|  |
| --- |
| **R E P O R T** |

|  |
| --- |
| **[ 컴퓨팅 사고]** |



|  |  |
| --- | --- |
| **학 과** | **> 컴퓨터공학부** |
| **교수님** | **> 김종남교수님** |
| **학 번** | **> 201911608** |
| **이 름** | **> 김지환** |
| **제출일** | **> 2022.12.12** |



**목 차**

**과제 8 – 7장 파이썬 자료구조 II 연습문제** **……………… 03p**

**과제 9 – 8장 객체와 클래스 연습문제 …..…………………… 07p**

**과제 10 - 9장 GUI 프로그래밍 연습문제 …………………… 09p**

**과제 11 – 10장 파일과 예외처리 연습문제 ……………… 18p**

**과제 12 – 11장 람다식, 모듈, etc 연습문제** **……………… 23p**

**과제 13 – 12장 상속 연습문제 ………………………… 26p**

**과제 8 – 7장 파이썬 자료구조 II(튜플 ,딕셔너리, 세트, 문자열)**

02. (x, x\*x) 형식의 숫자 (1과 10 사이)를 포함하는 딕셔너리를 생성하고 출력하는 프로그램을 작성해보자.

ex) 실행결과

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}

|  |
| --- |
| =======Source Code======= |
| dic = {} for i in range(1, 11):  if i not in dic:  dic[i] = i\*i print(dic) |
| ========실행결과========= |
|  |

05. 딕셔너리 쇼핑몰에서 구입한 상품의 가격이 저장되어 있다. 딕셔너리에 있는 모든 상품 가격의 합계를 계산하는 프로그램을 작성해보자.

myDict = {“옷”: 100, “컴퓨터”: 2000, “모니터”: 320}

ex) 실행결과

총합계 = 2420

|  |
| --- |
| =======Source Code======= |
| myDict = {"옷": 100, "컴퓨터": 2000, "모니터": 320} print(f"총합계 = {sum(myDict.values())}") |
| ========실행결과========= |
|  |

07. 일정 애플리케이션을 작성하고자 한다. 일정 애플리케이션을 구현하는 한 가지 방법은 딕셔너리를 사용하고 날짜를 키로, 일정을 값으로 저장하면 된다. 하지만 이 방법은 특정한 날짜에 하나의 일정이 있는 경우에만 유효하다. 특정한 날짜에 여러 개의 일정을 저장할 수 있게 하려면 어떤 방법을 사용할 수 있는가?

Ex) 실행결과

날짜를 입력하시오: 2020 3 4

일정을 입력하시오: "파이선 과제 제출!!"

날짜를 입력하시오: 2020 4 2

일정을 입력하시오: "파이썬 프로젝트 제안서 발표!!"

|  |
| --- |
| =======Source Code======= |
| cal = {} for i in range(2):  date = input("날짜를 입력하시오: ")  date = date.replace(' ', '-')  rt = input("일정을 입력하시오: ")  if date not in cal:  cal[date] = [rt]  else:  cal[date].append(rt)  print()  for i in cal:  print(f"{i}의 일정은 {cal[i]}입니다.") |
| ========실행결과========= |
|  |

09. 2개의 문자열을 받아서 이들 문자열에 모두 포함된 글자를 반환하는 프로그램을 작성해보자.

Ex) 실행결과

첫 번째 문자열 : Hello World!

두 번째 문자열 : Hi! Welcome!!

모두 포함된 글자: H W o e

|  |
| --- |
| =======Source Code======= |
| str1 = input("첫 번째 문자열 : ") str2 = input("두 번째 문자열 : ")  print() intersection = ", ".join(list(set(str1) & set(str2))) print(f"모두 포함된 글자: {intersection}") |
| ========실행결과========= |
|  |

* **교과서 오타인 것 같습니다. 교집합이 실행 예시보다 더 많습니다.**

12. 2개의 문자열을 받아서 이들 문자열에 모두 포함된 글자를 반환하는 프로그램을 작성해보자.

