
else :
    print("케이스2", end = ' ')
print("케이스3", end = ' ')
```

- ① 케이스1 케이스2
- ② 케이스2 케이스3
- ③ 케이스1 케이스3

28. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

정수를 입력하세요 : 8
5의 배수가 아닙니다.

```
num = int(input("정수를 입력하세요 : "))
if (  ) :
    print("5의 배수가 아닙니다.")
else :
    print("5의 배수입니다.")
```

29. 다음 코드의 중첩 if 문을 if~elif 문으로 변경하시오.

```
score = int(input("점수를 입력하세요 : "))

if score >= 90 :
    print("장학생", end="")
else :
    if score >= 60 :
        print("합격", end="")
    else :
        print("불합격", end="")

print("입니다.")
```

```
score = int(input("점수를 입력하세요 : "))
```

```
print("입니다.")
```

30. 다음 중 삼항 연산자로 변경된 문법이 옳은 코드를 고르시오. (num = 5 로 가정함)

```
num = 5
if num % 2 == 0 :
    res = '짝수'
else :
    res = '홀수'
print(res)
```

- ① res = print('홀수') if num % 2 == 0 else print('짝수')
- ② res = if num % 2 == 0 print('짝수') else print('홀수')
- ③ res = print('짝수') if num % 2 == 0 else print('홀수')
- ④ res = if num % 2 == 0 print('홀수') else print('짝수')

31. for 문의 기본 형식을 참고해 빈칸을 채우시오.

보기 : 시작값, 증가값, 끝값, 끝값+1

```
for 변수 in range( [ 1 ], [ 2 ], [ 3 ] ) :
    이 부분을 반복
```

32. 다음 코드의 출력 결과를 작성하시오.

```
for i in range(0, 101, 1) :
    i += 1
print(i)
```

33. 다음 중 출력되지 않는 값을 고르시오.

```
for i in range(5, -1, -1) :
    print("%d" % i)
```

- ① 5
- ② 1
- ③ 0
- ④ -1

34. 다음 실행 결과가 나오도록 고르시오.

실행결과 :

15

```
hap = 0
for i in range( [ ] ) :
    hap += i
print(hap)
```

- ① 0, 10, 5
- ② 0, 11, 5
- ③ 1, 10, 5
- ④ 1, 11, 5

35. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

```
15 × 9 = 135
15 × 8 = 120
15 × 7 = 105
15 × 6 = 90
15 × 5 = 75
15 × 4 = 60
15 × 3 = 45
15 × 2 = 30
15 × 1 = 15
```

```
dan = 15
for i in range( [ 1 ] ) :
    print("%d × %d = %3d" % ( [ 2 ] ) )
```

36. 다음 코드는 "Software"를 몇 번 출력하는지 결과를 쓰시오.

```
for i in range(4) :
    for j in range(3) :
        for k in range(2) :
            print("Software")
```

37. while()문의 문법에 맞는 것을 고르시오.

- ① while(초기값, 끝값+1, 증가값)
- ② while(반복횟수)
- ③ while(조건식)
- ④ while(초기값)

38. 다음 3개 변수(n1, n2, n3)의 값을 리스트(nn)로 변경하는 코드를 고르시오.

```
n1, n2, n3 = 100, 200, 300
```

- ① nn = [100 200 300]
- ② nn = [100, 200, 300]
- ③ nn = [100 & 200 & 300]
- ④ nn = [100 # 200 # 300]

39. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :
668 ※ 10개의 랜덤한 값의 합계, 실행 때 마다 결과가 바뀜

```
import random
nn = []
for _ in range(10) :
    num = random.randrange(1, 100)
    ( 1 )
hap = 0
for i in range(10) :
    num = ( 2 )
    hap += num
print(hap)
```

40. 다음 리스트 중 문법상 옳은 것을 모두 고르시오.

- ① nn = []
- ② nn = [100, 200, 300]
- ③ nn = ["안녕", 1234, '하세요']
- ④ nn = [3.14, 'ㅋ' , 500]

41. 다음 실행 결과가 나오도록 고르시오.

실행 결과 :

[1, 2, 3, 4]

[4, 3, 2, 1]

```
ary1 = [ 1, 2, 3, 4 ]
```

```
ary2 = []
```

```
for i in range( ) :
```

```
    ary2.append(ary1[i])
```

```
print(ary1)
```

```
print(ary2)
```

- ① 3, 1, -1
- ② 3, -1, -1
- ③ 4, -1, -1
- ④ 3, 0, -1

42. 다음 코드의 출력 결과를 작성하시오.

