내 맘대로 가계부

2019311896 구성현
2019311305 조윤영
2019312864 임채림

프로그램 소개

내 맘대로 가계부는 데이터 백업, 계산기, 통계 기능뿐만 아니라 최소 잔액과 자신만의 캐릭터를 설정해 돈을 현명하게 관리할 수 있다.

배경 및 목적

배경

기존의 가계부는 글 형태로만 표현되어서 지루하다는 느낌을 받았다. 그래서 거북의 모듈을 이용해서 이모지로 표현되는 가계부를 만들고 싶었다.

목적

효율적인 돈 관리를 위해 기존의 가계부와 차별화된 나만의 이모지 기능을 넣음으로써 시각화된 정보로 돈 관리에 경각심을 준다.

프로그램 순서

새로 만들기

가계부 기본설정

불러오기

메뉴 선택

1. 가계부 이름 변경
2. 잔액 변경
3. 최소 기준 변경
4. 이모지 설정
5. 수입 기록
6. 지출 기록
7. 월말 결산
8. 나가기

모듈 설명

head.py

```python
#얼굴_별
def star(head_color):
    t.pensize(1)
    t.color(head_color)
    move(280, 60, t)
    t.left(36)
    t.begin_fill()
    for i in range(5):
        t.left(144)
        t.forward(550)
    t.end_fill()
    t.right(36)
    return
```

```python
#코_세모
def triangle(nose_color):
    t.color('black', nose_color)
    move(-20, -50, t)
    t.begin_fill()
    for i in range(3):
        t.forward(40)
        t.left(120)
    t.end_fill()
    return
```

eye.py

```python
#눈_원
def circle(eye_color):
    t.begin_fill()
    t.color('black', eye_color)
    move(-80, 0, t)
    t.circle(40)
    t.end_fill()
    move(80, 0, t)
    t.begin_fill()
    t.circle(40)
    t.end_fill()
    return
```

```python
def notbad():
    t.pensize(10)
    # 눈썹
    move(-125, 80, t)
    t.forward(80)

    move(40, 80, t)
    t.forward(80)

    # 입
    move(0, -160, t)
    t.circle(25)
    return
```

emoji.py

```python
def drawing_emoji(items, money, minimum):
    import head as h
    import eye as e
    import nose as n
    import mood as m
    h.head(items[3], items[4])
    e.eye(items[5], items[6])
    n.nose(items[7], items[8])
    m.mood(money, minimum)
```

```python
def making_emoji(file_name):
    f = open(file_name+'.txt', 'r')
    items = f.read().split('₩n')
    f.close()

    head_dic = {'a':'원', 'b':'네모', 'c':'세모','d':'별', 'e':'하트'}
    print('얼굴 모양 =', head_dic)
    head_shape = input('얼굴 모양을 선택하세요: ')
    h_color = input('얼굴 색깔을 입력하세요(영어로): ')
    print()

    eye_dic = {'a':'원', 'b':'네모'}
    print('눈 모양 =', eye_dic)
    eye_shape = input('눈 모양을 선택하세요: ')
    e_color = input('눈 색깔을 입력하세요(영어로): ')
    print()

    nose_dic = {'a':'원', 'b':'세모'}
    print('코 모양 =', nose_dic)
    nose_shape = input('코 모양을 선택하세요: ')
    n_color = input('코 색깔을 입력하세요(영어로): ')
    print()

    f = open(file_name+'.txt', 'r')
    items = f.read().split('₩n')
    f.close()

    items[3] = head_shape
    items[4] = h_color
    items[5] = eye_shape
    items[6] = e_color
    items[7] = nose_shape
    items[8] = n_color

    f = open(file_name+'.txt', 'w')
    for i in range(len(items)-1):
        record = items[i]+'₩n'
        f.write(record)
    f.close()
    drawing_emoji(items, 1, 0)
```

graph.py

```python
def piePlt(total, data, colors):
    if total == 0:
        print('기록이 없습니다.')
        return
    else:
        categories = list(data.keys())
        expences = list(data.values())

        percentages = []
        for i in range(len(data)):
            percentages.append(expences[i]/total)

        t.hideturtle()
        t.penup()
        t.goto(0, -30)
        t.pendown()

        t.left(90)
        for j in range(len(percentages)):
            angle = percentages[j]*360
            if angle == 0:
                continue
            t.color('white', colors[j])
            t.pensize(2)
            t.speed(10)
            t.begin_fill() # 도형 그리기
            t.forward(180)
            t.left(90)
            t.circle(180, angle)
            t.left(90)
            t.forward(180)
            t.end_fill()

            t.penup()
            t.right(180 + angle/2)
            t.forward(90)
            t.pendown()

            text = '%s(%d%%)'%(categories[j],percentages[j]*100)
            t.write(text, False, 'center',('함초롬바탕',15,'bold'))
            t.penup()
            t.right(180)
            t.forward(90)
            t.right(180-angle/2)
            t.pendown()
        t.speed(4)
```