Ex) 실행결과

문자열을 입력하시오: I read a java book

금칙어를 입력하시오: java red

I read a \*\*\*\* book

|  |
| --- |
| =======Source Code======= |
| str = input("문자열을 입력하시오: ") ben = list(input("금칙어를 입력하시오: ").split()) for i in ben:  str = str.replace(i, "\*"\*len(i)) print(str) |
| ========실행결과========= |
|  |

**과제 9 – 8장 객체와 클래스 프로그래밍 연습문제**

01. 고양이를 클래스로 정의하고 몇 개의 인스턴스를 생성해보자. 접근자와 설정자를 사용해보자.

missy = Cat('Missy', 3)  
lucky = Cat('Lucky', 5)  
print(missy)  
print(lucky)

ex) 실행 결과

Missy 3

Lucky 5

|  |
| --- |
| =======Source Code======= |
| class Cat:  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  def \_\_str\_\_(self):  return f"{self.name} {self.age}" missy = Cat('Missy', 3) lucky = Cat('Lucky', 5) print(missy) print(lucky) |
| ========실행결과========= |
|  |

02. 로켓을 나타내는 Rocket 클래스를 작성해보자. Rocket 클래스는 다음과 같은 인스턴스 변수와 메소드를 가진다.

myRocket = Rocket()  
print("로켓의 높이:", myRocket.y)  
  
myRocket.moveUp()  
print("로켓의 높이:", myRocket.y)

ex) 실행결과

로켓의 높이: 0

로켓의 높이: 1

|  |
| --- |
| =======Source Code======= |
| class Rocket:  def \_\_init\_\_(self):  self.x = 0  self.y = 0  def moveUp(self):  self.y += 1  myRocket = Rocket() print("로켓의 높이:", myRocket.y)  myRocket.moveUp() print("로켓의 높이:", myRocket.y) |
| ========실행결과========= |
|  |

04. 사각형을 나타내는 Rectangle 클래스를 작성하여 보자. Rectangle 클래스는 다음과 같은 인스턴스 변수와 메소드를 가진다.

r1 = Rectangle(0, 0, 100, 100)  
r2 = Rectangle(10, 10, 100, 100)  
r1.overlap(r2)

ex) 실행결과

r1과 r2는 서로 겹칩니다.

|  |
| --- |
| =======Source Code======= |
| class Rectangle:  def \_\_init\_\_(self, x, y, w, h):  self.x = x  self.y = y  self.width = w  self.height = h  def getArea(self):  x = self.x + self.width  y = self.y - self.height  return (self.x, x, self.y, y)  def overlap(self, r):  left, right, top, bottom = self.getArea()  r\_left, r\_right, r\_top, r\_bottom = r.getArea()  if left < r\_right and right > r\_left and top > r\_bottom and bottom < r\_top:  print("r1과 r2는 서로 겹칩니다.")  else:  print("r1과 r2는 서로 겹치지 않습니다.")  r1 = Rectangle(0, 0, 100, 100) r2 = Rectangle(10, 10, 100, 100) r1.overlap(r2) |
| ========실행결과========= |
|  |

06. Person이라는 클래스를 작성해보자. Person 클래스는 다음과 같은 인스턴스 변수와 메소드를 가진다.

p1 = Person("Kim", office="1234567", email="kim@company.com")  
p2 = Person("Park", office="2345678")  
p2.setEmail("park@company.com")

|  |
| --- |
| =======Source Code======= |
| class Person:  def \_\_init\_\_(self, n, mobile=None, office=None, email=None):  self.name = n  self.mobile = mobile  self.office = office  self.email = email  def \_\_str\_\_(self):  s = str(self.name)+" "+str(self.mobile)+" "+str(self.office)+" "+str(self.email)  s = s.replace("None", "미기재")  return s  def setEmail(self, e):  self.email = e  def setMobile(self, m):  self.mobile = m  def setOffice(self, o):  self.office = o  p1 = Person("Kim", office="1234567", email="kim@company.com") p2 = Person("Park", office="2345678") print(p1) print(p2) p2.setEmail("park@company.com") print(p1) print(p2) |
| ========실행결과========= |
|  |

