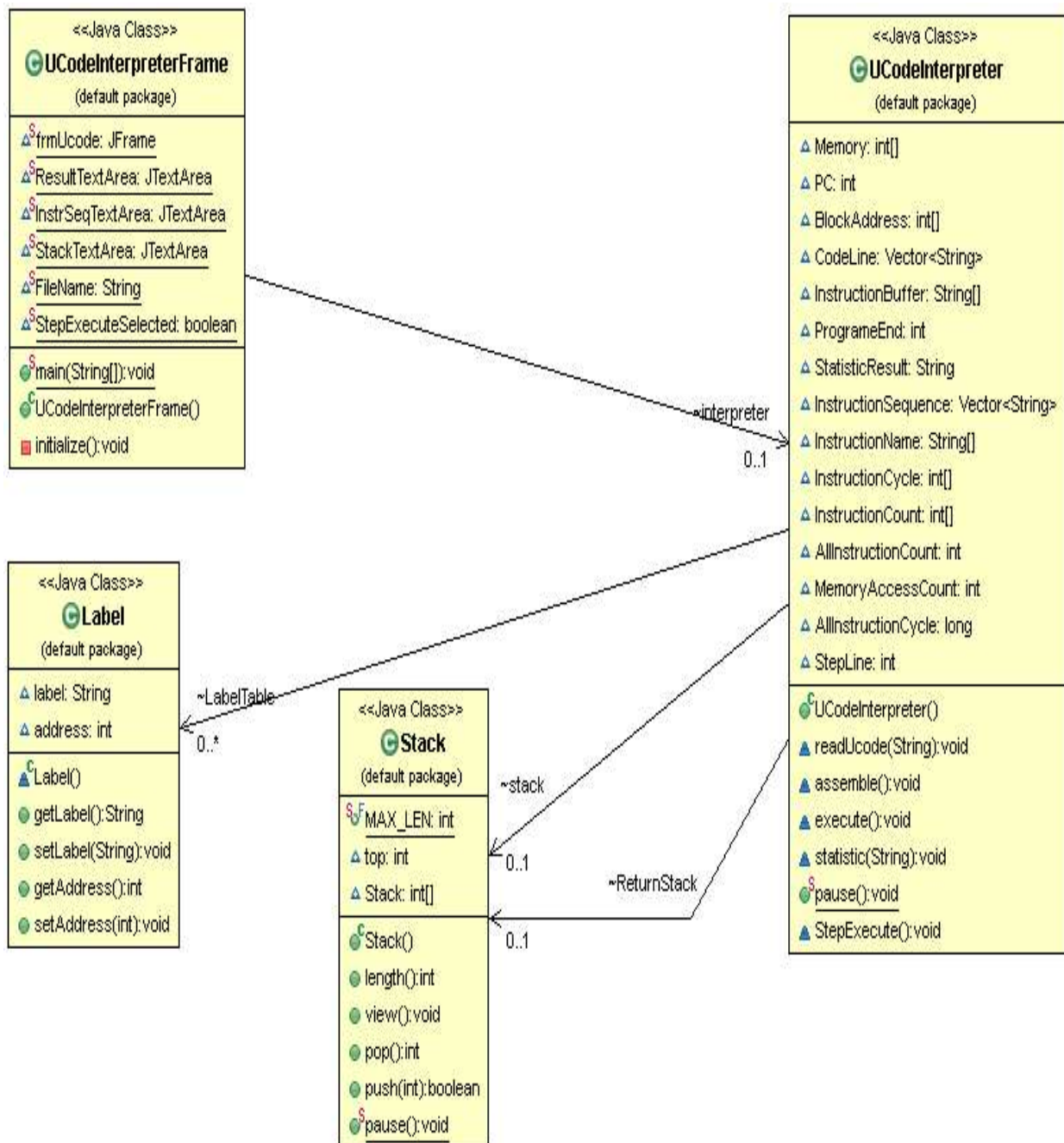

프로젝트 결과 보고서

U-Code Interpreter (GUI)

과목명	소프트웨어프로젝트
학 과	컴퓨터공학과
이 름	김민성
제작기간	2015.11 ~ 2015.12

1 설계 추가 및 보완

■ 클래스 다이어그램 수정



<수정된 클래스 다이어그램>

■ 설명

GUI를 구현하기위해서, CUI에서 구현했던 클래스들을 그대로 가져와서 쓰되, GUI로 프로그램을 구동시키기위해서 JFrame을 상속받은 UCodeInterpreterFrame 클래스를 새로이 선언하였습니다.GUI틀을 만들기 위해서 WindowBuilder를 사용하였고, 단계별로 보여줄수있도록 하기위해서, UCodeInterpreter에 StepExecute() 메소드를 추가하였습니다.StepExecute메소드는 Execute메소드와 거의 유사하지만, 단계별로 실행하기위해서, 반복문을 통해서, 명령어를 해석하는것이아니라, 버튼을 클릭할때마다, 명령어를 하나씩 실행하도록 메소드를 만들었습니다.또한 스택클래스에도 단계별로 실행에따른 스택의 상황을 볼수있도록 스택에 들어있는 값을 보여주는 view() 메소드를 추가하였습니다.

