**프로그래밍 언어개론 과제 보고서**

Item02

Cute17 Project

01분반\_13조

201204441 김수현

권오현

201102448 사명기

1. **구현 방식**
2. **Define**

* Define의 구현에 앞서 Define의 정의 및 이해가 필요하였다. Define이란 (define a 1)과 같이 사용되며, 파이썬에서 딕셔너리 자료형 구조를 사용하여 a라는 key값이 들어오면 뒤에 오는 1은 value로써 저장을 하게 된다. 즉, 딕셔너리 구조에 맞춰 {key:value, key2:value2}형태로 저장되는 것이다.
* 위의 define을 통해 저장해야할 key와 value를 저장할 전역 딕셔너리 매트릭스가 필요하였으며, insertDicM이라는 딕셔너리 매트릭스를 만들었다. (이하 insertDicM)
* 해당 key, value를 받아서 저장할 기능을 수행할 함수가 필요하였다. 이를 위하여 def insert\_table를 만들었으며, 해당 부분에서는 l\_node.value(ID값)와 run\_expr(r\_node)(INT 등 value로서 저장할 값)의 값을 각각 key와 value로 받아서 insertDicM에 저장하게 하였다.
* insertDicM에 저장한 값을 가져다가 run\_func(op\_code\_node)에 정의된 함수들이 가져다가 사용을 해야하고, 혹, Key값을 입력하면 insertDicM에 존재하면 해당 value를 연산시에 사용해야하기 때문에 lookup\_table(id)이라는 함수를 만들어주었다.
* 앞서 위에서 연산시, 해당 값이 define을 통해 저장된 key:value쌍이 있는지 lookup\_table에서 한번 찾아보고자 run\_expr내의 ID타입시에 해당 lookup\_table에 들어가 해당 부분의 과정을 거치게 하였다.
* 해당 lookup\_table은 id값(key값)을 받아와서 insertDicM[id]을 하여 해당 key값을 딕셔너리에서 찾아서 key에 대응되는 value값을 return하게 하였고, 없으면 그대로 넣은 node.value값을 반환하게 구성하였다.
* 사용자가 (define a 1)을 한 후 a 를 쳤을 시 해당 key인 a가 insertDicM에 존재한다면 해당 값을 출력시켜야 하기 때문에 print\_node부분에 ID타입 INT타입 노드가 들어오는 부분에서 조건을 넣어 insertDicM에 해당 key(“a”)가 있다면 insertDicM에서 대응되는 value값을 반환하게 코드를 수정하였다.

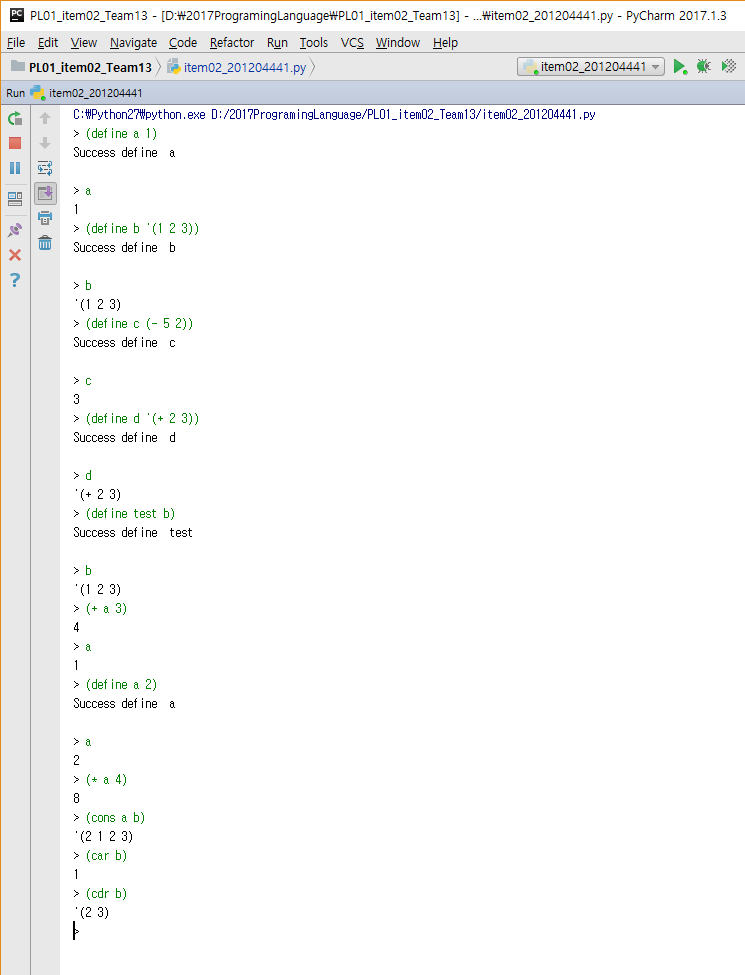
1. **LAMBDA**

* 람다의 경우 식을 대응시켜 저장하는 형태이고 ((LAMBDA (x) (+ x 1)) 2) 일시 해당 2의 값을 x에 대입하여 (+ 2 1)을 수행하게 끔 하는 형태이다.
* 또한 함수형 언어에서 define과 lambda를 이용하여 해당 식을 저장하고 이를 불러다 사용하는 형태로 되어있다. 이를 구현하기 위한 LAMBDA가 필요한 것이다.
* 이를 위하여 앞서 LAMBDA를 정의해줄 필요가 있다. 허나, LAMBDA라는 함수 자체가 이미 Python에 존재하기 때문에 해당 부분에 있어서 run\_lambda로 대체하여 이를 table에 넣어 인식하게 하였다.

1. **역할 분담 및 느낀점**

|  |  |
| --- | --- |
| 김 수 현 | Define의 전체적인 부분을 코딩하였고, 해당 부분에 있어서 오류가 발생하여 팀원들의 피드백 및 refactory를 거쳐서 오류를 완전히 수정하고, test code의 define에 해당하는 7번까지 모두 결과를 뽑아낼 수 있었다. refactory과정에서 존재하였던 기타 수정할 코드 및 오류사항들을 잡았다. 추가적으로 팀의 Github사용을 모두 총괄하여 진행하였다. 팀의 github관리와 필요시 교육 등을 담당하였다. 이번에 파이썬을 통하여 interpreter를 구현함으로서, 코딩에 약간은 자신감이 붙었지만 아직 많이 부족함을 느낀다. 추가적으로 git에 대한 사용법은 기본사용법은 모모 익혔지만, 팀원들에게 모두 완벽히 이해하게 전파하지 못한 점은 아쉬웠다. LAMDA의 구현이 확실하게 완성이 되었으면 하였지만 이 부분은 여전히 아쉬움으로 남는다. 서로 코딩을 하면서 리뷰하고 장단점을 파악하여, 고쳐나감으로서 스스로 얻을 수 없는 값진 경험을 한 것이 좋았다. 추후 코드 중복이 많은 run\_func부분과 구현 못한 LAMBDA를 다시 한번 도전해보면 어떨까 한다. |
| 권 오 현 |  |
| 사 명 기 |  |

1. **결과 화면**



1. **Git 주소**

https://github.com/skyfall13/PL01\_item02\_Team13.git