각 모듈은 얼굴 모양, 눈 모양, 코 모양, 눈썹과 입모양을 그리는 함수로 구성되어 있다.

사용자가 원하는 이모지 설정을 입력 받는 함수와 출력하는 함수로 구성되어 있다.

월말결산 시, 수입내역과 지출내역을 원그래프로 그려주는 함수로 구성되어 있다.

새로 만들기 및 불러오기

새로 만들기

```
print('+내맘대로 가계부+')
print('"환영합니다! +내맘대로 가계부+는 데이터 백업, 계산기, 통계 기능뿐만 아니라')
print('최소 잔액과 자신만의 캐릭터를 설정해 돈을 현명하게 관리할 수 있습니다."')
print()
choice1 = input('새로 만들려면 A, 불러오려면 B: ')
if choice1 == 'A':
    file_name = input('가계부 이름을 입력하세요: ')
    f=open(file_name+'.txt', 'w')
    f.write(file_name+'₩n')
    money = input('잔액을 입력하세요: ')
    f.write(money+'₩n')
    minimum = input('살아남기 위한 최소 잔액을 입력하세요: ')
    f.write(minimum+'₩n')
    for i in range(6):
        f.write('₩n')
    f.close()
    print()
    print('자신만의 캐릭터를 만들어봅시다')
    import emoji
    emoji.making_emoji(file_name)
    print('당신의 캐릭터가 완성되었습니다!')
```

↓

```
+내맘대로 가계부+
"환영합니다! +내맘대로 가계부+는 데이터 백업, 계산기, 통계 기능뿐만 아니라
최소 잔액과 자신만의 캐릭터를 설정해 돈을 현명하게 관리할 수 있습니다."

새로 만들려면 A, 불러오려면 B: A
가계부 이름을 입력하세요: 문알못+가계부
잔액을 입력하세요: 100000
살아남기 위한 최소 잔액을 입력하세요: 30000

자신만의 캐릭터를 만들어봅시다
얼굴 모양 = {'a': '원', 'b': '네모', 'c': '세모', 'd': '별', 'e': '하트'}
얼굴 모양을 선택하세요: d
얼굴 색깔을 입력하세요(영어로): gold

눈 모양 = {'a': '원', 'b': '네모'}
눈 모양을 선택하세요: a
눈 색깔을 입력하세요(영어로): green

코 모양 = {'a': '원', 'b': '세모'}
코 모양을 선택하세요: a
코 색깔을 입력하세요(영어로): red

당신의 캐릭터가 완성되었습니다!
```

가계부 이름과 잔액, 살아남기 위한 최소 잔액의 기준을 정하고 나만의 이모지를 만든다.

불러오기

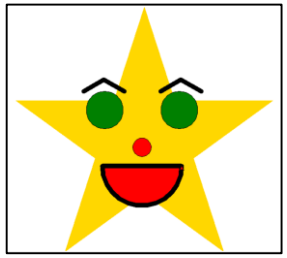
```
if choice1 == 'B':
    file_name = input('불러올 가계부 이름을 입력하세요: ')
    f=open(file_name+'.txt', 'r')
    items = f.read().split('₩n')
    print('잔액은',items[1], '입니다')
    f.close()
```

↓

```
+내맘대로 가계부+
"환영합니다! +내맘대로 가계부+는 데이터 백업, 계산기, 통계 기능뿐만 아니라
최소 잔액과 자신만의 캐릭터를 설정해 돈을 현명하게 관리할 수 있습니다."

새로 만들려면 A, 불러오려면 B: B
불러올 가계부 이름을 입력하세요: 문알못+가계부
잔액은 100000 입니다
```

기존의 가계부 이름을 입력해서 파일을 불러온다.