■ GUI 인터페이스 설계



1. 명령어 배열에서 읽어온 명령어를 출력하는 화면입니다.
2. 명령어를 실행하면서, 연산을 위해 사용되는 임시기억장치인 Stack의 상태를 보여주는 화면입니다.
3. 읽어온 명령어의 레이블과 그에 해당하는 주소를 보여주는 화면입니다.
4. 명령어를 실행한 결과를 보여주는 화면입니다.
5. 명령어의 실행결과를 바로 확인할 때 사용하는 버튼입니다.
6. 명령어를 하나씩 실행하면서 스택의상태를 확인하고싶은 경우 사용하는 버튼입니다.
7. 명령어를 하나씩 실행 하지않고, 레이블단위로 Jump하면서 스택의 상태를 확인할 때 사용하는 버튼입니다.
8. 사용된 명령어 수, 각 명령어별로 실행된 수,메모리 접근횟수 등 실행결과에대한 통계파일을 생성하고 싶을 때, 사용하는 버튼입니다.
9. U-CODE파일을 읽어올 때 사용하는 버튼입니다.

2 GUI 개발 계획

개발 기간 15년 11월 30일 ~ 15년 12월 07일							
개발내용 \ 기간	11/30~ 12/1	12/1~ 12/2	12/2~ 12/3	12/3~ 12/4	12/4~ 12/5	12/5~ 12/6	
단계별로 보여주기 위한 알고리즘 작성							
GUI 기반 U-CODE Interpreter 클래스 다이어그램 작성							
GUI 기반 U-CODE Interpreter 소스코드 작성							
테스트 및 디버깅							
통계파일(*.lst) 개선 통계 부분에 명령어별 소요된 사이클 계산 추가 결과 보고서 작성							

3 기능 리스트

기본 기능	완성 여부	설명
난이도 별로 3가지의 U-CODE 파일 작성	O	설계 발표에서 말씀하셨던 난이도 별로 U-CODE를 준비하라고 하셔서, 피보나치의 수열을 1번째 열부터 10번째 열까지 출력하는 Fibo.uco와 퀵정렬을 사용하여 사용자로부터 받은 입력을 오름차순으로 정렬하는 QuickSort.uco를 준비하였습니다.
(*uco)파일을 해석하여 화면에 실행결과 출력	O	콘솔 창으로부터 사용자의 입력을 받아서 실행결과를 화면에 출력하는 기능을 구현하였습니다.
실행 과정 및 통계정보를 보여주는 파일 생성(*.lst)	O	사용자의 입력에 따른 실행 결과를 바탕으로, 통계파일을 작성하여 텍스트파일(*.lst)로 출력하도록 구현하였습니다.

확장 기능	완성 여부	설명
GUI 구현	O	사용자 인터페이스를 편리하게 하기위해서, GUI로 구현하였습니다.사용자의 입력을 다이얼로그를 통해서 입력받아서, 실행결과 창에다가 출력하고, 레이블 테이블, 스택 현황,실행되는 명령어를 실시간으로 볼수있도록 구성하였습니다.
실행 과정을 단계별로 보여준다. (스택 현황을 보여줌)	O	1단계씩 사용자가 실행해가면서, 스택의 현황을 보면서 실행하도록 구현하였습니다.
통계파일 내용 추가	O	통계파일의 내용을 개선하여, 명령어 당 필요한 사이클을 정의하고, 명령어가 사용된 횟수를 이용하여, 명령어 당 사용된 사이클수를 계산하여 텍스트파일에 출력하도록 추가하였습니다.또한, 정의한 사이클에 대해서, 알수있도록 <부록>으로 명령어 당 필요한 사이클 수도 추가하였습니다.
실행 과정을 단계별로 보여주는 기능 개선	O	1단계씩 실행 할 경우, 오래 걸린다는 단점이있는데, 이 점을 개선하기위해서 단계별로 보여주는 기능을 개선하여서, 레이블 단위로 실행하도록 수정하였습니다.