```
nn = [ 100, 200, 300, 400, 500 ]
```

- ① nn[4] : (1)
- ② nn[-1] : (2)
- ③ nn[-2] : (3)
- ④ nn[1:4] : (4)
- ⑤ nn[0:1] : (5)

43. 다음 코드의 실행 후 nn 리스트에 저장된 결과를 작성하시오.

```
nn = [ 100, 200, 300, 400, 500 ]
```

- ① nn[1] = 777 : (1)
- ② nn[1] = [444, 555] : (2)
- ③ nn[1:4] = [444, 555] : (3)
- ④ nn[2:] = [] : (4)

44. 다음 괄호안에 들어갈 용어를 보기에서 고르시오.

보기 : append(), pop(), sort(), reverse(), index(), insert(), remove(), extend(), count(), del(), len(), copy()

- ① (1) : 리스트의 전체 개수를 반환
- ② (2) : 두 리스트를 연결
- ③ (3) : 리스트 맨 뒤의 항목을 추출
- ④ (4) : 지정한 값을 찾아서 위치를 반환
- ⑤ (5) : 리스트에서 특정 값의 개수를 반환

45. 2행 3열의 2차원 리스트인 lst 를 옳게 표현한 것을 고르시오.

- ① lst = [1, 2, 3] , [4, 5, 6]
- ② lst = [[1, 2, 3] , [4, 5, 6]]
- ③ lst = [[1, 2] , [3, 4] , [5, 6]]
- ④ lst = [1, 2, 3, 4, 5, 6]

46. 다음 코드의 출력 결과를 작성하시오.

```
myList = [ [ 1, 2, 3, 4 ] , [ 5, 6, 7, 8 ] , [ 9, 10, 11, 12 ] ]
```

- ① myList [0][1]
- ② myList [1][3]
- ③ myList [2]

47. 자료구조 초기화를 위해 괄호안에 들어갈 용어를 보기에서 고르시오.

보기 : 튜플, 리스트, 딕셔너리, 일반변수

- ① (1) = []
- ② (2) = { }
- ③ (3) = ()
- ④ (4) = None

48. 다음 코드의 출력 결과를 작성하시오.

```
myData = [ [ n * m for n in range(1, 3) ] for m in range (2, 4) ]  
print(myData)
```

49. 다음 코드의 실행 결과가 올바르지 않은 것을 모두 고르시오.

```
ss = 'Software'
```

- ① ss → 'Software'
- ② ss[0:2] → 'So'
- ③ ss[3:-1] → 'tware'
- ④ ss[:4] → 'Soft'
- ⑤ ss[0:-1] → 'Software'

50. 다음 코드의 출력 결과를 작성하시오.

```
ss='Python'
```

- ① print(ss[0]) : (1)
- ② print(ss[1:2]) : (2)
- ③ print(ss[-3:-1]) : (3)
- ④ print(ss[3:]) : (4)

51. 다음 코드의 출력 결과를 작성하시오. (오류가 있다면 오류라고 표기할 것)

```
str1 = 'Soft'  
str2 = 'ware'
```

- ① str1 + str2 : (1)
- ② 2 * str1 : (2)
- ③ str1 * str2 : (3)
- ④ str1 / str2 : (4)
- ⑤ str1 - str2 : (5)

52. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

원본 내용 : Python_Software

변경 내용 : e#a#t#o#_#o#t#P

```
inStr = 'Python_Software'
outStr = ''

for i in range(0, len(inStr)) :
    if i % 2 == 0 :
        ( )
    else :
        outStr += '#'

print("원본 내용 :", inStr)
print("변경 내용 :", outStr)
```

53. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 : P\$ y\$ t\$ h\$ o\$ n\$

```
ss = 'Python'

( 1 ) i in range(0, ( 2 ) :
    print(( 3 ) + '$', end = ' ')
```

54. 다음 괄호안에 들어갈 용어를 보기에서 고르시오.

보기 : upper(), lower(), swapcase(), title()

- ① (1) : 각 단어의 첫 글자를 대문자로 변경
- ② (2) : 모두 영문 대문자로 변경
- ③ (3) : 영문 대소문자를 상호 변경
- ④ (4) : 모두 영문 소문자로 변경

55. 다음 코드의 출력 결과를 작성하시오.

