

파이썬

상명대학교 융합공과대학

지능·데이터융합학부

휴먼지능정보공학전공

dkim@smu.ac.kr

강의개요

- 강의소개 및 프로그래밍 개념
 - 프로그래밍과 컴퓨팅사고력 소개
 - 프로그래밍 맛보기
- 변수, 자료형, 연산, 함수
 - 코딩과 기초실습
- 조건문, 연산자
 - 코딩과 기초실습
- 반복문
 - 코딩과 기초실습
- 함수, 매개변수
 - 코딩과 기초실습
- 중간고사

강의개요

- 자료형, 리스트
 - 코딩과 기초실습
- 자료형, 튜플
 - 코딩과 기초실습
- 자료형, 딕셔너리
 - 코딩과 기초실습
- 실습예제
 - 코딩과 기초실습
- 파일읽고 쓰기
 - 코딩과 기초실습
- 객체지향 프로그래밍
 - 코딩과 기초실습
- 기말고사

프로그래밍 문법

- 파이썬: 파이썬은 고급 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑 대화형 언어
 - 고급 프로그래밍 언어
 - 고급프로그래밍언어: 사람이 이해하기 쉽게 작성된 프로그래밍 언어 (C++, Java, Python), 가독성 컴파일러, 인터프리터에 의해 저급 프로그래밍언어로 번역되어 실행
 - 저급프로그래밍언어: 기계(컴퓨터)가 이해하기 쉽게 작성된 프로그래밍 언어(기계어, 어셈블리어)
 - 인터프리터식
 - 인터프리터식언어: 코드를 컴파일(Compile)하지 않고도 바로 실행할 수 있는 프로그래밍 언어
 - 컴파일 언어: 코드를 (한꺼번에) 기계어로 번역되어 실행할 수 있는 프로그래밍 언어

프로그래밍 문법

- 파이썬: 파이썬은 고급 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑 대화형 언어
 - 객체 지향적 언어
 - 객체지향적 언어: 물리적, 추상적인 것 중에서 자신의 속성을 가지고 있고 다른 것과 식별 가능한 것 (데이터와 함수)을 프로그래밍 언어로 표현 가능 (C++, Java, Python)
 - 절차지향적: 수행되어야 할 연속적인 계산(수행) 과정을 포함하여 프로그래밍 언어로 표현 가능 (C)
 - 대화형 언어
 - Command Prompt 환경 제공: Python 3.6.5 Shell (파이썬 명령프롬프트)
 - GUI(Graphical User Interface) 환경 제공: Turtle Graphic, Tkinter

프로그래밍 문법

- 파이썬: 파이썬은 고급 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑 대화형 언어

- 파이썬 객체: 클래스

- 클래스 정의와 생성

class 클래스 이름:

#클래스 데이터1,... → 필드1,...

#클래스 함수1,... → 메소드1

프로그래밍 문법

- 파이썬: 파이썬은 고급 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체 지향적, 동적 타이핑 대화형 언어

프로그래밍 문법

- 파이썬: 파이썬은 고급 프로그래밍 언어로, 플랫폼에 독립적이며 인터프리터식, 객체지향적, 동적 타이핑 대화형 언어

- 파이썬 객체: 클래스
 - 클래스 정의와 생성

```
class Car:
    #필드
    car_number=""
    car_color=""
    car_speed=0
    #메소드
    def speed_up(pedal_value):
        self.car_speed = self.car_speed + pedal_value
        print("자동차 속도는 %d 입니다"%(self.car_speed))
    def check_car_color(color):
        self.car_color = color
        print("자동차는 %색 입니다"%(self.car_color))
```


프로그래밍 문법

- 클래스 정의와 생성과 인스턴스

```
class Car:
    #필드
    car_number=""
    car_color=""
    car_speed=0
    #메소드
    def speed_up(pedal_value):
        self.car_speed = self.car_speed + pedal_value
        print("자동차 속도는 %d 입니다"%(self.car_speed))
    def check_car_color(color):
        self.car_color = color
        print("자동차는 %색 입니다"%(self.car_color))

myCar_first = Car()
myCar_first.number="1234"
myCar_first.speed_up(10) → 자동차 속도는 10입니다
myCar_second = Car()
myCar_second.number="4567"
myCar_second.speed_up(20) → 자동차 속도는 20입니다
```