4 프로젝트를 진행하면서 느낀점

CUI를 구현하는 과정에서 필요한 알고리즘의 대부분을 개발한 상황에서 시작한 GUI 개발이어서, 구현하는데 시간이 상당히 많이 단축되었습니다. 이번 프로젝트를 통해서, 자바에 대해서 자신감도 많이 생긴것같고, 컴퓨터 구조에 대한 이해도도 많이 향상된것같습니다. 특히, 이번 프로젝트를 통해서 평소에 어려워했던 재귀적인 구조를 이해하는데 많은 도움이되었던 것 같습니다.

5 향후 개선사항

기능리스트에서 언급했던것과같이 단계별로 실행시에, 한단계씩 실행하다 보니까, 프로그램의 실행에 많은 시간과 노력((?)마우스 클릭)이 소요되는 단점이 있었습니다.이 점을 개선하려면, IDE에서 제공해주는 디버깅기능처럼 실행단계를 Jump하거나, 브레이크 포인트까지 바로 실행하는 기능 등을 제공하여야하는데, 이 점에대해서는 아직 자바프로그래밍이 미숙하다보니 구현을 하지못했습니다.다음에 자바프로그래밍 실력이 더 향상된다면 이 프로그램을 개선하여 단계별로 실행이아닌 완벽한 디버깅 기능을 수행하는 프로그램으로 개발해보고싶습니다.

6 프로젝트를 진행하면서 어려웠던 점

WindowBuilder를 통해서, 개발을 하다보니, 인터페이스 개발하는데는 유용하였지만,코드를 분석하는데는 어려움이 있었습니다.또한, 자바 GUI에서 사용되는 스윙 컴포넌트들을 잘 모르다보니, 인터넷과 책을 일일이 찾아가면서 ,프로그램을 작성하는데 많은 시간이 걸렸던것 같습니다.

7 실행 결과

factorial.uco실행결과 (단계별로 실행)

명령어 실행 과정

Line	Instruction	Operand1	Operand2	Operand3
1	bgn	0		
2	ldp			
3	end			
4	proc	2	2	
5	sym	2	0	1
6	sym	2	1	1
7	ldp			
8	lda	2		

Stack

1

레이블 테이블

Address	Label
4	factorial
14	\$\$0
23	\$\$1
25	main

실행 결과

입력

사용자 입력

확인 취소

실행

단계별로 실행

통계파일 생성

파일 불러오기

제작자 : 김민성

U-CODE 인터프리터

명령어 실행 과정

71	nop			
72	lod	1	0	
73	ldp			
74	lod	1	0	
75	ldc	1		
76	sub			
77	end			
78	proc	1	1	
79	sym	1	0	1
80	str	1	0	
81	lod	1	0	
82	ldc	1		
83	eq			
84	fjp	\$\$0		
85	ldc	1		

Stack

1
2
3
4
5
6

레이블 테이블

Address	Lable
4	factorial
14	\$\$0
23	\$\$1
25	main

실행

단계별로 실행


통계파일 생성

파일 불러오기

실행 결과

사용자 입력 : 6

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정

88	call	factorial		
89	mult			
90	retv			
91	nop			
92	call	factorial		
93	mult			
94	retv			
95	nop			
96	call	factorial		
97	mult			
98	retv			
99	nop			
100	call	factorial		
101	mult			
102	retv			

Stack

120
6

레이블 테이블

Address	Lable
4	factorial
14	\$\$0
23	\$\$1
25	main

실행

단계별로 실행


통계파일 생성

파일 불러오기

실행 결과

사용자 입력 : 6

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정

```

101      mult
102      retv
103      nop
104      call      factorial
105      mult
106      retv
107      nop
108      call      factorial
109      str      2      1
110      ldp
111      lod      2      1
112      call      write
113      call      lf
114      ret
115      end

```

Stack

레이블 테이블

Address	Lable
4	factorial
14	\$\$0
23	\$\$1
25	main

실행

단계별로 실행


통계파일 생성

파일 불러오기

실행 결과

사용자 입력 : 6
720
프로그램이 정상적으로 종료되었습니다. End Main

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정

```

101      mult
102      retv
103      nop
104      call      factorial
105      mult
106      retv
107      nop
108      call      factorial
109      str      2      1
110      ldp
111      lod      2      1
112      call      write
113      call      lf
114      ret
115      end

