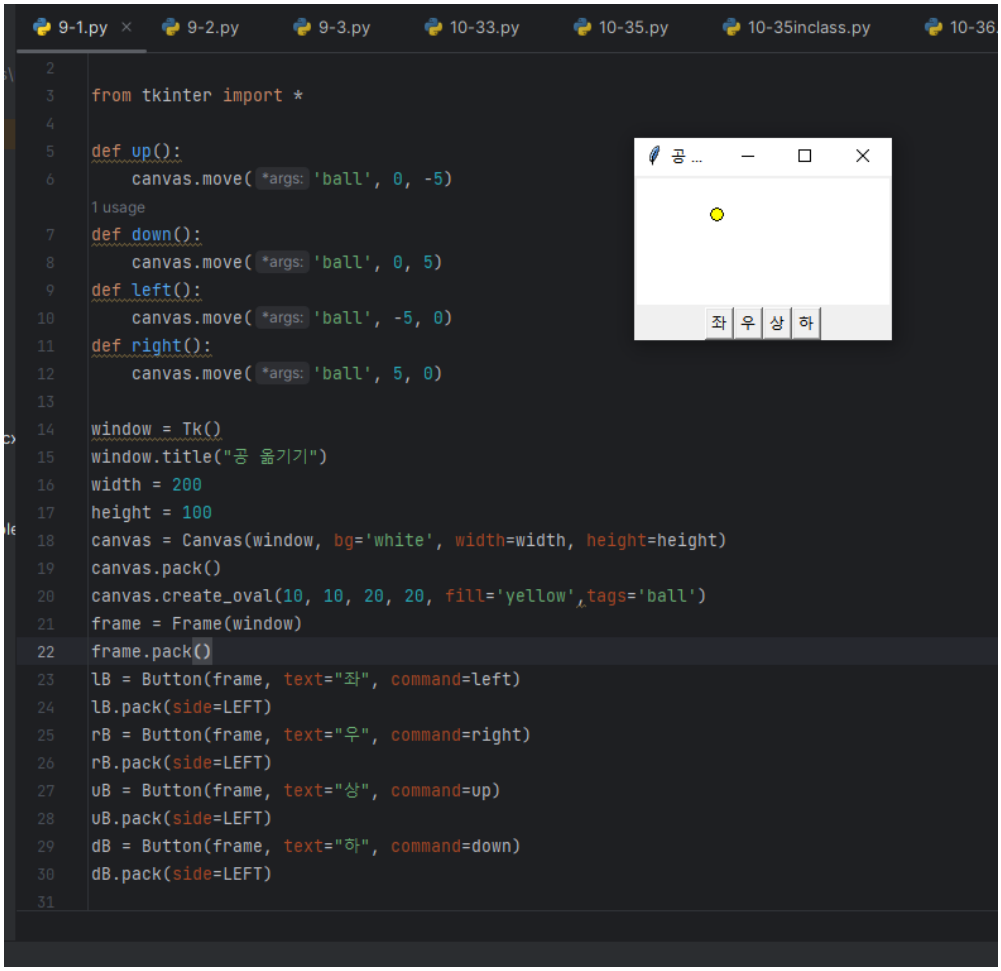
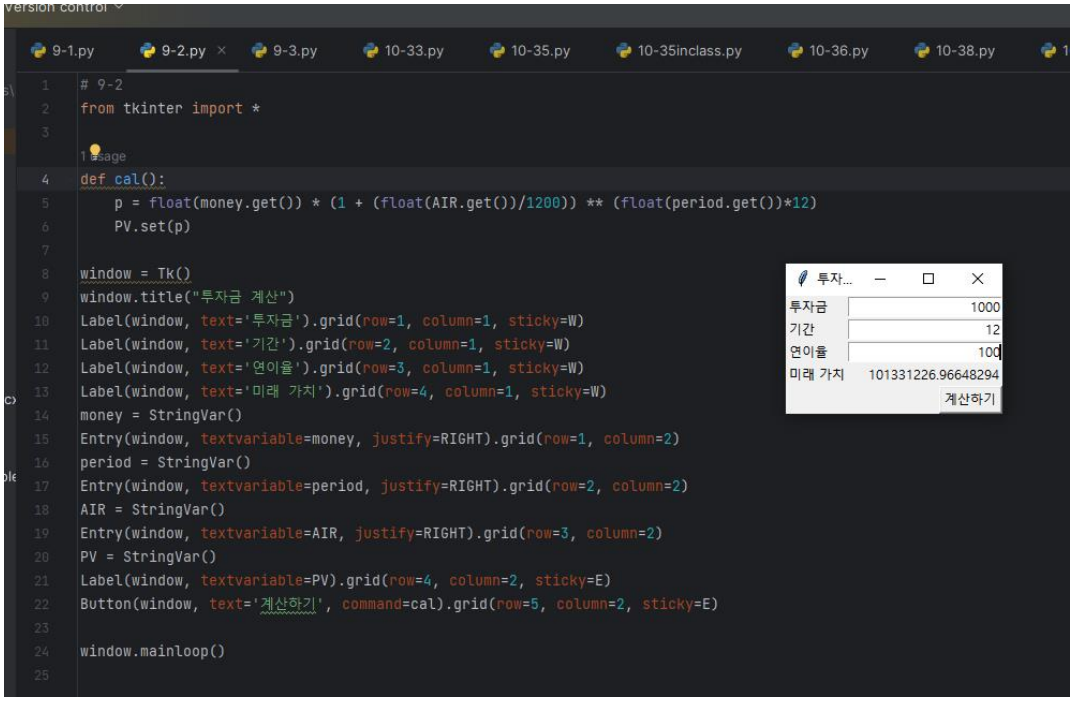
9-1.



