**2017년 1학기**

**프로그래밍언어개론 설계 계획서**

**(팀이름) : 자... 이제 누가 버스기사지???**

<작업의 분할>

Item 1

-각자 구현

Item 2

a. define 테이블 생성 및 구현 (동하)

b. insertTable 구현 (동하)

c. lookupTable 구현 (진수)

d.기존 코드에서 INT Type 대신 ID Type입력시 Table에서 꺼내오도록 수정

(cons ~ plus 부분 -> 동하) (minus ~ eq 부분 -> 진수)

=> run\_expr 수정으로 모두 적용

e. lambda 구현

(평일 짬짬이 하며 주말에 같이 동방에서 구현)

Symbol Table 추가

Define에서 Lambda 처리

Lambda 변수 바인딩 및 삭제

Lambda 변수 여러 개 처리

Lambda 옵션 적용

구현 일정 계획

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 작업내용 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 6.1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 팀결성 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item2 a |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item2 b |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item2 c |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item2 d |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Item2 e |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| refactoring |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bug fix |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* 1. 팀 구성

|  |  |  |  |
| --- | --- | --- | --- |
| **학번** | **이름** | **역할** | |
| **Item1** | **Item2** |
| **201202141** | **김동하** | **All** | **defTable ,insertTable구현, Symbol Table 생성, Lambda 기본구현, Lambda 옵션 처리,**  **test, refactoring** |
| **201203391** | **임진수** | **All** | **lookupTable구현, run\_expr 수정, Lambda 변수 바인딩 및 삭제. Lambda 다중 변수 처리,**  **test, refactoring** |

\* 팀장 (조교가 연락을 취할 수 있는 대표) : 김동하

1-3 수정 로그

17.05.21 Item1 구현 Test 코드 삭제 및 input 추가

17.05.21 define 구현

17.05.22 insertTable 구현

17.05.22 lookupTable 구현

* + 1. 최종 완료
  1. 문제 해결방식

Define )

l\_node은 define의 key 값이 되고 r\_node의 값은 value 값이 됩니다. Define을 할 때는 기본적을 value 값을 run\_expr 해서 리스트 내부의 값을 실행한 후 quote 된 값만 저장을 합니다. 하지만 저장해야할 value가 lambda 라면 실행을 하지 않고 lambda 식을 저장합니다.

insertTable )

defTable[id] = value 이런 식으로 l\_node의 value(해당 ID 노드의 값)을 id로 받아 key로 사용하고 r\_node를 value로 해서 저장합니다.

lookupTable )

defTable에 id라는 key 값을 가진 페어가 있다면 저장된 value를 리턴하고 없으면 none을 리턴하도록 했습니다.

ID값 복원 )

Define으로 정의한 ID를 표현식에서 다시 복원을 하도록 run\_expr 함수에서 기존 id 타입이면 해당 노트를 리턴하고 끝이였는데 그 곳에서 id이면 lookupTable 함수를 이용하여 값이 있을 경우 해당 value를 리턴하여 각 오퍼레이션 함수 내부에서 run\_expr()을 실행하여 new\_node를 만들 때 자동으로 치환 되도록 구현을 하였습니다.

Lambda )

Lambda의 기본 모형은 argument까지 포함하였을 때 ((lambda (x) (defines) (expression)) argument) 이런 모형입니다. 여기서 기존 다른 오퍼레이션과 다르게 lambda는 리스트 내부가 아닌 밖에 argument를 가져와서 함수에서 사용을 합니다. 그래서 lambda 함수를 실행하기전에 run\_list에서 내부에 lambda가 있는지 검사를 한 후 argument에 대하여 처리를 한 후 lambda 식으로 넘어가서 argument를 받아서 사용하도록 구현하였습니다.