

컴퓨터 프로그래밍 과제

소속 : 한국교통대학교_안전공학과

학 번 : 2018054

성 명 : 공왕택

먼저 저는 많은 대학생이 고민하듯이 안전공학과를 졸업한다고 해서 안전 쪽으로 취업을 할 수 있을지에 대한 의문이 생겨 전공 주제가 아닌 자유 주제로 선정하여 계산기를 코딩하게 되었습니다.

1. 계산기의 목적 :

많은 대학생들이 졸업 전까지 학기당 몇 학점을 이수해야 하는지 고민하는 상황을 고려하여, 대학생들이 학업을 계획적으로 수행할 수 있도록 돕는 프로그램을 작성했습니다. 이 프로그램은 학기가 끝날 때마다 남은 학점을 계산하여 대학생들이 다음 학기 학업 목표를 설정하고 달성할 수 있도록 도와줍니다. 이를 통해 대학생들이 학업 계획을 더욱 효율적으로 수립할 수 있도록 지원하고자 작성했습니다.

2. 계산기 개발 계획 :

먼저 사용자의 직전 학기 성적, 남은 학점, 남은 학기의 입력값을 받는 함수의 매개변수를 만듭니다.

이후 저희 학교는 직전 학기 성적이 4.0 이상이면 다음 학기 수강 가능한 최대 학점이 23학점이고 4.0 미만일 경우 19학점을 최대로 들 수 있으므로 if문을 사용하여 직전 학기 성적을 비교하여 최대 수강 가능 학점에 대한 조건문을 만듭니다.

```
def 계산된_학점(직전학기성적, 남은_학점, 남은_학기):  
    if 직전학기성적 >= 4.0:  
        다음학기_최대수강학점 = 23  
    else:  
        다음학기_최대수강학점 = 19
```

다음으로 while문을 사용하여 다음 학기 최대 수강 가능 학점보다 많은 학점 입력 시 다시 입력하도록 반복문을 사용합니다.

조건이 충족되면 break문을 사용하여 반복문을 종료합니다.

```
while True:  
    다음학기_수강학점 = float(input(f"다음 학기 수강할 학점을 입력해주세요 (최대 {다음학기_최대수강학점} 학점): "))  
    if 다음학기_수강학점 <= 다음학기_최대수강학점:  
        break  
    else:  
        print(f"입력된 수강학점이 최대 수강 가능 학점을 초과했습니다. 최대 {다음학기_최대수강학점} 학점 이하로 입력해주세요.")
```

그 이후 남은 학점에서 다음 학기 수강학점을 뺀 객체(계산된_학점_결과)를 만들고 그 객체를 남은 학기로 나눈 한_학기_학점이란 객체를 만들어 결과값으로 반환하게 합니다.

```
계산된_학점_결과 = 남은_학점 - 다음학기_수강학점
한_학기_학점 = 계산된_학점_결과 / 남은_학기
return 한_학기_학점
```

마지막으로 함수 사용 결과를 다른 객체에 저장 후 print 명령어를 사용하여 적절한 문구와 함께 출력하면 됩니다.

```
계산된_학점_결과 = 계산된_학점(직전학기성적, 남은_학점, 남은_학기)
print("(다음학기를 제외한) 한 학기에 들어야 하는 학점 :", 계산된_학점_결과)
```

3. 계산기 개발 과정 및 후기 :

- 에러 발생 지점 확인 : 에러 발생 지점은 총 3곳으로

첫 번째는 직전 학기 성적을 입력할 때 정수형을 입력받은 int 함수를 사용했습니다. 대학교 성적의 경우 정수형으로 떨어질 수도 있겠지만 대부분 성적이 소수점으로 떨어지는 실수형이기에 소수점 값 입력 시 오류가 발생했습니다.

```
직전학기성적 = int(input("직전 학기 성적을 입력해주세요: "))
```

두 번째로 남은 학기를 구할 때 다음학기_수강학점을 제외하고 구해야 했는데 그 부분을 인지하지 못해 계산기가 실제 결과와 다른 값을 구했습니다.

```
한_학기_학점 = 계산된_학점_결과 / 남은_학기
```

세 번째로 한 학기에 들어야 하는 학점에 대한 값을 소수점 자리까지 구하게 하여 혼동을 주었습니다. 대학교 수강 학점의 경우 소수점 자리가 없는 정수형을 반환하도록 해야 합니다.

```
한_학기_학점 = 계산된_학점_결과 / 남은_학기
```

- 에러 해결 방법 제시 :

첫 번째 에러의 경우 int (문자열을 정수형으로 바꿔주는 함수) 함수를 float (주어진 값이나 문자열을 부동 소수점 숫자로 변환해 주는 함수)로 변환하여 오류를 해결했습니다.

```
직전학기성적 = float(input("직전 학기 성적을 입력해주세요: "))
```