```
str1 = "코딩_중에서_파이썬_코딩이_가장_즐거움_코딩" # 총 24글자
```

- ① str1.count('코딩')
- ② str1.rfind('코딩')
- ③ str1.startswith('코딩')
- ④ str1.find('파이썬')

56. 다음 실행 결과가 나오도록 고르시오.

실행 결과 : Python Python

```
ss = "Python 파이썬"
```

- ① ss.change('파이썬', 'Python')
- ② ss.replace('파이썬', 'Python')
- ③ ss.change('Python', '파이썬')
- ④ ss.replace('Python', '파이썬')

57. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

```
Reverse of input string: n
Reverse of input string: no
Reverse of input string: noh
Reverse of input string: noht
Reverse of input string: nohty
Reverse of input string: nohtyP
```

```
inStr, outStr = "Python", ""
strLen = len(inStr)
```

```
for i in range(0, strLen) :
    outStr += ( [ ] )
    print("Reverse of input string:", outStr)
```

58. 다음 실행 결과가 나오도록 고르시오.

실행 결과 : S#o#f#t#w#a#r#e
ss = '#' [1]

- ① 'Software'.join(ss)
- ② ss.join('Software')
- ③ ss.split('Software')
- ④ 'Software'.split(ss)

59. 다음 설명 중 틀린 것을 모두 고르시오.

- ① 함수는 뒤에 함수명() 처럼 뒤에 괄호 또는 함수명[] 처럼 뒤에 대괄호가 붙음
- ② print("값") 처럼 괄호 안에 데이터를 전달할 수 있음
- ③ 파이썬은 자체 함수는 제공하지 않지만, 필요시 사용자가 직접 만들어서 사용할 수 있음
- ④ 함수의 사용을 위해서 내부 작동을 꼭 알아야할 필요는 없음

60. 다음 빈 칸에 들어갈 내용을 쓰시오.

함수는 [1]를 입력 받은 후에, 그 [1]를 가공해서 [2]을 돌려줍니다. 자판기의 예를 들면 동전 및 버튼 입력이 [1]이고, 그 결과 나오는 커피가 [2]입니다.

61. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 : 600
def plus([1]) : result = 0 result = v1 + v2 + v3 [2] hap = plus(100, 200, 300) print(hap)

62. 다음 코드의 출력 결과를 작성하시오.

```
def f1() :  
    print(var)  
  
def f2() :  
    var = 10  
    print(var)  
  
var = 100
```

- ① f1()
- ② f2()

63. pass 예약어의 의미를 가장 잘 설명한 것을 고르시오.

- ① 동일한 내용을 반복한다는 의미다
- ② 이후의 코드를 모두 건너뛴다는 의미다
- ③ 그냥 아무 것도 하지 않고 비워 놓을 때 사용한다
- ④ if 문에서만 사용된다

64. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행 결과 :

매개변수 없이 호출 : 6
매개변수가 1개로 호출 : 6
매개변수가 2개로 호출 : 6
매개변수가 3개로 호출 : 6

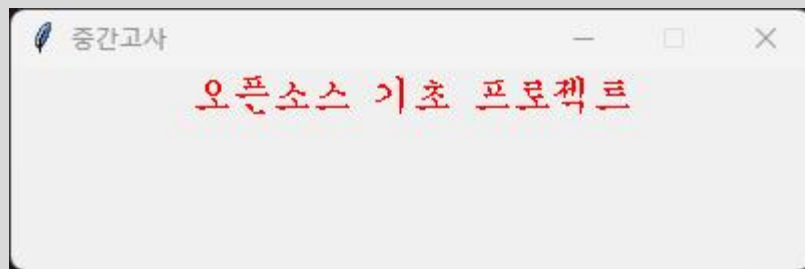
```
def myFunc( [ ] ) :  
    ret = p1 + p2 + p3  
    return ret  
  