설정된 이모지와 파일 입력 예시



```
문알못+가계부
100000
30000
d
gold
a
green
a
red
```

가계부 이름, 잔액, 최소 기준 변경

```
#가계부 이름 변경
    if m_choice == 1:
        print()
        print('<1:가계부 이름 변경>을 선택하셨습니다.')
        f = open(file_name+'.txt', 'r')
        items = f.read().split('₩n')
        f.close()
        print('기존 이름은',items[0],'입니다.')
        c_name = input('이름을 변경하시겠습니까 (Y/N): ')

        if c_name == 'Y':
            name = input('변경할 이름을 입력하세요: ')
            items[0] = name
            file_name = name
            f = open(file_name+'.txt', 'w')
            for i in items:
                f.write(i+'₩n')
            f.close()
        else:
            continue
```

```
#잔액 변경
    if m_choice == 2:
        print()
        print('<2:잔액 변경>을 선택하셨습니다.')
        f = open(file_name+'.txt', 'r')
        items = f.read().split('₩n')
        f.close()
        print('기존 잔액은',items[1],'입니다.')
        c_money = input('잔액을 변경하시겠습니까? (Y/N): ')

        if c_money == 'Y':
            money = input('변경할 잔액을 입력하세요: ')
            items[1] = money
            f = open(file_name+'.txt', 'w')
            for i in items:
                f.write(i+'₩n')
            f.close()
```

```
#최소기준변경
    if m_choice == 3:
        print()
        print('<3:최소 기준 변경>을 선택하셨습니다.')
        f = open(file_name+'.txt', 'r')
        items = f.read().split('₩n')
        f.close()
        print('기존 살아남기 위한 최소 잔액은',items[2],'입니다.')
        c_minimum = input('살아남기 위한 최소 잔액을 변경하시겠습니까? (Y/N): ')
        if c_minimum == 'Y':
            minimum = input('변경할 최소 잔액을 입력하세요: ')
            items[2] = minimum
            f = open(file_name+'.txt', 'w')
            for i in items:
                f.write(i+'₩n')
            f.close()
```

가계부 이름 변경

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지
설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 1

<1:가계부 이름 변경>을 선택하셨습니다.
기존 이름은 문알못+가계부 입니다.
이름을 변경하시겠습니까 (Y/N): Y
변경할 이름을 입력하세요: 문잘알+가계부
```

잔액 변경

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지
설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 2

<2:잔액 변경>을 선택하셨습니다.
기존 잔액은 100000 입니다.
잔액을 변경하시겠습니까? (Y/N): Y
변경할 잔액을 입력하세요: 120000
```

최소기준 변경

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지
설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 3

<3:최소 기준 변경>을 선택하셨습니다.
기존 살아남기 위한 최소 잔액은 30000 입니다.
살아남기 위한 최소 잔액을 변경하시겠습니까? (Y/N): Y
변경할 최소 잔액을 입력하세요: 20000
```

변경여부를 입력 받아 if-else 조건문으로 실행한다. 변경할 경우 가계부 파일을 리스트로 읽은 후 리스트 요소를 변경하고 파일을 새로 쓴다.

이모지 설정, 수입/지출 기록

```
#이모지 설정
    if m_choice == 4:
        print()
        print('<4:이모지 설정>을 선택하셨습니다.')
        import emoji
        emoji.clear()
        emoji.making_emoji(file_name)
        print('당신의 새로운 캐릭터입니다!')
```

emoji.py 모듈을 가져와서 이모지를 만드는 함수를 실행하고 출력한다.

