

기다리면 얼마나 줄꺼야?

학번: 2018066

이름: 배정훈

<https://github.com/BAEJUNGHUN2018066/How-much-will-you-give-me-if-I-wait->

1. 계산기의 목적

은행마다 달마다 넣을 수 있는 최대 금액이 다르고 기간과 이자율도 다르기 때문에 자신이 얼마나 넣을지 입력하면 기간이 끝난 후 금액이 얼마나 나오는지 더욱 손쉽게 알아볼 수 있고 은행마다 어떤 차이가 있는지 비교해 자신에게 맞는 은행 적금을 찾게 하는 것이 이 계산기의 목적이다.

계산기 활용 대상: 여러 은행 적금 중에 어디 은행 적금이 가장 나에게 적합한지 찾는 사람들

2. 계산기의 네이밍의 의미

자신이 적금에 얼마를 넣고 적금 기한이 끝날 때까지 기다렸을 때 은행마다 얼마나 주는지에 대해 손쉽게 비교해서 알아보는 계산기이기에 기다리면 얼마나 줄꺼야? 계산기라고 이름을 지었다.

3. 계산기 개발 계획

Input

은행의 종류, 매달 저축 금액, 일년 중 입금 횟수입니다.

함수

매 년 넣는 금액을 더하고 적금 기간이 끝나고 저축된 금액에서 이자율을 곱해 이자를 주는 계산을 해주기 위해 함수를 만들었습니다.

For 반복문

저축_기간과 매달 금액을 넣기 때문에 12 개월로 잡아 Future_value 에 매달 금액을 더합니다. 반복문을 통해 다 더해진 금액을 매달마다 저축할 횟수를 12 로 나누어 1 년에 몇 번 넣느냐에 따라서 받는 금액과 금액에 따른 이자가 달라질 수 있도록 future_value 값을 설정

if 조건문

제가 정한 은행인 경우와 아닌 경우를 나누고 은행일 경우 어디 은행이고 저축기간은 몇 년인지 print 되도록 하고 은행마다 최대 넣을 수 있는 금액에 맞춰 적금 가치가 print 소수점 둘째 자리까지 나올 수 있도록 코드를 짭니다.

4. 계산기 개발 과정

4.1 실제 개발 과정

<p>초기에는 일일이 다 입력 값을 받는 식으로 코드를 작성했고 Chat gpt 를 통해 적금 가치를 구하는 반복문을 작성했다.</p> <p>처음에는 input 앞에 문자형을 바꿔주는 int, float, str 등을 붙이지 않았지만 오류가 생겨 이후에 붙였다.</p>	<pre># 초기 투자 금액 초기_투자_금액 = int(input("얼마를 넣을건가요?")) # 연 이자율 (백분율 입력을 받아 소수로 변환) 연_이자율 = float(input("연 이자율을 몇 퍼센트인가요?")) / 100 # 저축 기간 저축_기간 = int(input("몇 년까지의 적금인가요?")) # 매달마다 저축할 횟수 매달마다_저축할_횟수 = int(input("1년에 몇 번 넣으실건가요?")) future_value = 0 for year in range(저축_기간): for month in range(매달마다_저축할_횟수): future_value += 초기_투자_금액 future_value *= (1 + (연_이자율 / 매달마다_저축할_횟수)) print(f"{저축_기간}년 후의 적금 가치는 {future_value:.2f}원 입니다.")</pre>
--	--