print("매개변수 없이 호출 :", myFunc())  
print("매개변수가 1개로 호출 :", myFunc(1))  
print("매개변수가 2개로 호출 :", myFunc(1, 2))  
print("매개변수가 3개로 호출 :", myFunc(1, 2, 3))
```

65. 다음 설명 중 거리가 먼 것을 모두 고르시오.

- ① 모듈은 함수들의 집합으로 볼 수 있다
- ② 파이썬은 자체 모듈을 제공하지 않지만, 필요하면 직접 모듈을 만들어서 사용할 수 있다
- ③ mod.py 파일을 임포트 하려면 import mod 문을 사용하면 된다
- ④ 모듈 안의 함수를 호출할 때는 '모듈명[함수명]' 형식을 사용한다
- ⑤ 패키지는 주로 'from 패키지명.모듈명 import 함수명' 사용한다

66. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :



```
from tkinter import *

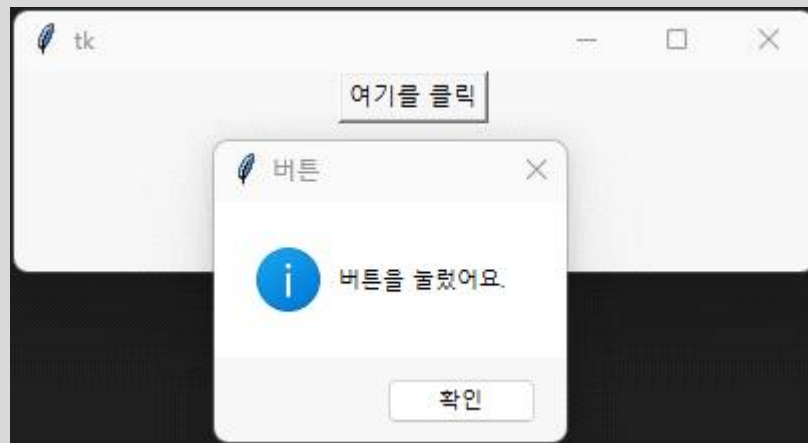
window = Tk()
window.【 1 】 ("중간고사")
window.【 2 】 ("400x100")
window.【 3 】 (width = FALSE, height = FALSE)

label1 = Label(window, text = "오픈소스 기초 프로젝트", font = ("궁서체", 15), fg = "red")
label1.【 4 】();

window.【 5 】()
```

67. 다음 실행 결과가 나오도록 고르시오.

실행결과 :



```
from tkinter import *
from tkinter import messagebox

def click_button() :
    messagebox.showinfo("버튼", "버튼을 눌렀어요.")

window = Tk()
button1 = Button(window, text='여기를 클릭', ( , ))
button1.pack()

window.mainloop()
```

- ① `command = click_button()`
- ② `execute= click_button()`
- ③ `command = click_button`
- ④ `execute = click_button()`

68. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :



```
from tkinter import *
window = Tk()
window.geometry("400x100")

def rdo_change() :
    if [ 1 ] == 1 :
        label1.configure(text = "벤츠")
    else :
        label1.configure(text = "포르쉐")

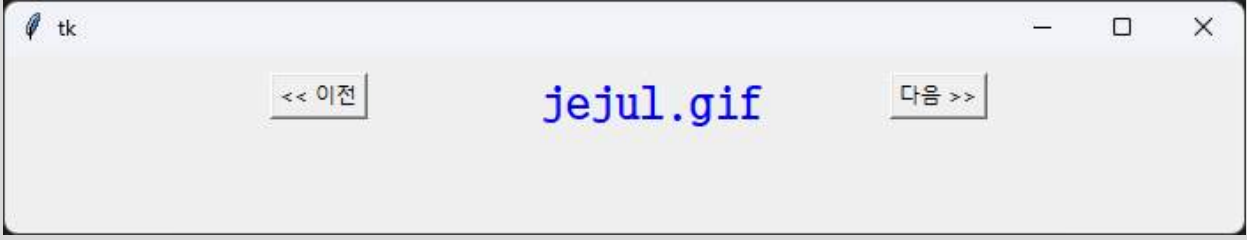
var = [ 2 ]
rdo1 = Radiobutton(window, text = "벤츠", variable = [ 3 ], value = 1,
                    command = rdo_change)
rdo2 = Radiobutton(window, text = "포르쉐", variable = [ 3 ], value = 2,
                    command = rdo_change)
label1 = Label(window, text="선택한 차량", fg="red")