08. printSong이라는 클래스를 작성해보자. printSong의 생성자는 노래의 가사를 리스트 형태로 받아서 객체의 내부에 저장한다. sing() 메소드는 한 줄에 한 항목씩 출력한다.

aSong = printSong(["TWINKLE, twinkle, little star,",  
 "How I wonder what you are!",  
 "Up above the world so high,",  
 "Like a diamond in the sky"])  
aSong.sing()

ex) 실행결과

TWINKLE, twinkle, little star,

How I wonder what you are!

Up above the world so high,

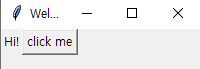
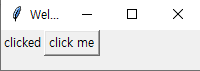
Like a diamond in the sky

|  |
| --- |
| =======Source Code======= |
| class printSong:  def \_\_init\_\_(self, s):  self.s = s  def sing(self):  for i in self.s:  print(i)  aSong = printSong(["TWINKLE, twinkle, little star,",  "How I wonder what you are!",  "Up above the world so high,",  "Like a diamond in the sky"]) aSong.sing() |
| ========실행결과========= |
|  |

**과제 10 - 9장 GUI 프로그래밍 연습문제**

01. 다음과 같이 하나의 레이블과 하나의 버튼을 가지는 프로그램을 작성해보자. 버튼을 누르면 레이블은 “clicked”를 표시한다.

Ex) 실행결과

 -> 

|  |
| --- |
| =======Source Code======= |
| from tkinter import \* tk = Tk() tk.title("Welcome to tkinter") tk.geometry('200x40') lb = Label(window, text="Hi!") lb.grid(column=0) def clicked():  if lb['text'] == "Hi!":  lb.configure(text="clicked")  else:  lb.configure(text="Hi!") btn = Button(window, text="click me", command=clicked) btn.grid(column=1, row=0)  tk.mainloop() |
| ========실행결과========= |
| -> |

02. 다음과 같이 하나의 레이블을 가지는 프로그램을 작성해보자. 레이블의 배경색은 오렌지색으로 하고, 글자 색은 파랑색으로 한다. 레이블의 크기는 50x3으로 한다.

Ex) 실행결과

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| =======Source Code======= |
| from tkinter import \* tk = Tk() tk.title("Welcome to tkinter") tk.geometry('350x50') lb = Label(tk, width=50, height=3, text="Hello, I'm Label", fg='blue', bg='orange') lb.grid(column=0) tk.mainloop() |
| ========실행결과========= |
|  |

04. 주소를 입력하는 다음과 같은 애플리케이션을 작성해보자. 아래와 최대한 유사하게 작성하라.

Ex) 실행결과

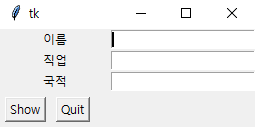
텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| =======Source Code======= |
| from tkinter import \* tk = Tk() tk.title("tk") tk.geometry('400x150')  frame = Frame(width=397, height=114, relief='sunken', bd=1)  name = Label(frame, text="이름:") addr = Label(frame, text="주소:") area = Label(frame, text="도:") code = Label(frame, text="우편번호:") cont = Label(frame, text="국가:") labelList = [name, addr, area, code, cont]  e\_name = Entry(frame, width=47) e\_addr = Entry(frame, width=47) e\_area = Entry(frame, width=47) e\_code = Entry(frame, width=47) e\_cont = Entry(frame, width=47) entryList = [e\_name, e\_addr, e\_area, e\_code, e\_cont]  def \_\_clear\_\_():  for i in entryList:  i.delete(0,"end") def \_\_submit\_\_():  for i, j in zip(labelList, entryList):  print(f"{i['text']} {j.get()}")  \_\_clear\_\_()  clear = Button(tk, text="Clear", command=\_\_clear\_\_, width=7) sumbit = Button(tk, text="Submit", command=\_\_submit\_\_, width=9)  frame.place(x=1, y=2)  name = name.place(x=26, y=2) addr = addr.place(x=26, y=24) area = area.place(x=37, y=46) code = code.place(x=0, y=68) cont = cont.place(x=26, y=90)  e\_name = e\_name.place(x=60, y=2) e\_addr = e\_addr.place(x=60, y=24) e\_area = e\_area.place(x=60, y=46) e\_code = e\_code.place(x=60, y=68) e\_cont = e\_cont.place(x=60, y=90)  clear = clear.place(x=250,y=120) sumbit = sumbit.place(x=315,y=120)  tk.mainloop() |
| ========실행결과========= |
|  |