<p>위 코드를 토대로 chat gpt 를 통해 함수를 써서 코드를 짜니 기본 값들을 함수 변수로 지정해놓고 매달 저축 금액과 future value 값을 정의했다.</p> <p>반복문을 사용해 적금 가치를 구하는 것은 같고 return 을 통해 함수 값을 받을 수 있게 해놨다.</p>	<pre>def calculate_future_value(초기_투자_금액, 연_이자율, 저축_기간, 매달마다_저축할_횟수): # 매달 넣는 총 금액 매달_저축_금액 = 초기_투자_금액 * 매달마다_저축할_횟수 future_value = 0 for year in range(저축_기간): for month in range(12): future_value += 매달_저축_금액 # 매달 저축 금액을 더함 # 월 이자율 연 이자로 변경 future_value += (future_value * 연_이자율 / 12) # 연 이자 계산 return future_value 초기_투자_금액 = int(input("얼마를 넣을건가요?")) 연_이자율 = float(input("연 이자율을 몇 퍼센트원가요?")) / 100 저축_기간 = int(input("몇 년짜리의 적금원가요?")) 매달마다_저축할_횟수 = int(input("1년에 몇 번 넣으실건가요?")) future_value = calculate_future_value(초기_투자_금액, 연_이자율, 저축_기간, 매달마다_저축할_횟수) print(f"{저축_기간}년 후의 적금 가치는 {future_value:.2f}원 입니다.")</pre>
<p>다음으로 은행 별로 비교를 할 수 있게 하기 위해 조건문을 통해 각 은행에 맞는 연 이자율을 지정해놓고 '은행' 정보가 맞지 않을 경우 메시지를 보내고 프로그램을 종료하도록 exit()을 작성했다.</p>	<pre>초기_투자_금액 = int(input("얼마를 넣을건가요?")) 은행 = input("은행을 선택하세요 (신한, 하나, 농협, 기업):") if 은행 == "신한": 연_이자율 = 0.055 # 5.5% 연 이자율 elif 은행 == "하나": 연_이자율 = 0.045 # 4.5% 연 이자율 elif 은행 == "농협": 연_이자율 = 0.06 # 6% 연 이자율 elif 은행 == "기업": 연_이자율 = 0.05 # 5% 연 이자율 else: print("지원하지 않는 은행입니다.") exit()</pre>
<p>이후 은행마다 저축 기간과 최대 투자 금액을 추가하고 저축기한에 대해서 메시지를 프린트하도록 작성하였다.</p>	<pre>if 은행 == "신한": 연_이자율 = 0.055 # 5.5% 연 이자율 저축_기간 = 3 # 신한은행의 경우 3년 최대_투자_금액 = 400000 # 최대 40만원 print("저축기한은 3년입니다.")</pre>
<p>최대 투자 금액을 추가했기에 초기 투자 금액을 최대 투자 금액 사이에서 다시 정의를 했고 조건문을 작성해 알맞지 않은 금액을 적었을 때 메시지와 함께 프로그램이 종료되도록 했다.</p>	<pre>초기_투자_금액 = float(input(f"얼마를 넣을건가요? (1 ~ {최대_투자_금액}원 사이):")) if 초기_투자_금액 < 1 or 초기_투자_금액 > 최대_투자_금액: print(f"1 ~ {최대_투자_금액}원 사이의 금액을 입력해야 합니다.") exit()</pre>

<p>초기 투자 금액이 의미가 없어서 매달 투자 금액으로 변수 이름을 다시 지정했다.</p>	<pre> 매달_투자_금액 = float(input(f"매달 얼마를 넣을건가요? (1 ~ {최대_투자_금액}원 사이) if 매달_투자_금액 < 1 or 매달_투자_금액 > 최대_투자_금액: print(f"1 ~ {최대_투자_금액}원 사이의 금액을 입력해야 합니다.") exit() </pre>
<p>마지막으로 코드를 깔끔하게 다듬어서 최종적으로 계산기가 만들어졌다.</p>	<pre> 은행 = input("은행을 선택하세요 (신한, 하나, 농협, 기업):") 은행별_정보 = { "신한": {"연_이자율": 0.055, "저축_기간": 3, "최대_투자_금액": 400000}, "하나": {"연_이자율": 0.045, "저축_기간": 2, "최대_투자_금액": 300000}, "농협": {"연_이자율": 0.06, "저축_기간": 4, "최대_투자_금액": 500000}, "기업": {"연_이자율": 0.05, "저축_기간": 3, "최대_투자_금액": 500000} } if 은행 in 은행별_정보: 은행정보 = 은행별_정보[은행] 연_이자율 = 은행정보["연_이자율"] 저축_기간 = 은행정보["저축_기간"] 최대_투자_금액 = 은행정보["최대_투자_금액"] print(f"{은행} 은행, 저축기간은 {저축_기간}년입니다.") else: print("지원하지 않는 은행입니다.") exit() # 얼마를 넣을건지 입력값 받기 매달_투자_금액 = float(input(f"매달 얼마를 넣을건가요? (1 ~ {최대_투자_금액}원 사이):")) if 1 <= 매달_투자_금액 <= 최대_투자_금액: 매달마다_저축할_횟수 = int(input("1년에 몇 번 넣으실건가요?")) future_value = calculate_future_value(매달_투자_금액, 연_이자율, 저축_기간, 매달마다_저축할_횟수) print(f"{은행} 은행, {저축_기간}년 후의 적금 가치는 {future_value:.2f}원 입니다.") else: print(f"1 ~ {최대_투자_금액}원 사이의 금액을 입력해야 합니다.") </pre>

4.2 에러 발생 지점 및 해결책

```
for year in range(저축_기간):
    for month in range(12):
        future_value += 매달_저축_금액 # 매달 저축 금액을 더함

    # 월 이자를 연 이자로 변경
    future_value += (future_value * 연_이자율 / 12) # 연 이자 계산

return future_value
```