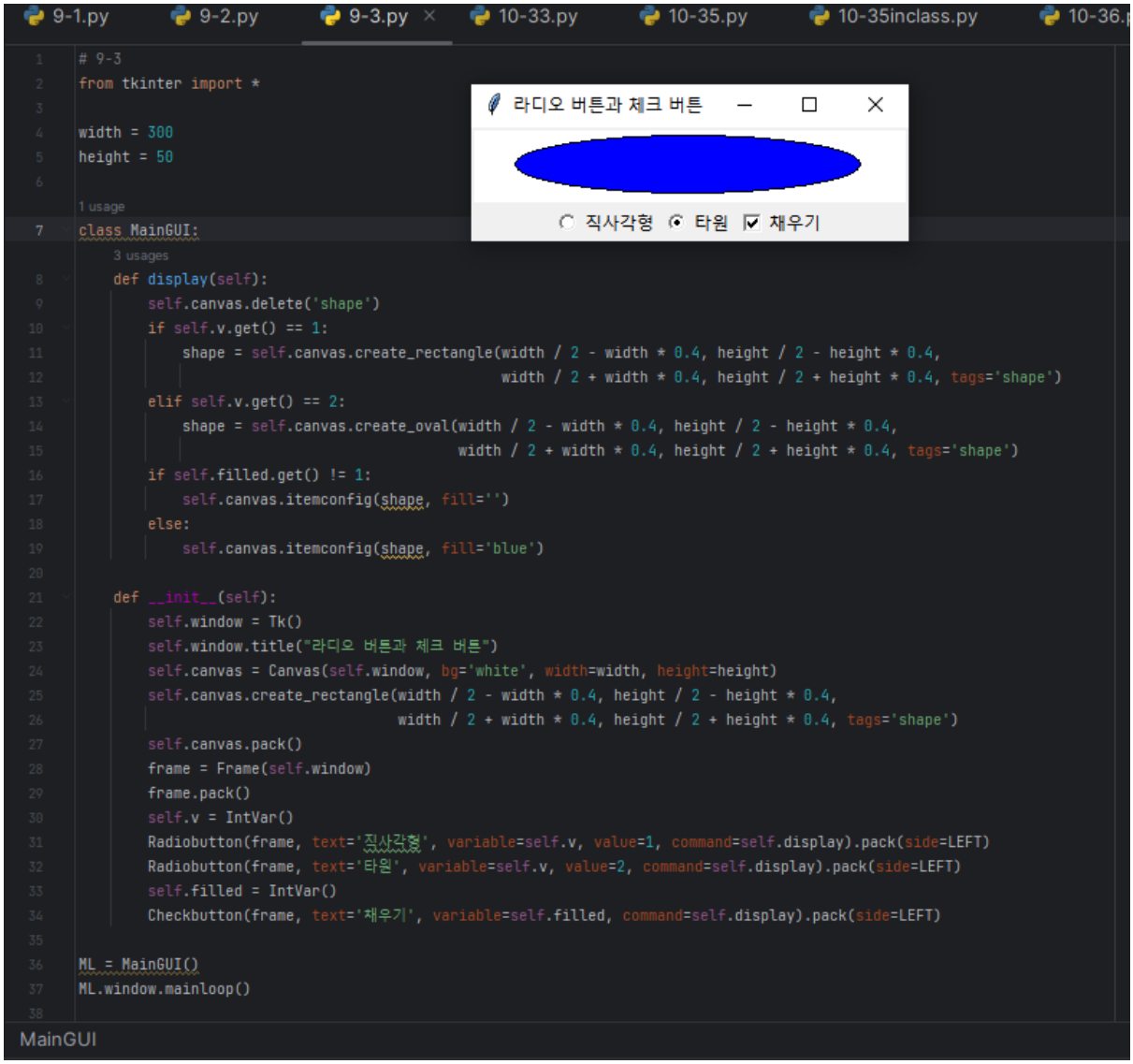

```python

from tkinter import *

def up():
    canvas.move( *args: 'ball', 0, -5)
1 usage
def down():
    canvas.move( *args: 'ball', 0, 5)
def left():
    canvas.move( *args: 'ball', -5, 0)
def right():
    canvas.move( *args: 'ball', 5, 0)

window = Tk()
window.title("공 옮기기")
width = 200
height = 100
canvas = Canvas(window, bg='white', width=width, height=height)
canvas.pack()
canvas.create_oval(10, 10, 20, 20, fill='yellow',tags='ball')
frame = Frame(window)
frame.pack()
lB = Button(frame, text="좌", command=left)
lB.pack(side=LEFT)
rB = Button(frame, text="우", command=right)
rB.pack(side=LEFT)
uB = Button(frame, text="상", command=up)
uB.pack(side=LEFT)
dB = Button(frame, text="하", command=down)
dB.pack(side=LEFT)