프로그래밍 문법

- 파이썬 객체: 클래스와 인스턴스, 생성자
 - 생성자: 인스턴스를 생성하면 호출되는 메소드 (초기화 과정 작성)

class 클래스 이름:

def __init__(self, 매개변수, ...): ##클래스 생성자 (초기화)

self.클래스 데이터변수 이름= 매개변수

self.클래스 데이터변수 이름...

def 함수이름(self, 매개변수, ...):

print("self.클래스 데이터변수 이름")

myinstance1 = 클래스이름(데이터1, ...)

myinstance2 = 클래스이름(데이터2, ...)

프로그래밍 문법

- 파이썬 객체: 클래스와 인스턴스

- 필드나 메소드 사용

```
class 클래스 이름:
```

```
    def __init__(self, 매개변수, ...):
```

```
        self.클래스 데이터변수 이름 = 매개변수
```

```
        self.클래스 데이터변수 이름...
```

```
    def 함수이름(self, 매개변수, ...):
```

```
        print("self.클래스 데이터변수 이름")
```

```
myinstance1 = 클래스이름(데이터1, ...)
```

```
myinstance2 = 클래스이름(데이터2, ...)
```

```
myinstance1.필드이름1 = 값1
```

```
myinstance1.필드이름 2 = 값2
```

```
myinstance1.메소드이름1()
```

```
myinstance1.메소드이름2()
```

프로그래밍 문법

- 클래스 정의와 생성과 인스턴스, 생성자

```
class Car:
    def __init__(self, val1, val2, val3):
        self.car_number = val1
        self.car_color = val2
        self.car_speed = val3

    def speed_up(self, pedal_value):
        self.car_speed = self.car_speed + pedal_value
        print("자동차 속도는 %d 입니다" % (self.car_speed))

    def check_car_color(self, color):
        self.car_color = color
        print("자동차는 %색 입니다" % (self.car_color))

myCar_first = Car("1234", "Black", 10)
myCar_first.speed_up(10) → 자동차 속도는 20입니다
myCar_second = Car("4567", "Silver", 20)
myCar_second.speed_up(20) → 자동차 속도는 40입니다
```

프로그래밍 문법

- Tkinter: 파이썬에서 그래픽 사용자 인터페이스(GUI: graphical user interface)를 개발할 때 필요한 모듈
 - Button: 간단한 버튼
 - Canvas: 화면에 어떤 것을 그릴 때 사용
 - Checkbutton: 2가지의 구별되는 값을 가지는 변수를 표현
 - Entry: 한 줄의 텍스트를 입력받는 필드
 - Label: 텍스트나 이미지를 표시
 - Menu: 메뉴를 표시(풀다운, 팝업 메뉴)
 - Radiobutton: 여러 값을 가질 수 있는 변수를 표시
 - Text: 형식을 가지는 텍스트를 표시(여러 가지 스타일과 속성으로 텍스트를 표시)

프로그래밍 문법

- Tkinter: 파이썬에서 그래픽 사용자 인터페이스(GUI: graphical user interface)를 개발할 때 필요한 모듈
 - 버튼이 있는 윈도우

```
from tkinter import *  
window = Tk()  
button = Button(window, text="버튼입니다")  
button.pack()  
window.mainloop()
```

프로그래밍 문법

- Tkinter: 파이썬에서 그래픽 사용자 인터페이스(GUI: graphical user interface)를 개발할 때 필요한 모듈

- 라벨과 엔트리(텍스트박스)가 있는 윈도우

```
from tkinter import *  
window = Tk()
```

```
Label1 = Label(window, text="첫번째")  
Label2 = Label(window, text="두번째")  
Label1.pack()  
Label2.pack()
```

```
Entry1 = Entry(window)  
Entry2 = Entry(window)  
Entry1.pack()  
Entry2.pack()
```

```
window.mainloop()
```