rdo1.pack()
rdo2.pack()
label1.pack()

window.mainloop()
```

69. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :



```
from tkinter import *
from time import *
fnameList = ["jeju1.gif", "jeju2.gif", "jeju3.gif", "jeju4.gif", "jeju5.gif", "jeju6.gif",
             "jeju7.gif", "jeju8.gif", "jeju9.gif"]
num = 0

def clickNext() :
    global num
    [ 1 ]

    pLabel.configure(text = fnameList[num])
def clickPrev() :
    global num
    [ 2 ]

    pLabel.configure(text = fnameList[num])

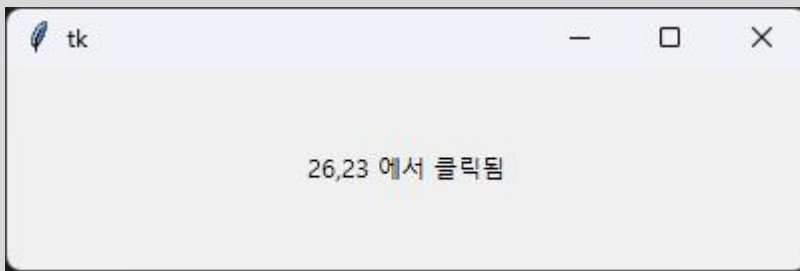
window = Tk()
window.geometry("700x100")
btnPrev = Button(window, text = "<< 이전", command = clickPrev)
btnNext = Button(window, text = "다음 >>", command = clickNext)
pLabel = Label(window, text = "파일명", font = ("궁서체", 20), fg = "blue")
btnPrev.place(x = 150, y = 10)
btnNext.place(x = 500, y = 10)
pLabel.place(x = 300, y = 10)
window.mainloop()
```

70. 다음 설명 중 거리가 먼 것을 모두 고르시오.

- ① <Button> : 모든 버튼의 공통 클릭할 때
- ② <Button-2> : 오른쪽 버튼 클릭할 때
- ③ <ButtonRelease-1> : 왼쪽 버튼 떼었을 때
- ④ <ButtonDouble-1> : 왼쪽 버튼 더블 클릭할 때
- ⑤ <ButtonDrag-1> : 왼쪽 버튼 드래그 할 때
- ⑥ <Enter> : 위젯 위로 마우스 커서가 올라갈 때

71. 다음 실행 결과가 나오도록 고르시오.

실행결과 :