```

9-2.




```python
# 9-2
from tkinter import *

1 usage
def cal():
    p = float(money.get()) * (1 + (float(AIR.get())/1200)) ** (float(period.get())*12)
    PV.set(p)

window = Tk()
window.title("투자금 계산")
Label(window, text='투자금').grid(row=1, column=1, sticky=W)
Label(window, text='기간').grid(row=2, column=1, sticky=W)
Label(window, text='연이율').grid(row=3, column=1, sticky=W)
Label(window, text='미래 가치').grid(row=4, column=1, sticky=W)
money = StringVar()
Entry(window, textvariable=money, justify=RIGHT).grid(row=1, column=2)
period = StringVar()
Entry(window, textvariable=period, justify=RIGHT).grid(row=2, column=2)
AIR = StringVar()
Entry(window, textvariable=AIR, justify=RIGHT).grid(row=3, column=2)
PV = StringVar()
Label(window, textvariable=PV).grid(row=4, column=2, sticky=E)
Button(window, text='계산하기', command=cal).grid(row=5, column=2, sticky=E)

window.mainloop()

```

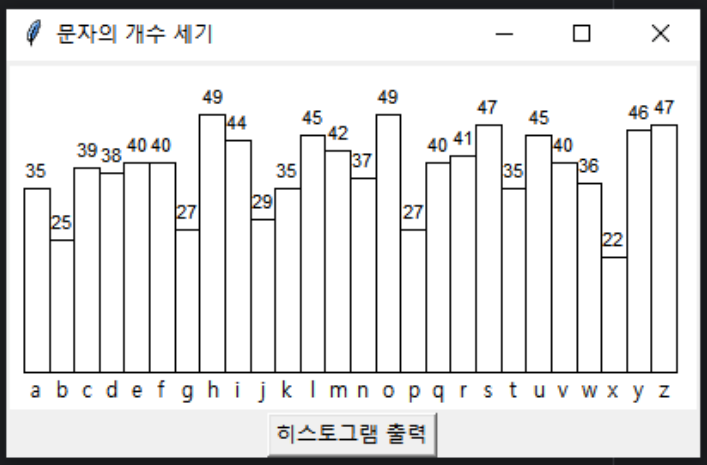
9-3.