두 번째 에러의 경우 다음학기_수강학점을 단순히 빼주기만 하면 되므로 -1을 계산식에 추가합니다.

```
한_학기_학점 = 계산된_학점_결과 / (남은_학기 - 1) # 다음학기_수강학점을 제외시켜야 하므로 -1
```

세 번째 나눗셈의 몫값을 정수형까지 구하는 명령어인 '//'를 사용해 해결했습니다.

```
한_학기_학점 = 계산된_학점_결과 // (남은_학기 - 1)
```

- 해결 방법 적용 시 변화한 내용

첫 번째 오류

수정 전

```
직전 학기 성적을 입력해주세요: >? 3.4
Traceback (most recent call last):
  File "C:\Users\공왕택\AppData\Local\Programs\Python\Python38\lib\code.py", line 90, in runcode
    exec(code, self.locals)
  File "<input>", line 18, in <module>
ValueError: invalid literal for int() with base 10: '3.4'
```

수정 후

```
직전 학기 성적을 입력해주세요: >? 3.4
남은 학점을 입력해주세요:
```

실수형을 입력해도 오류가 뜨지 않고 다음 입력값을 받는 input 함수로 넘어가는 걸 확인할 수 있다.

두 번째 오류

수정전

```
직전 학기 성적을 입력해주세요: >? 3.8
남은 학점을 입력해주세요: >? 60
남은 학기를 입력해주세요: >? 4
다음 학기 수강할 학점을 입력해주세요 (최대 19 학점): >? 19
(다음학기를 제외한) 한 학기에 들어야 하는 학점 : 10
```

저의 경우 이번 학기를 끝내면 총 80학점을 이수하게 됩니다. 졸업 학점인 140학점까지 60학점이 남아 있으니 다음 학기 19학점을 듣는 경우 한 학기 동안 13학점을 들으면 됩니다.

하지만 실행 결과 10학점이 나오는데 그 이유는 다음 학기 수강할 학점을 남은 학기에서 제외시키고 계산해야 하지만 그러지 못하여 계산값이 다르게 나왔습니다.

수정 후

```
직전 학기 성적을 입력해주세요: >? 3.8
남은 학점을 입력해주세요: >? 60
남은 학기를 입력해주세요: >? 4
다음 학기 수강할 학점을 입력해주세요 (최대 19 학점): >? 19
(다음학기를 제외한) 한 학기에 들어야 하는 학점 : 13
```

세 번째 오류

수정 전

```
(다음학기를 제외한) 한 학기에 들어야 하는 학점 : 19.666666666666668
```

수정 후

(다음학기를 제외한) 한 학기에 들어야 하는 학점 : 19.0

실수형에서 정수형으로 떨어지는 것을 확인할 수 있습니다.

- 개발 후 느낌점 :

프로그램 하나를 개발하는 데에 생각보다 많은 변수와 함수가 필요하다는 걸 몸소 경험할 수 있었던 기회였습니다. 위 코드의 경우에 변수를 입력받을 매개변수가 4개가 필요하고 각 변수 입력값에 따른 함수도 다양한 걸 볼 수 있습니다.

만약 저기서 계절학기를 추가하려면 전체 코드를 다시 수정해야 하므로 코드를 한번 작성할 때 초기 제작 과정에서 많은 심의를 기울여야 한다는 것도 배울 수 있었습니다.

또 반복문을 단순히 특정 코드 블록을 반복하는 것이 아닌 조건을 검사하는 방법으로 사용할 수 있다는 것도 배울 수 있었습니다.

마지막으로, ChatGPT의 효용성을 깨달았습니다. 코드의 오류를 빠르게 식별하고 수정하는 과정에서 ChatGPT의 지원이 큰 도움이 되었습니다. 이를 통해 학습에 필요한 지식과 해결책을 빠르게 얻을 수 있었고, 앞으로 학업에 많은 도움이 될 것 같습니다.

4. 계산기의 효과

: 해당 계산기를 활용하여 대학생들이 졸업 학기 전까지 평균적으로 몇 학점을 들어야 하는지 계산할 수 있어 계획적으로 학업을 이어갈 수 있을 거 같습니다.

저를 예시로 들었을 경우 매 학기가 끝나고 다음 학기와 졸업 학기 전까지 평균적으로 몇 학점을 들어야 하는지 계산하기 바빴지만 해당 계산기를 활용하면 보다 빠르고 효과적으로 계산할 수 있을 것입니다. 다만 정규 학기가 아닌 계절학기를 수강할 대학생들에게는 효과적이지 않을 것으로 예상됩니다.