프로그래밍 문법

- Tkinter: 파이썬에서 그래픽 사용자 인터페이스(GUI: graphical user interface)를 개발할 때 필요한 모듈
 - 위치 배치

```
Label1 = Label(window, text="라벨1", bg="red", fg="white")  
Label2 = Label(window, text="라벨2", bg="green", fg="white")  
Label3 = Label(window, text="라벨1", bg="blue", fg="white")
```

```
Label1.place(x=0,y=0)  
Label2.place(x=20,y=20)  
Label3.place(x=40,y=40)
```

```
window.mainloop()
```

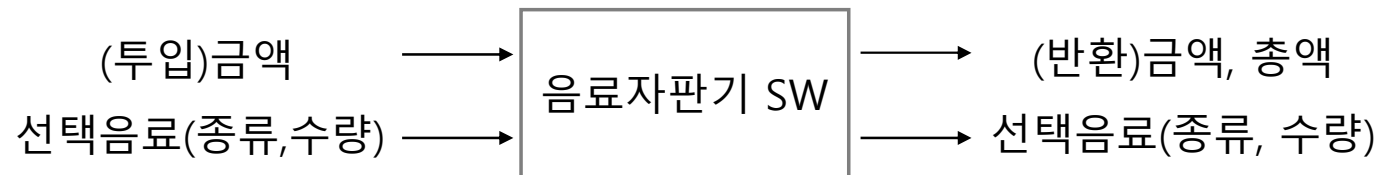

프로그래밍 설계

- 자판기



프로그래밍 설계

- 자판기 - 음료자판기 S/W - Level 0



모듈	음료자판기 S/W
입력	<ul style="list-style-type: none">- 투입금액- 3가지 선택음료(종류, 수량)
출력	<ul style="list-style-type: none">- 반환금액, 총액- 3가지 선택음료(종류, 수량)
기능	<ul style="list-style-type: none">- 사용자로부터 금액을 입력 받아 사용자가 선택한 음료를 잔액과 함께 제공

프로그래밍 설계

프로그래밍 명세

■ 화면설계

- 1번 엔트리 박스는 투입금액을 입력 받는다
- 2번 버튼은 투입 금액을 확인한다
- 3번 라벨은 투입 금액을 출력한다
- 4번 라벨은 구매 내역을 출력한다(음료의 종류, 수량)
- 5번 라벨은 결제 내역을 출력한다(결제금액, 거스름돈)
- 6번 버튼은 버튼은 캔커피를 표시한다 (버튼 클릭 시 음료 개수 증가)
- 7번 버튼은 소다를 표시하고 (버튼 클릭시 음료 개수 증가)
- 8번 버튼은 생수를 표시한다 (버튼 클릭시 음료 개수 증가)
- 9번 버튼은 음료를 계산한다

■ 시나리오 설계

- 사용자는 투입금액을 입력 하고 투입 버튼을 누른다
 - 버튼을 누르면 투입금액이 라벨에 출력된다
- 사용자는 원하는 음료 버튼을 누른다
- 사용자는 계산하기 버튼을 누른다
 - 사용자는 원하는 음료 버튼을 추가로 누른다 (버튼 클릭 시 음료 개수 증가)
 - 선택 음료(버튼 종류, 횟수)에 따라 구매, 결제 내역이 출력된다
 - 결제 금액이 투입 금액보다 크면 금액 부족 메시지를 출력한다

프로그래밍 설계

프로그래밍 명세

- 사용자로부터 투입금액과 선택음료의 종류와 수량을 확인하고 선택음료와 함께 계산된 반환금액을 제공한다

입력(Input)

- 투입금액
- 선택음료(종류, 수량)

- 데이터:
- 변수(이름, 형태):
- 자료형: 객체
- 연산자:
- 명령문:
- 함수:
- 이벤트: command

처리(Processing)

- 투입금액, 반환금액
- 선택음료 수량, 종류, 단가

- 데이터:
- 변수(이름, 형태):
- 자료형: 객체
- 연산자:
- 명령문:
- 함수:
- 이벤트: command

출력(Output)

- 반환금액, 총액
- 선택음료(종류, 수량)

- 데이터:
- 변수(이름, 형태):
- 자료형: 객체
- 연산자:
- 명령문:
- 함수:
- 이벤트: command

프로그래밍 설계

프로그래밍 명세

- 사용자로부터 투입금액과 선택음료의 종류와 수량을 확인하고 선택음료와 함께 계산된 반환금액을 제공한다



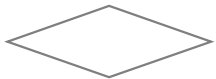
시작, 끝



초기값 선언(설정)