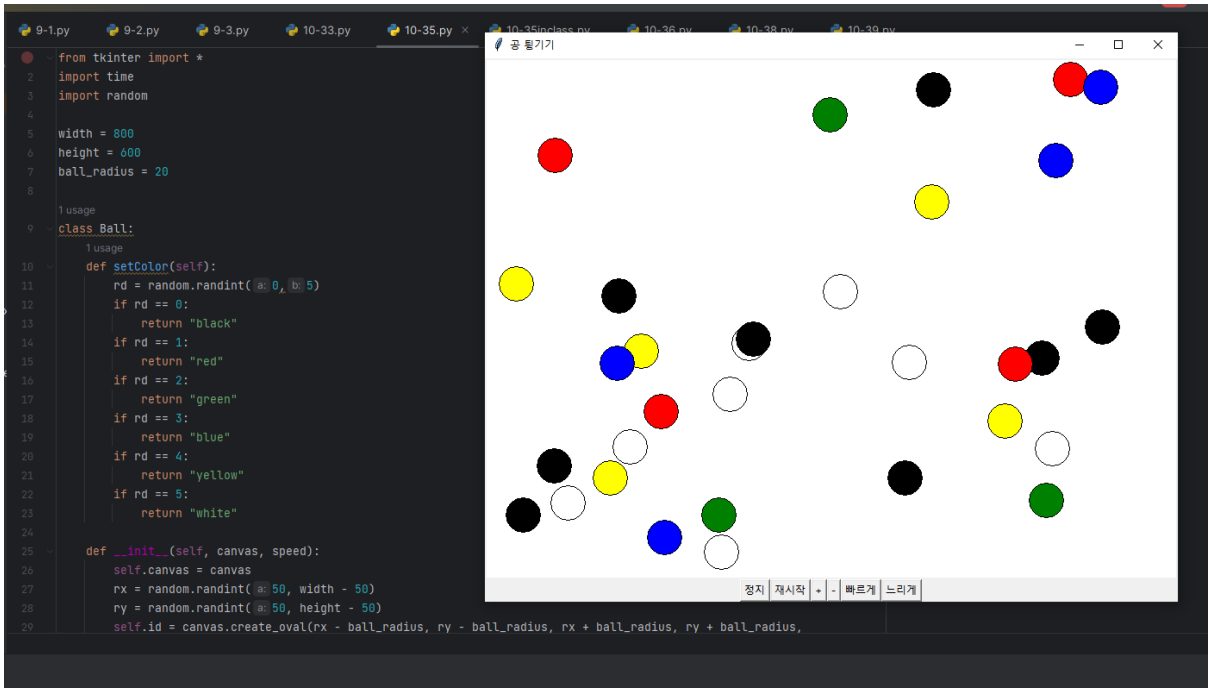

```python
# 9-3
from tkinter import *

width = 300
height = 50

1 usage
class MainGUI:
    3 usages
    def display(self):
        self.canvas.delete('shape')
        if self.v.get() == 1:
            shape = self.canvas.create_rectangle(width / 2 - width * 0.4, height / 2 - height * 0.4,
                                                 width / 2 + width * 0.4, height / 2 + height * 0.4, tags='shape')
        elif self.v.get() == 2:
            shape = self.canvas.create_oval(width / 2 - width * 0.4, height / 2 - height * 0.4,
                                            width / 2 + width * 0.4, height / 2 + height * 0.4, tags='shape')
        if self.filled.get() != 1:
            self.canvas.itemconfig(shape, fill='')
        else:
            self.canvas.itemconfig(shape, fill='blue')

    def __init__(self):
        self.window = Tk()
        self.window.title("라디오 버튼과 체크 버튼")
        self.canvas = Canvas(self.window, bg='white', width=width, height=height)
        self.canvas.create_rectangle(width / 2 - width * 0.4, height / 2 - height * 0.4,
                                     width / 2 + width * 0.4, height / 2 + height * 0.4, tags='shape')
        self.canvas.pack()
        frame = Frame(self.window)
        frame.pack()
        self.v = IntVar()
        Radiobutton(frame, text='직사각형', variable=self.v, value=1, command=self.display).pack(side=LEFT)
        Radiobutton(frame, text='타원', variable=self.v, value=2, command=self.display).pack(side=LEFT)
        self.filled = IntVar()
        Checkbutton(frame, text='채우기', variable=self.filled, command=self.display).pack(side=LEFT)

ML = MainGUI()
ML.window.mainloop()
```

10-33.

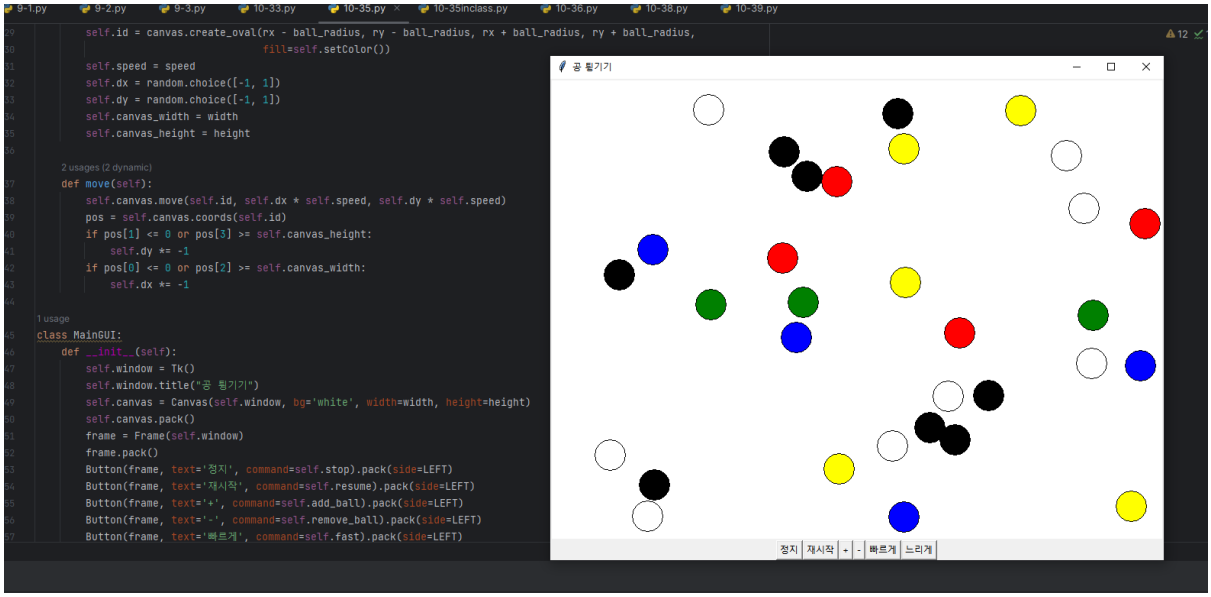
```python
from tkinter import *
import random

1 usage
class MainGUI:
    1 usage
    def random_words(self):
        self.hist = [0 for _ in range(26)]
        for i in range(1000):
            rnd = random.randint( a: 0, b: 25)
            self.hist[rnd] += 1

    1 usage
    def show(self):
        self.random_words()
        self.canvas.delete("all")
        width = int(self.canvas['width'])
        height = int(self.canvas['height'])
        maxCounts = max(self.hist)
        heightBar = height * 0.75
        widthBar = width - 20
        for i in range(26):
            x0 = 10 + i * widthBar / 26
            y0 = height - 20
            x1 = 10 + (i + 1) * widthBar / 26
            y1 = height - heightBar * self.hist[i] / maxCounts - 20
            self.canvas.create_rectangle(x0, y0, x1, y1)
            x_text = (x0 + x1) / 2
            y_text = y1 - 10
            self.canvas.create_text(x_text, y_text, text=str(self.hist[i]), font=("Helvetica", 8))
            self.canvas.create_text(i * widthBar / 26 + 10 + 0.5 * widthBar / 26, height - 10, text=chr(i + ord('a')))

    def __init__(self):
        self.window = Tk()
        self.window.title("문자의 개수 세기")
        frame1 = Frame(self.window)
        frame1.pack()
        self.canvas = Canvas(self.window, bg='white', width=400, height=200)
        self.canvas.pack()
        Button(self.window, text='히스토그램 출력', command=self.show).pack()

ML = MainGUI()
ML.window.mainloop()
```