```
# 수입기록
    if m_choice == 5:
        print()
        print('<5:수입 기록>을 선택하셨습니다.')
        month, date = input('오늘의 날짜는 무엇입니까? (월/일): ').split('/')
        income_dic = {'a':'알바비', 'b':'용돈', 'c':'기타'}
        print(income_dic)
        part = input('분류를 선택하세요: ')
        income = input('금액은 얼마인가요?: ')
        income_record = '수입 %s %s %s %s %s₩n'%(month, date, income_dic[part], income)
        f = open(file_name+'.txt', 'r')
        items = f.read().split('₩n')
        f.close()
        items.append(income_record)
        money = int(items[1])+int(income)
        items[1] = str(money)
        minimum = int(items[2])
        f = open(file_name+'.txt', 'w')
        for i in items:
            f.write(i+'₩n')
        f.close()
        print('살아남기 위한 최소 잔액:',items[2]+'원')
        print('잔액은', str(money)+'원 입니다')
        minimum = int(items[2])
        import emoji
        emoji.clear()
        emoji.drawing_emoji(items, int(money), int(minimum))
        emoji.condition(int(money), int(minimum))
```

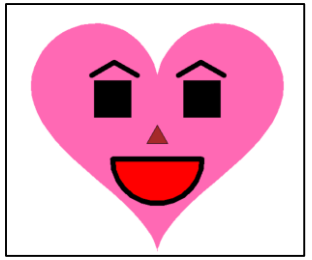
사용자에게 날짜, 분류, 금액을 입력 받아 파일에 덧대어 쓴다.

지출기록도 똑같은 알고리즘이다.

이모지 설정

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지 설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 4

<4:이모지 설정>을 선택하셨습니다.
얼굴 모양 = {'a': '원', 'b': '네모', 'c': '세모', 'd': '별', 'e': '하트'}
얼굴 모양을 선택하세요: e
얼굴 색깔을 입력하세요(영어로): hotpink

눈 모양 = {'a': '원', 'b': '네모'}
눈 모양을 선택하세요: b
눈 색깔을 입력하세요(영어로): black

코 모양 = {'a': '원', 'b': '세모'}
코 모양을 선택하세요: b
코 색깔을 입력하세요(영어로): brown

당신의 새로운 캐릭터입니다!
```

변경된 이모지



수입/지출 기록

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지 설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 5

<5:수입 기록>을 선택하셨습니다.
오늘의 날짜는 무엇입니까? (월/일): 11/28
{'a': '알바비', 'b': '용돈', 'c': '기타'}
분류를 선택하세요: b
금액은 얼마인가요?: 30000
살아남기 위한 최소 잔액: 20000원
잔액은 150000원 입니다
```

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지 설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 6

<6:지출 기록>을 선택하셨습니다.
오늘의 날짜는 무엇입니까? (월/일):11/28
{'a': '생활비', 'b': '식비', 'c': '패션/미용비', 'd': '여가비', 'e': '교통비', 'f': '기타'}
분류를 선택하세요: d
금액은 얼마입니까?: 50000
살아남기 위한 최소 잔액: 20000원
잔액은 100000원 입니다
```

파일 저장 예시:

수입 11 28 용돈 30000

지출 11 28 여가비 50000

월말 결산

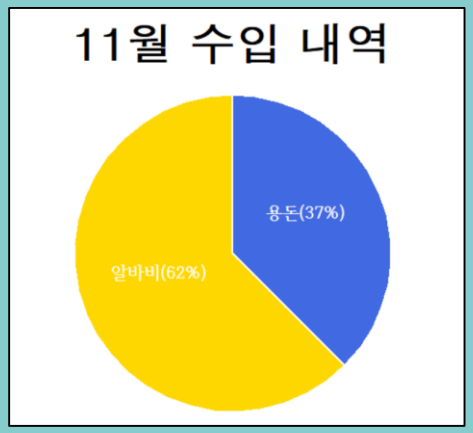
```
# 결산
    if m_choice == 7:
        print()
        print('<7:월말결산>을 선택하셨습니다.')
        f = open(file_name+'.txt', 'r')
        items = f.read().split('₩n')
        f.close()
        for i in range(9, len(items)):
            lst = items[i].split(' ')
            items[i] = lst
        print('현재 잔액은',items[1]+'원 입니다.')
        check_month = input('결산을 원하는 Month를 입력하세요: ')
        flow = input('어떤 것을 보시겠습니까? (수입/지출): ' )

        if flow == '수입':
            total_income = 0
            wage = 0
            cash = 0
            etc = 0
            for i in range(9, len(items)):
                if items[i][0] == '수입' and items[i][1] == check_month:
                    total_income = total_income+int(items[i][4])
                    if items[i][3] == '알바비':
                        wage += int(items[i][4])
                    if items[i][3] == '용돈':
                        cash += int(items[i][4])
                    if items[i][3] == '기타':
                        etc += int(items[i][4])
            print()
            print(check_month+'월의 총 수입은 %d원입니다.'%total_income)
            print('알바비: %d원'%wage)
            print('용돈: %d원'%cash)
            print('기타: %d원'%etc)

            income_data = {'알바비':wage, '용돈':cash, '기타':etc}
            colors = ['gold', 'royal blue', 'tomato']
            import emoji
            emoji.clear()
            import graph
            graph.title(str(check_month), flow)
            graph.pieplt(total_income, income_data, colors)
```