데이터 입,출력



조건에 따른 판단(분기)

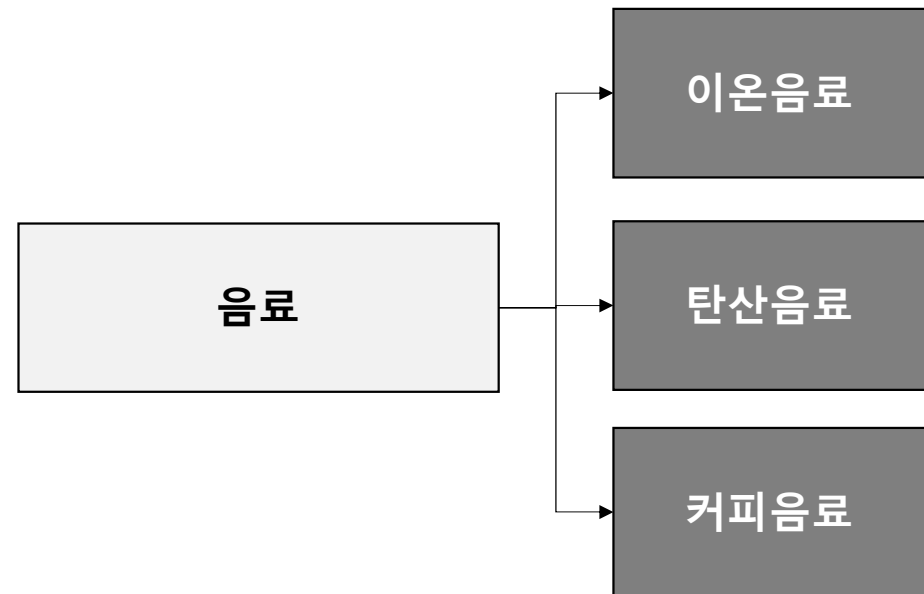
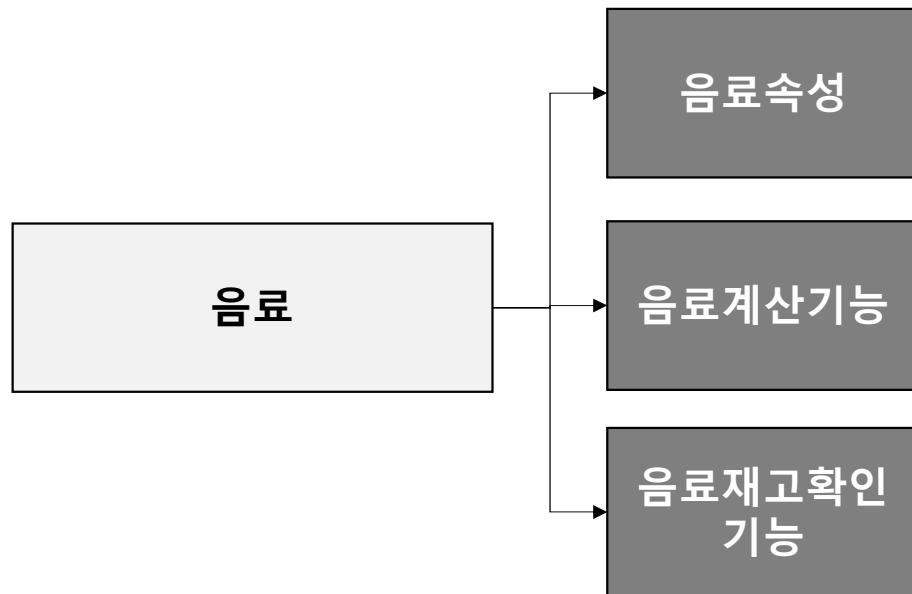


연산 및 처리



데이터 제어(흐름)

프로그래밍 설계



프로그래밍 설계

자판기 예제 (Ver 1.1)

상명 파이썬 자판기(Ver1.1)

투입

투입금액: 원

구매내용

결제내용







계산하기

프로그래밍 구현