09. 데이터를 입력받을 때 사용할 수 있는 다음과 같은 애플리케이션을 작성해보자.

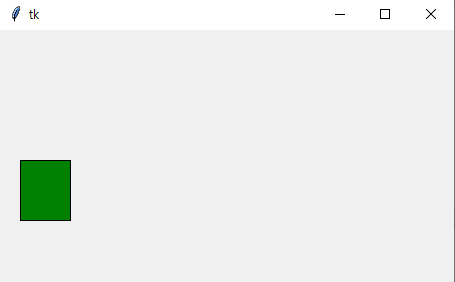
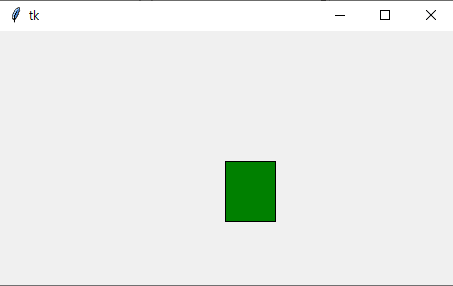
Ex) 실행결과



|  |
| --- |
| =======Source Code======= |
| from tkinter import \* fields = '이름', '직업', '국적' tk = Tk() entries = []  def fetch():  for i in entries:  print(f"{i[0]}: {i[1].get()}")  for i in fields:  frame = Frame()  lb = Label(frame, width=15, text=i)  ent = Entry(frame)  frame.pack(side=TOP)  lb.pack(side=LEFT)  ent.pack(side=RIGHT)  entries.append((i, ent))  b1 = Button(tk, text='Show', command=fetch) b1.pack(side=LEFT, padx=5, pady=5) b2 = Button(tk, text='Quit', command=tk.quit) b2.pack(side=LEFT, padx=5, pady=5) tk.mainloop() |
| ========실행결과========= |
|  |

13. 화면에서 사각형을 그리고, 화살표 키로 사각형을 움직이는 프로그램을 작성해 보자.

Ex) 실행결과

 -> 

|  |
| --- |
| =======Source Code======= |
| from tkinter import \*  def left(event):  canvas.move(rect, -5, 0)  def right(event):  canvas.move(rect, 5, 0)  def down(event):  canvas.move(rect, 0, 5)  def up(event):  canvas.move(rect, 0, -5)  tk = Tk() frame = Frame(tk, width=450, height=250) frame.bind('<Left>', left) frame.bind('<Right>', right) frame.bind('<Up>', up) frame.bind('<Down>', down) frame.focus\_set() frame.pack()  canvas = Canvas(frame, width=450, height=250) canvas.grid(row=0, column=0, columnspan=4) rect = canvas.create\_rectangle(20, 130, 70, 190, fill ="green")  tk.mainloop() |
| ========실행결과========= |
| -> |

**과제 11 – 10장 파일과 예외처리**

01. 파일 “input.txt”의 처음 3줄을 읽는 프로그램을 작성하라.

Ex) 실행결과

All's well that ends well.