10-35.

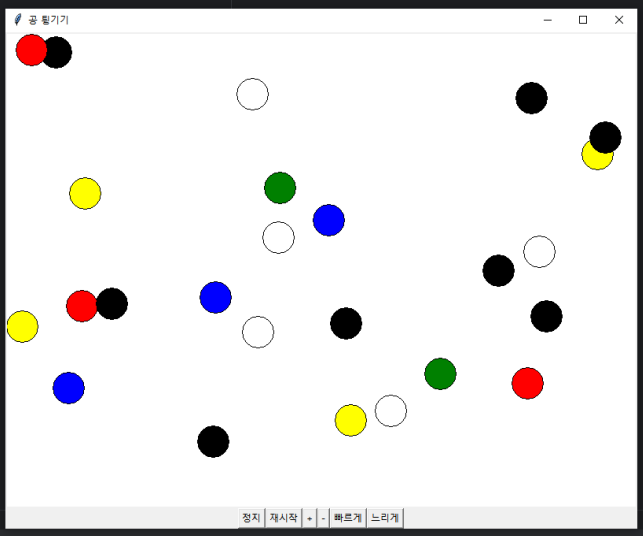



```python
from tkinter import *
import time
import random

width = 800
height = 600
ball_radius = 20

1 usage
class Ball:
    1 usage
    def setColor(self):
        rd = random.randint( a: 0, b: 5)
        if rd == 0:
            return "black"
        if rd == 1:
            return "red"
        if rd == 2:
            return "green"
        if rd == 3:
            return "blue"
        if rd == 4:
            return "yellow"
        if rd == 5:
            return "white"

    def __init__(self, canvas, speed):
        self.canvas = canvas
        rx = random.randint( a: 50, width - 50)
        ry = random.randint( a: 50, height - 50)
        self.id = canvas.create_oval(rx - ball_radius, ry - ball_radius, rx + ball_radius, ry + ball_radius,
```