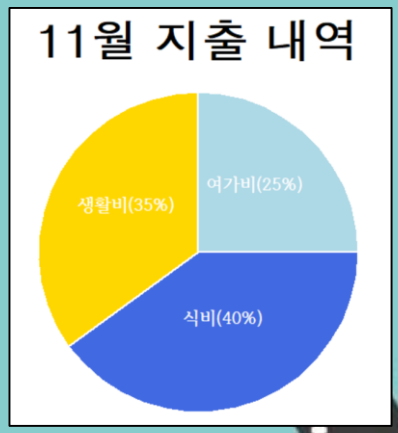
```
else:
    total_spending = 0
    living = 0
    food = 0
    beauty = 0
    leisure = 0
    transportation = 0
    etc = 0
    for i in range(9, len(items)):
        if items[i][0] == '지출' and items[i][1] == check_month:
            total_spending = total_spending+int(items[i][4])
            if items[i][3] == '생활비':
                living += int(items[i][4])
            elif items[i][3] == '식비':
                food += int(items[i][4])
            elif items[i][3] == '패션/미용비':
                beauty += int(items[i][4])
            elif items[i][3] == '여가비':
                leisure += int(items[i][4])
            elif items[i][3] == '교통비':
                transportation += int(items[i][4])
            elif items[i][3] == '기타':
                etc += int(items[i][4])
    print()
    print(check_month+'월의 총 지출은 %d원입니다.'%total_spending)
    print('생활비: %d원'%living)
    print('식비: %d원'%food)
    print('패션/미용비: %d원'%beauty)
    print('여가비: %d원'%leisure)
    print('교통비: %d원'%transportation)
    print('기타: %d원'%etc)

    spending_data = {'생활비':living, '식비':food, '패션/미용비':beauty, '여가비':leisure, '교통비':transportation, '기타':etc}
    colors = ['gold', 'royal blue', 'tomato', 'lightblue', 'lightgreen', 'gray']
    import emoji
    emoji.clear()
    import graph
    graph.title(str(check_month), flow)
    graph.pieplt(total_spending, spending_data, colors)
```

월말 결산

```
메뉴 = {1: '가계부 이름 변경', 2: '잔액 변경', 3: '최소 기준 변경', 4: '이모지 설정', 5: '수입 기록', 6: '지출 기록', 7: '월말결산', 8: '나가기'}
메뉴를 선택하세요: 7

<7:월말결산>을 선택하셨습니다.
현재 잔액은 0원 입니다.
결산을 원하는 Month를 입력하세요: 11
어떤 것을 보시겠습니까? (수입/지출): 지출

11월의 총 지출은 200000원입니다.
생활비: 70000원
식비: 80000원
패션/미용비: 0원
여가비: 50000원
교통비: 0원
기타: 0원
```

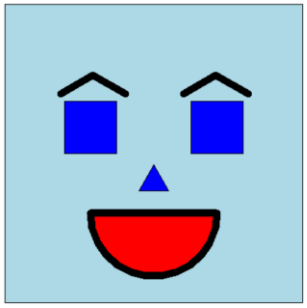
파일을 리스트로 읽고, 순차 탐색 알고리즘을 이용해 분류별로 금액을 찾아 더한다.
이후 graph.py 모듈을 가져와 원 그래프를 그리는 함수를 실행한다.





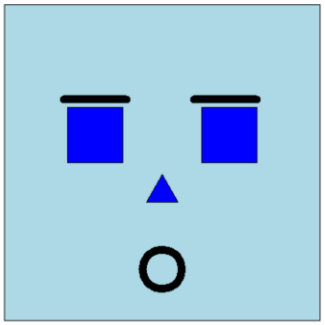
프로그램의 차별화된 특성

여유롭네요! 잘하고 있어요~



잔액 > 최소기준일 때

간당간당하네요, 조금만 아껴써요!



잔액 = 최소기준일 때

돈이 많이 부족해요! 밥은 먹고 다니는 거죠?

잔액 < 최소기준일 때

- 잔액에 따라 표정이 바뀌는 나만의 이모지가 채찍과 당근을 준다.
- 월말 결산에서 수입과 지출 비율을 원 그래프로 보여준다.
- 기존의 가계부를 불러와 계속 작성할 수 있다.

아쉬운 점

- 원그래프를 그릴 때 비율이 작으면 글씨가 잘 보이지 않는 경우가 있다.
- 'Matplotlib' 라이브러리를 설치하지 못해 원그래프 그리는 내장 모듈을 사용하지 못했다.