Bad news travels fast.

well begun is half done.

|  |
| --- |
| =======Source Code======= |
| fp = open("input.txt", "r") for i in range(3):  print(fp.readline(), end="") fp.close() |
| ========실행결과========= |
| Input.txt |

03. 파일에서 임의의 행을 읽는 Python 프로그램을 작성하시오.

Ex) 실행결과

파일 이름을 입력하시오: input.txt

행 번호를 입력하시오: 3

3번 행은 Well begun is half done.입니다.

|  |
| --- |
| =======Source Code======= |
| fname = input("파일 이름을 입력하시오: ") line = int(input("행 번호를 입력하시오: ")) fp = open(fname, "r") list = fp.read().splitlines() print(f"{line}번 행은 {list[line-1]}입니다.") fp.close() |
| ========실행결과========= |
|  |

08. 학생들의 성적이 부동소수점 수로 파일 scores.txt에 저장되어 있다고 하자(메모장에서 ANSI 엔코딩으로 저장한다.) 이 성적을 읽어서 파일의 끝에 평균값을 추가하라.

|  |
| --- |
| =======Source Code======= |
| rf = open("scores.txt").read().splitlines() avg = 0 for i in rf:  avg += float(i) avg/=len(rf) fp = open("scores.txt","a") fp.writelines(f"\n평균값: {avg}") fp.close() |
| ========실행결과========= |
| -> |

09. 사용자로부터 파일 이름과 삭제할 문자열을 입력받는다. 파일을 열어서 사용자가 원하는 문자열을 삭제한 후에 다시 파일에 쓴다.

Ex) 실행결과

파일 이름을 입력하시오: d:\\words.txt

삭제할 문자열을 입력하시오: black

변경된 파일이 저장되었습니다.

|  |
| --- |
| =======Source Code======= |
| file = input("파일 이름을 입력하시오: ") delete = input("삭제할 문자열을 입력하시오: ")  rf = open(file).read() rf = rf.replace(delete, "")  fp = open(file,"w") fp.write(rf) fp.close()  print("변경된 파일이 저장되었습니다.") |
| ========실행결과========= |
| -> |

12. 트윗 메시지에서 사용자 메시지만 추려보자. 즉 특수 문자나 URL, 해쉬태그, 이메일 주소, RT, CC는 삭제한다.

|  |
| --- |
| =======Source Code======= |
| twit = " ".join(filter(str.isalnum, list(input("트윗 문자열: ").split()))) twit = twit.replace("RT", "") twit = twit.replace("CC", "") print("정제된 문자열:"+twit) |
| ========실행결과========= |
|  |

**과제12 – 11장 내장함수, 람다식, 제너레이터, 모듈 연습문제**

02. eval() 함수를 사용하여서 사용자가 어떤 문자열을 입력하더라도 오류를 일으키지 않고 정수나 실수로 변환하는 코드를 작성해보자.

Ex) 실행결과

정수나 실수를 입력하시오: 10.2

10.2

정수나 실수를 입력하시오: 10

10

|  |
| --- |
| =======Source Code======= |
| for i in range(2):  num = eval(input("정수나 실수를 입력하시오: "))  print(num) |
| ========실행결과========= |
|  |

03. 람다식을 사용하여 튜플을 정렬하는 프로그램을 작성하라.

Ex) 실행결과

원래의 리스트

[('국어', 88), ('수학', 90), ('영어', 99), ('자연', 82)]

정렬된 리스트

[('자연', 82), ('국어', 88), ('수학', 90), ('영어', 99)]

|  |
| --- |
| =======Source Code======= |
| subjectScore = [('국어', 88),('수학', 90),('영어', 99),('자연', 82)] print(f"원래의 리스트\n{subjectScore}") subjectScore.sort(key=lambda x:x[1]) print(f"정렬된 리스트\n{subjectScore}") |
| ========실행결과========= |
|  |

06. 팩토리얼을 계산하는 함수 fact(n)을 작성하고 이것을 factorial이라는 모듈로 독립시켜 보자. 다른 파일에서 factorial 모듈을 포함시켜서 안에 들어 있는 함수는 사용해보자.