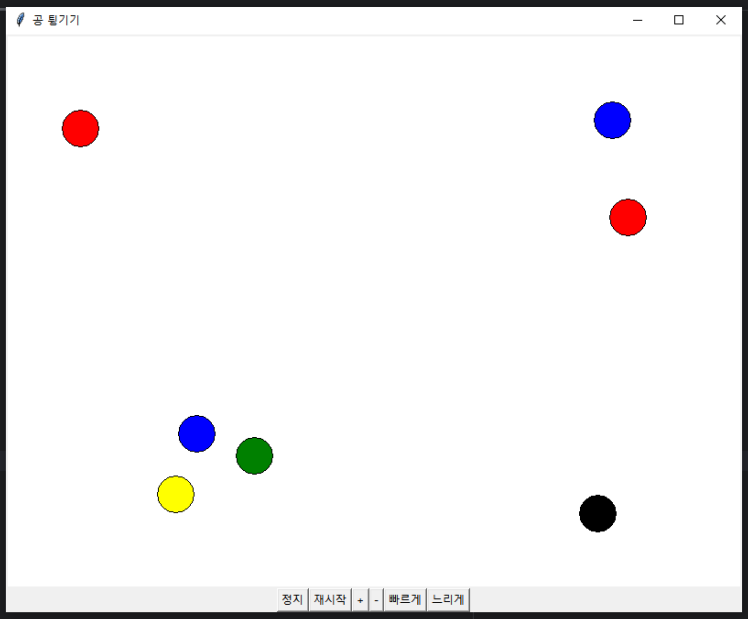

```python
        self.id = canvas.create_oval(rx - ball_radius, ry - ball_radius, rx + ball_radius, ry + ball_radius,
                                     fill=self.setColor())
        self.speed = speed
        self.dx = random.choice([-1, 1])
        self.dy = random.choice([-1, 1])
        self.canvas_width = width
        self.canvas_height = height

    2 usages (2 dynamic)
    def move(self):
        self.canvas.move(self.id, self.dx * self.speed, self.dy * self.speed)
        pos = self.canvas.coords(self.id)
        if pos[1] <= 0 or pos[3] >= self.canvas_height:
            self.dy *= -1
        if pos[0] <= 0 or pos[2] >= self.canvas_width:
            self.dx *= -1

1 usage
class MainGUI:
    def __init__(self):
        self.window = Tk()
        self.window.title("공 튕기기")
        self.canvas = Canvas(self.window, bg='white', width=width, height=height)
        self.canvas.pack()
        frame = Frame(self.window)
        frame.pack()
        Button(frame, text='정지', command=self.stop).pack(side=LEFT)
        Button(frame, text='재시작', command=self.resume).pack(side=LEFT)
        Button(frame, text='+', command=self.add_ball).pack(side=LEFT)
        Button(frame, text='-', command=self.remove_ball).pack(side=LEFT)
        Button(frame, text='빠르게', command=self.fast).pack(side=LEFT)
```

```
        Button(frame, text='빠르게', command=self.fast).pack(side=LEFT)
        Button(frame, text='느리게', command=self.slow).pack(side=LEFT)
        self.gamespeed = 1
        self.balls = []
        self.running = True
        self.move_balls()

    2 usages
    def move_balls(self):
        while self.running:
            for ball in self.balls:
                ball.move()
            self.window.update()
            time.sleep(0.01)
    1 usage
    def stop(self):
        self.running = False
    1 usage
    def resume(self):
        self.running = True
        self.move_balls()
    1 usage
    def add_ball(self):
        ball = Ball(self.canvas, self.gamespeed)
        self.balls.append(ball)
    1 usage
    def remove_ball(self):
        if self.balls:
            ball = self.balls.pop()
            self.canvas.delete(ball.id)
    1 usage
```



9-1.py | 9-2.py | 9-3.py | 10-33.py | 10-35.py | 10-35inclass.py | 10-36.py | 10-38.py | 10-39.py

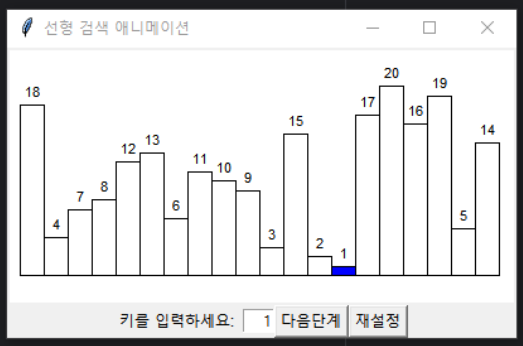
```
    def resume(self):
        self.running = True
        self.move_balls()
    1 usage
    def add_ball(self):
        ball = Ball(self.canvas, self.gamespeed)
        self.balls.append(ball)
    1 usage
    def remove_ball(self):
        if self.balls:
            ball = self.balls.pop()
            self.canvas.delete(ball.id)
    1 usage
    def fast(self):
        self.gamespeed += 0.5
        for ball in self.balls:
            ball.speed = self.gamespeed
    1 usage
    def slow(self):
        self.gamespeed -= 0.5
        for ball in self.balls:
            ball.speed = self.gamespeed

ML = MainGUI()
ML.window.mainloop()
```

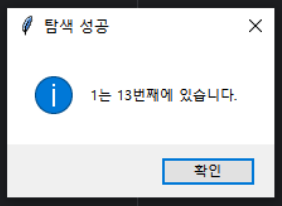


10-36.

```python
from tkinter import *
import random
from tkinter import messagebox

1 usage
class MainGUI:

    2 usages
    def draw(self):
        length = len(self.nums)
        self.canvas.delete("all")
        width = int(self.canvas['width'])
        height = int(self.canvas['height'])
        maxCounts = max(self.nums)
        heightBar = height * 0.75
        widthBar = width - 20
        for i in range(length):
            x0 = 10 + i * widthBar / length
            y0 = height - 20
            x1 = 10 + (i + 1) * widthBar / length
            y1 = height - heightBar * self.nums[i] / maxCounts - 20
            if i == self.current:
                self.canvas.create_rectangle(x0, y0, x1, y1, fill='blue')
            else:
                self.canvas.create_rectangle(x0, y0, x1, y1)
            x_text = (x0 + x1) / 2
            y_text = y1 - 10
            self.canvas.create_text(x_text, y_text, text=str(self.nums[i]), font=("Helvetica", 8))

    2 usages
    def reset(self):
        self.nums = [i for i in range(1, 21)]
        random.shuffle(self.nums)
        self.current = -1
        self.draw()


    def next(self):
        key = int(self.key.get())
        if key == self.nums[self.current]:
```