적금이 끝난 후 받는 금액을 계산하는 부분에서 굉장히 많은 에러가 발생했다. 정확히는 계산이 맞지 않았다는 것인데 적금의 경우 적금이 끝난 후에 적금 계좌에 있는 금액에 이자율에 따라 이자를 주는 형식인데 너무 큰 금액으로 계산이 되어서 굉장히 애를 먹었다.

```
for _ in range(저축_기간 * 12):
    future_value += 매달마다_저축할_횟수 * 매달_투자_금액

future_value = future_value * (1 + 연_이자율)
return future_value
```

이후 이렇게 코드를 바꾸니 미래 적금 가치는 제대로 나올 수 있도록 코드가 작성되었지만 코드를 실행해보면 값이 나오질 않았다.

```
for year in range(저축_기간):
    for month in range(12):
        future_value += 매달_투자_금액 # 매달 저축 금액을 더함

    future_value *= (1 + 연_이자율) * (매달마다_저축할_횟수/12)
```

반복문을 이중 반복문을 사용하는 식으로 다시 바꾸었고 매달마다 저축할 횟수/12 를 넣지 않으니 매달 몇 번 넣느냐에 따라서 금액이 바뀌질 않고 고정이 되어있어서 future value 를 구하는 코드에 추가를 해 매달 얼마를 넣냐 1 년에 몇 번 넣느냐에 따라서 값이 정상적으로 다르게 나왔다.

<pre>초기_투자_금액 = int(input("얼마를 넣을건가요?")) 연_이자율 = input("연 이자율을 몇 퍼센트인가요?") 저축_기간 = int(input("몇 년짜리의 적금인가요?")) 매달마다_저축할_횟수 = int(input("1년에 몇 번 넣으실건가요?"))</pre>	
연 이자율을 문자열로 받으니 아래 반복문을 통해 계산 시 계산이 실행되지 않았다.	
<pre>초기_투자_금액 = int(input("얼마를 넣을건가요?")) 연_이자율 = float(input("연 이자율을 몇 퍼센트인가요?")) 저축_기간 = int(input("몇 년짜리의 적금인가요?")) 매달마다_저축할_횟수 = int(input("1년에 몇 번 넣으실건가요?")) future_value = 0 for year in range(저축_기간): for month in range(매달마다_저축할_횟수): future_value += 초기_투자_금액 future_value *= (1 + (연_이자율/매달마다_저축할_횟수)) print(f"{저축_기간}년 후의 적금 가치는 {future_value:.2f}원 입니다.")</pre>	
연 이자율을 숫자로 변환시켜주기 위해 float 을 사용해 해결하였다.	

4.3 동작결과

은행과, 금액, 횟수를 입력하니 이후 적금 가치가 잘 계산되었고 입력값도 올바르게 나왔다.	<pre>은행을 선택하세요 (신한, 하나, 농협, 기업): 신한 신한 은행, 저축기한은 3년입니다. 매달 얼마를 넣을건가요? (1 ~ 400000원 사이): 300000 1년에 몇 번 넣으실건가요? 10 신한 은행, 3년 후의 적금 가치는 9495000.00원 입니다.</pre>
범위에 맞지 않는 금액을 적었을 때 알맞은 메시지가 나오면서 프로그램이 종료되었다.	<pre>은행을 선택하세요 (신한, 하나, 농협, 기업): 하나 하나 은행, 저축기한은 2년입니다. 매달 얼마를 넣을건가요? (1 ~ 300000원 사이): 400000 1 ~ 300000원 사이의 금액을 입력해야 합니다. Process finished with exit code 0</pre>

5. 계산기 개발 후기

제가 하나하나 설계해가면서 직접 짜지 않고 chatgpt 를 사용하여 계산기를 만들었지만 계속해서 chatgpt 를 통해 부족한 점을 개선하거나 필요한 부분을 추가해가면서 점점 코드들이 다듬어지고 결과 값이 개선되는 과정을 보면서 나중에 제가 이런 코드들을 만들 수 있는 수준이 되어서 직접 만들게 된다면 지금보다 머리는 아플 수 있지만 굉장히 재밌을 거 같다는 느낌을 받았습니다.

계산기를 만들면서 실제 은행들의 적금 조건 또는 이자율, 매해 바뀌는 적금에 관해서 코드를 넣어서 좀 더 유용하게 만들고 싶다는 생각이 들었지만 그렇게까지 해버리면 제가 코드를 이해를 못할뿐더러 오류가 발생했을 때 무엇이 문제인지 고치는 과정자체도 코드가 복잡해질수록 더 복잡해지기에 간단하게 밖에 못하는 것이 아쉬웠습니다.

적금 계산기 코딩이 어렵지는 않지만 직접 개발하는 과정이 재미있었고 조금 신선했습니다.

교수님 덕분에 컴퓨터 프로그래밍 과목에 관심이 생기고 재밌습니다.
감사합니다!