|  |
| --- |
| =======Source Code======= |
| #factoral.py  factorial = [1] def fact(n):  for i in range(len(factorial), n+1):  factorial.append(factorial[i-1]\*i)  return factorial[n]  #test.py  import factorial print(f"5! = {factorial.fact(5)}") |
| ========실행결과========= |
|  |

11. 원을 나타내는 클래스 Circle에 +, >, < 연산자를 중복 정의해보자.

class Circle:  
 def \_\_init\_\_(self, radius):  
 self.\_\_radius = radius

|  |
| --- |
| =======Source Code======= |
| class Circle:  def \_\_init\_\_(self, radius):  self.\_\_radius = radius  def \_\_add\_\_(self, other):  return Circle(self.\_\_radius+other.\_\_radius)  def \_\_gt\_\_(self, other):  return self.\_\_radius > other.\_\_radius  def \_\_lt\_\_(self, other):  return self.\_\_radius < other.\_\_radius  def \_\_str\_\_(self):  return f"rad = {self.\_\_radius}" c1 = Circle(10) c2 = Circle(15) c3 = c1+c2 print(c1>c2) print(c1<c2) print(c3) |
| ========실행결과========= |
|  |

13. random 모듈에 있는 함수를 사용하여서 알파벳 a부터 z 사이에서 랜덤하게 10개의 문자를 출력하여 보자.

|  |
| --- |
| =======Source Code======= |
| import random alphabet = [chr(i) for i in range(97,123)] for i in range(10):  print(random.choice(alphabet), end=", " if i<9 else "") |
| ========실행결과========= |
|  |

**과제 13 – 12장 상속 연습문제**

01. 2차원 공간의 한 점 (x, y)를 나타내는 클래스 Point를 정의한다. Point 클래스의 \_\_init\_\_() 메소드는 sell, x, y를 받아서 멤버 변수에 할당한다. \_\_str\_\_()을 정의하여 “(x, y)” 형태의 문자열을 반환한다. Point를 상속받아서 3차원 공간의 한 점 (x, y, z)을 나타내는 Point3D 클래스를 정의해보자.

|  |
| --- |
| =======Source Code======= |
| class Point:  def \_\_init\_\_(self, x, y):  self.x = x  self.y = y  def \_\_str\_\_(self) :  return f"({self.x}, {self.y})"  class Point3D(Point):  def \_\_init\_\_(self,x,y,z):  super().\_\_init\_\_(x, y)  self.z=z  def \_\_str\_\_(self) :  return f"({self.x}, {self.y}, {self.z})"  point2D = Point(3, 4) point3D = Point3D(3, 4, 5) print(f"{point2D}, {point3D}") |
| ========실행결과========= |
|  |

02. 주소를 나타내는 Address와 사람을 나타내는 Person 클래스를 정의한다. Address와 Person을 동시에 상속 받아서 Contact 클래스를 정의해보자. Contact 클래스는 연락처를 나타낸다.

class Address:  
 def \_\_init\_\_(self, street, city):  
 self.street = str(street)  
 self.city = str(city)  
class Person:  
 def \_\_init\_\_(self, name, email):  
 self.name = str(name)  
 self.email = str(email)

|  |
| --- |
| =======Source Code======= |
| class Address:  def \_\_init\_\_(self, street, city):  self.street = str(street)  self.city = str(city) class Person:  def \_\_init\_\_(self, name, email):  self.name = str(name)  self.email = str(email) class Contact(Address, Person):  def \_\_init\_\_(self, name, email, city, street):  Person.\_\_init\_\_(self, name, email)  Address.\_\_init\_\_(self, street, city)  def \_\_str\_\_(self):  return f"이름: {self.name}\n" \  f"이메일: {self.email}\n" \  f"주소: {self.city}-{self.street}" c = Contact("김지환","aa1535@pukyong.ac.kr","부산광역시","남구 대연동 ~~~") print(c) |
| ========실행결과========= |
|  |