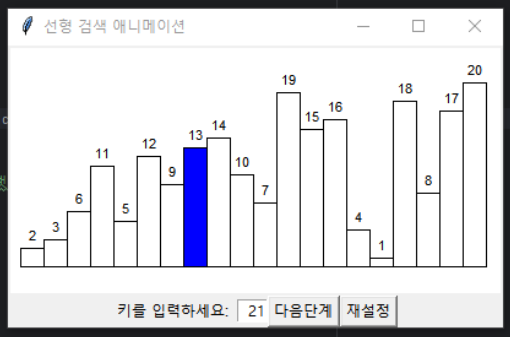
MainGUI › next() › if key == self.nums[self.curren...

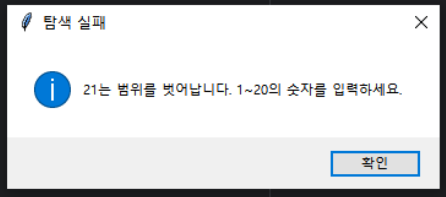




9-1.py 9-2.py 9-3.py 10-33.py 10-35.py 10-35inclass.py 10-36.py 10-38.py

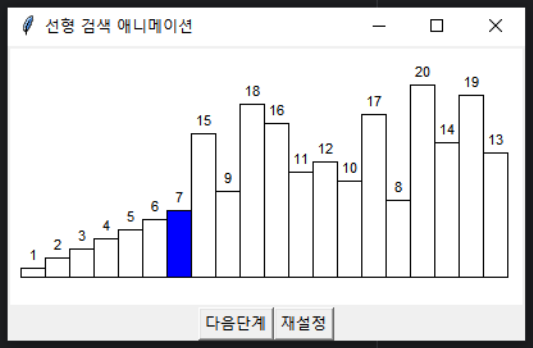
```python
        self.current = -1
        self.draw()


    def next(self):
        key = int(self.key.get())
        if key == self.nums[self.current]:
            messagebox.showinfo( title: "탐색 성공",  message: f" {key}는 {self.c
            return
        if key > 20 or key < 1:
            messagebox.showinfo( title: "탐색 실패",  message: f"{key}는 범위를 벗
            return

        self.current += 1
        self.draw()


    def __init__(self):
        self.window = Tk()
        self.window.title("선형 검색 애니메이션")
        self.canvas = Canvas(self.window, bg='white', width=400, height=200)
        self.canvas.pack()
        frame = Frame(self.window)
        frame.pack()
        Label(frame, text='키를 입력하세요: ').pack(side=LEFT)
        self.key = IntVar()
        Entry(frame, textvariable=self.key, justify=RIGHT, width=3).pack(side=LEFT)
        Button(frame, text='다음단계', command=self.next).pack(side=LEFT)
        Button(frame, text='재설정', command=self.reset).pack(side=LEFT)
        self.current = -1
        self.reset()


ML = MainGUI()
ML.window.mainloop()
```



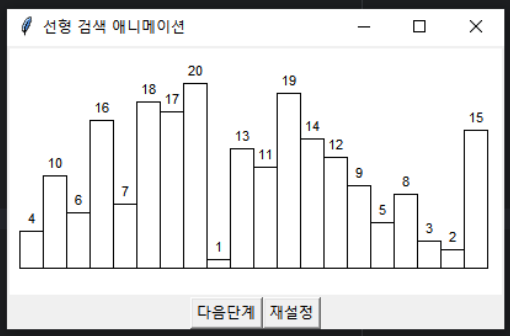


10-38.

```
from tkinter import *
import random
from tkinter import messagebox

1 usage
class MainGUI:

    2 usages
    def draw(self):
        length = len(self.nums)
        self.canvas.delete("all")
        width = int(self.canvas['width'])
        height = int(self.canvas['height'])
        maxCounts = max(self.nums)
        heightBar = height * 0.75
        widthBar = width - 20
        for i in range(length):
            x0 = 10 + i * widthBar / length
            y0 = height - 20
            x1 = 10 + (i + 1) * widthBar / length
            y1 = height - heightBar * self.nums[i] / maxCounts - 20
            if i == self.current:
                self.canvas.create_rectangle(x0, y0, x1, y1, fill='blue')
            else:
                self.canvas.create_rectangle(x0, y0, x1, y1)
            x_text = (x0 + x1) / 2
            y_text = y1 - 10
            self.canvas.create_text(x_text, y_text, text=str(self.nums[i]), font=("Helvetica", 8))

    2 usages
    def reset(self):
        self.nums = [i for i in range(1, 21)]
        random.shuffle(self.nums)
        self.current = -1
        self.draw()


    def next(self):
        self.current += 1
        for i in range(len(self.nums)):
ainGUI › next() › for i in range(len(self.nums))
```