```
from tkinter import *
```

```
def clickMouse(event) :
```

```
    [REDACTED]
```

```
window = Tk()
```

```
window.geometry("400x400")
```

```
label1 = Label(window, text = "이곳이 바뀜")
```

```
window.bind("<Button>", clickMouse)
```

```
label1.pack(expand = 1, anchor = CENTER)
```

```
window.mainloop()
```

- ① txt = str(event.y) + "," + str(event.x) + " 에서 클릭됨"
window.configure(text = txt)
- ② txt = str(mouse.y) + "," + str(mouse.x) + " 에서 클릭됨"
label1.configure(text = txt)
- ③ txt = str(event.y) + "," + str(event.x) + " 에서 클릭됨"
label1.configure(text = txt)
- ④ txt = str(event.y) + "," + str(event.x) + " 에서 클릭됨"
label1.chage(text = txt)

72. 다음 중 키보드 이벤트의 키가 아닌 것을 고르시오.

- ① <Enter> ② <BackSpace> ③ <Tab> ④ <Shift_L> ⑤ <Escape> ⑥ <End>

73. 다음 객체지향과 관련된 설명 중 거리가 먼 것을 고르시오.

- ① 자바, C++ 등에서도 사용되는 개념이다
- ② 클래스라는 핵심적인 단어를 사용한다
- ③ 속성은 색상, 속도 등 자동차의 상태를 표현한다
- ④ 기능은 변수 형태로 표현한다

74. 다음 생성자와 관련된 설명 중 거리가 먼 것을 고르시오.

- ① 인스턴스가 생성되면 자동으로 호출되는 메서드이다
- ② 생성자의 이름은 `__initilaize()`로 사용한다
- ③ 생성자는 반드시 클래스 안에 1개이상 코드로 작성해야 한다
- ④ 생성자는 여러 개 만들 수 있다

75. 다음 인스턴스 변수와 클래스 변수 관련된 설명 중 거리가 먼 것을 고르시오.

- ① 클래스 변수를 선언할 때는 `static`을 앞에 붙인다.
- ② 인스턴스 변수와 클래스 변수는 선언하는 시점에는 구분할 수 없다.
- ③ 인스턴스 변수는 `self.변수명` 형식으로 사용한다.
- ④ 클래스 변수는 아무 것도 붙이지 않고 변수명 형식으로 사용한다.

76. 다음 코드의 출력 결과를 작성하시오.

```
class Car :
    def method(self):
        print("슈퍼 클래스")

class Sedan(Car) :
    def method(self):
        print("서브 클래스")

myCar = Car()
mySedan = Sedan()
myCar.method()
mySedan.method()
```

- ① 서브 클래스
- ② 슈퍼 클래스
- ③ 슈퍼 클래스
서브 클래스
- ④ 서브 클래스
슈퍼 클래스

77. Car 클래스를 상속받는 RVCar 클래스를 정의하기 위해 코드를 작성하시오.

```
class Car :
    speed = 0

    def upSpeed(self, value) :
        self.speed = self.speed + value

class RVCar(Car) :
    seatNum = 0

    def getSeatNum(self) :
        return self.seatNum
```


78. 다음 괄호안에 들어갈 용어를 보기에서 고르시오.

`__repr__()`, `__le__()`, `__add__()` , `__del__()`, `__lt__()`, `__gt__()`, `__ge__()`, `__eq__()`, `__ne__()`

- ① (1) : 소멸자라고 부르며 인스턴스가 제거될 때는 자동으로 호출된다
- ② (2) : 인스턴스 사이의 덧셈작업이 일어나면 실행되는 메서드다
- ③ (3) : 인스턴스를 `print()` 문으로 출력할 때 실행되는 메소드다
- ④ (4) : 인스턴스 사이의 비교 연산자(`<=`)가 사용될 때 호출되는 메서드다

79. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :

Traceback (most recent call last):

File "c:\Users\Wrryan\Downloads\Untitled-1.py", line 9, in <module>
 sub.method()

File "c:\Users\Wrryan\Downloads\Untitled-1.py", line 3, in method

(오버라이딩 하지 않을 때 강제로 발생하는 오류)

(오버라이딩 하지 않을 때 강제로 발생하는 오류)

```
class SuperClass :
```

```
    def method(self) :
```

```
        (    1    )
```

```
class SubClass(SuperClass) :
```

```
    pass
```

```
sub = SubClass()
```

```
sub.method()
```

80. 다음 실행 결과가 나오도록 코드를 작성하시오.

실행결과 :

@얼룩말

#조랑말

@얼룩말

#조랑말

@얼룩말

#조랑말

```
import threading
```

```
import time
```

```
class RunningHorse :
```

```
    horseName = ''
```

```
    def __init__(self, name) :
```

```
        self.horseName = name
```

```
    def runHorse(self) :
```

```
        for _ in range(0, 3) :
```

```
            print(self.horseName)
```

```
            time.sleep(0.1)
```

```
horse1 = (1) ('@얼룩말')
```

```
horse2 = (1) ('#조랑말')
```

```
thread1 = threading.Thread(target = horse1.(2) )
```

```
thread2 = threading.Thread(target = horse2.(2) )
```

```
thread1.(3)
```

```
thread2.(3)
```