06. 대학교에는 학과(Department), 교과목(Course), 학생(Student)들이 존재한다. 학과에는 많은 교과목들이 개설될 수 있고, 하나의 교과목에는 여러 학생들이 등록할 수 있다. Has-a 관계를 염두에 두고 이것을 다음과 같이 프로그래밍 해보자.

dept = Department("컴퓨터")  
math1 = dept.add\_course("공업수학", 3)  
math2 = dept.add\_course("이산수학", 2)  
  
kim = Student("Kim", 20200001)  
kim.enroll(math1)

|  |
| --- |
| =======Source Code======= |
| class Course:  def \_\_init\_\_(self, department, title, credit):  self.department = department  self.title = title  self.credit = credit  self.students = []  def show(self):  print(f"강의 명:{self.title}\n수강학점: {self.credit}\n개설학과: {self.department}학과")  print("======수강 학생======")  if self.students:  for student in self.students:  print(student.name, student.studentNum)  else:  print("수강학생이 없습니다.")  print()  class Department:  def \_\_init\_\_(self, name):  self.name = name  self.subjects = []  def add\_course(self, title, credit):  for subject in self.subjects:  if title == subject.title:  return  course = Course(self.name, title, credit)  self.subjects.append(course)  return course  def show(self):  print(f"{self.name}학과 커리큘럼\n")  for subject in self.subjects:  subject.show() class Student:  def \_\_init\_\_(self, name, studentNum):  self.name = name  self.studentNum = studentNum   def enroll(self, course):  course.students.append(self)  dept = Department("컴퓨터") math1 = dept.add\_course("공업수학", 3) math2 = dept.add\_course("이산수학", 2)  kim = Student("Kim", 20200001) kim.enroll(math1)  dept.show() |
| ========실행결과========= |
|  |

07. 하나의 앨범(Album)에는 많은 노래(Song)가 담길 수 있다. 노래를 부른 가수 (Artist)는 여러 노래를 소유할 수 있다. 재생 목록(Playlist)은 여러 노래를 가질 수 있다. Has-a 관계를 염두에 두고 프로그래밍 해보자.

lee = Artist("Lee's Band")  
album = Album("첫 번째 앨범", lee, 2020)  
album.add\_track("첫 번째 노래")  
album.add\_track("두 번째 노래")  
  
playlist = Playlist("애창곡")

|  |
| --- |
| =======Source Code======= |
| class Song:  def \_\_init\_\_(self, title, artist, album, track\_number):  self.title = title  self.artist = artist  self.album = album  self.track\_number = track\_number  artist.add\_song(self)  def show(self):  print(f"{self.track\_number}-{self.artist.name}/{self.title}\n앨범 정보")  self.album.show()  class Album:  def \_\_init\_\_(self, title, artist, year):  self.title = title  self.artist = artist  self.year = year  self.tracks = []  artist.add\_album(self)   def add\_track(self, title, artist=None):  if artist is None:  artist = self.artist   track\_number = len(self.tracks)  song = Song(title, artist, self, track\_number)  self.tracks.append(song)  def show(self):  print(f"{self.title}\n{self.artist.name}\n{self.year}\n")  class Artist:  def \_\_init\_\_(self, name):  self.name = name  self.albums = []  self.songs = []   def add\_album(self, album):  self.albums.append(album)   def add\_song(self, song):  self.songs.append(song)   class Playlist:  def \_\_init\_\_(self, name):  self.name = name  self.songs = []   def add\_song(self, song):  self.songs.append(song)  lee = Artist("Lee's Band") album = Album("첫 번째 앨범", lee, 2020) album.add\_track("첫 번째 노래") album.add\_track("두 번째 노래")  playlist = Playlist("애창곡")  for song in album.tracks:  playlist.add\_song(song)  for song in playlist.songs:  song.show() |
| ========실행결과========= |
|  |