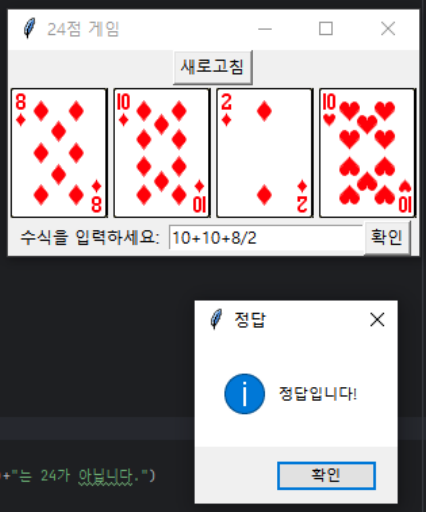


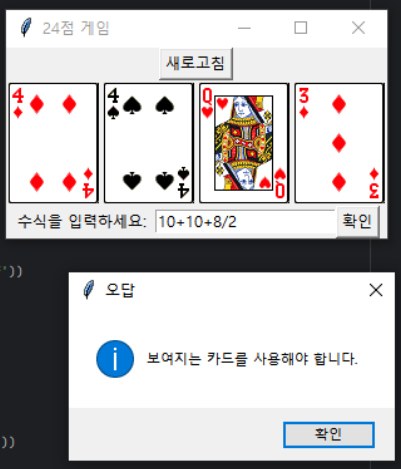

```
            self.canvas.create_text(x_text, y_text, text=str(self.nums[i]), font=("Helvetica", 8))

    2 usages
    def reset(self):
        self.nums = [i for i in range(1, 21)]
        random.shuffle(self.nums)
        self.current = -1
        self.draw()


    def next(self):
        self.current += 1
        for i in range(len(self.nums)):
            if self.current + 1 == self.nums[i]:
                self.nums.insert(self.current, self.nums.pop(i))
        self.draw()


    def __init__(self):
        self.window = Tk()
        self.window.title("선형 검색 애니메이션")
        self.canvas = Canvas(self.window, bg='white', width=400, height=200)
        self.canvas.pack()
        frame = Frame(self.window)
        frame.pack()
        Button(frame, text='다음단계', command=self.next).pack(side=LEFT)
        Button(frame, text='재설정', command=self.reset).pack(side=LEFT)
        self.current = -1
        self.reset()


ML = MainGUI()
ML.window.mainloop()
```



10-39.

```
from tkinter import *
import tkinter.simpledialog
import random
1 usage
class MainGUI:
    def check(self):
        fourCards=[]
        for i in range(4):
            fourCards.append(self.index[i] % 13 + 1)
        fourCards.sort()
        ex = self.answer.get()
        ex = ex.replace( _old: '+', _new: ' ')
        ex = ex.replace( _old: '-', _new: ' ')
        ex = ex.replace( _old: '*', _new: ' ')
        ex = ex.replace( _old: '/', _new: ' ')
        ex = ex.replace( _old: '(', _new: ' ')
        ex = ex.replace( _old: ')', _new: ' ')
        numbers = [eval(i) for i in ex.split()]
        numbers.sort()
        if fourCards == numbers:
            if eval(self.answer.get()) == 24:
                tkinter.messagebox.showinfo("정답", "정답입니다!")
            else:
                tkinter.messagebox.showinfo("오답", self.answer.get()+"는 24가 아닙니다.")
        else:
            tkinter.messagebox.showinfo("오답", "보여지는 카드를 사용해야 합니다.")



    2 usages
    def reset(self):
        random.shuffle(self.index)
        for i in range(4):
            self.labelList[i]['image'] = self.imageList[self.index[i]]

    def __init__(self):
        window = Tk()
        window.title('24점 게임')
        self.index = [x for x in range(52)]
        self.imageList = []
```




```
    2 usages
    def reset(self):
        random.shuffle(self.index)
        for i in range(4):
            self.labelList[i]['image'] = self.imageList[self.index[i]]

    def __init__(self):
        window = Tk()
        window.title('24점 게임')
        self.index = [x for x in range(52)]
        self.imageList = []
        for i in range(1, 53):
            self.imageList.append(PhotoImage(file='image/'+str(i)+'.gif'))
        frame1 = Frame(window)
        frame1.pack()
        Button(frame1, text='새로고침',command=self.reset).pack()
        frame2 = Frame(window)
        frame2.pack()
        self.labelList = []
        for i in range(4):
            self.labelList.append(Label(frame2, image=self.imageList[i]))
            self.labelList[i].pack(side=LEFT)
        self.reset()
        frame3 = Frame(window)
        frame3.pack()
        Label(frame3,text='수식을 입력하세요: ').pack(side=LEFT)
        self.answer = StringVar()
        Entry(frame3, textvariable=self.answer).pack(side=LEFT)
        Button(frame3,text='확인',command=self.check).pack(side=LEFT)

        window.mainloop()

MainGUI()
```