```

Stack

레이블 테이블

Address	Lable
4	factorial
14	\$\$0
23	\$\$1
25	main

실행

단계별로 실행


통계파일 생성

파일 불러오기

실행 결과

사용자 입력 : 6
720
프로그램이 정상적으로 종료되었습니다. End Main

제작자 : 김민성



```

Java - UCodeInterpreter_GUI_PROJECT/src/UCodeInterpreterFrame.java - Eclipse
factorial.lst - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)
115          end
-----
*****실행 결과*****
720
-----
*****명령어 별 실행 횟수*****
nop      5      ldi      0      sti      0      not      0      neg      0
inc      0      dec      0      dup      0      add      0      sub      5
mult     5      div      0      mod      0      gt       0      lt       0
ge       0      le       0      eq       6      ne       0      and      0
or       0      swp      0      ldp      9      ret      1      retv     6
end      6      bgn      1      ujp      1      tjp      0      fjp      6
ldc      12     call     10     lod      18     lda      1      str      7
proc     7      sym      8
-----
*****메모리 접근 횟수*****
메모리 접근 명령어 : ldi,sti,lod,lda,str
메모리 접근 횟수 : 26
-----
*****총 명령어 실행 횟수*****
총 명령어 실행 횟수 : 114
-----
*****명령어 별 사용 비율*****
nop      4.386%
ldi      0%
sti      0%
not      0%
neg      0%
inc      0%
dec      0%
dup      0%
add      0%
sub      4.386%
mult     4.386%
div      0%
mod      0%
gt       0%
lt       0%
ge       0%
le       0%

```

```

factorial.lst - 메모장
파일(F)  편집(E)  서식(O)  보기(V)  도움말(H)

div      0 Cycle
mod      0 Cycle
gt       0 Cycle
lt       0 Cycle
ge       0 Cycle
le       0 Cycle
eq       120 Cycle
ne       0 Cycle
and      0 Cycle
or       0 Cycle
swp      0 Cycle
ldp      90 Cycle
ret      30 Cycle
retv     180 Cycle
end      0 Cycle
bgn      0 Cycle
ujp      10 Cycle
tjp      0 Cycle
fjp      60 Cycle
ldc      60 Cycle
call     300 Cycle
lod      90 Cycle
lda      5 Cycle
str      35 Cycle
proc     210 Cycle
sym      0 Cycle

*****명령어에 사용된 총 사이클 수*****

프로그램을 실행하는데 사용된 총 사이클 수 : 1515 Cycle

*****부                                  른*****

*****명령어 별 필요한 사이클 수*****

nop      5      ldi      10      sti      10      not      5      neg      5
inc      1      dec      1      dup      5      add      10      sub      10
mult     50     div      100     mod      100     gt       20      lt       20
ge       20     le       20     eq       20     ne       20     and      10
or       10     swp      10     ldp      10     ret      30     retv     30
end      0      bgn      0      ujp      10     tjp      10     fjp      10
ldc      5      call     30     lod      5      lda      5      str      5
proc     30     sym      0

```

■ QuickSort.uco 실행결과 [실행]

U-CODE 인터프리터

명령어 실행 과정				Stack	레이블 테이블
1184	lda	4	0		103 \$\$4
1185	add				112 QuickSort
1186	ldi				144 \$\$5
1187	call	write			147 main
1188	lod	4	101		157 \$\$6
1189	inc				174 \$\$7
1190	str	4	101		186 \$\$8
1191	nop				201 \$\$9
1192	lod	4	101		
1193	lod	4	102		
1194	lt				
1195	nop				
1196	call	lf			
1197	ret				
1198	end				

실행 결과

사용자 입력 : 9
 사용자 입력 : 7
 사용자 입력 : 100
 사용자 입력 : 3
 사용자 입력 : 0
 3 5 7 9 31 100

Stack

레이블 테이블


실행

단계별로 실행

통계파일 생성

파일 불러오기

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정				Stack	레이블 테이블
1184	lda	4	0		103 \$\$4
1185	add				112 QuickSort
1186	ldi				144 \$\$5
1187	call	write			147 main
1188	lod	4	101		157 \$\$6
1189	inc				174 \$\$7
1190	str	4	101		186 \$\$8
1191	nop				201 \$\$9
1192	lod	4	101		
1193	lod	4			
1194	lt				
1195	nop				
1196	call	lf			
1197	ret				
1198	end				

통계파일 생성

통계파일 (*.lst)이 생성되었습니다.

확인

실행 결과

사용자 입력 : 9
 사용자 입력 : 7
 사용자 입력 : 100
 사용자 입력 : 3
 사용자 입력 : 0
 3 5 7 9 31 100

Stack

레이블 테이블


실행

단계별로 실행

통계파일 생성

파일 불러오기

제작자 : 김민성



```

QuickSort.lst - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

1194      lt      % i < len
1195      $$9      nop
1196      call lf
1197      ret
1198      end

*****실행 결과*****
3 5 7 9 31 100

*****명령어 별 실행 횟수*****
nop      18      ldi      43      sti      20      not      0      neg      0
inc      15      dec      6      dup      0      add      85      sub      7
mult     0      div      0      mod      0      gt      0      lt      7
ge       17     le       41     eq       0      ne      7      and     17
or        0     swp      0      ldp      40     ret      21     retv    6
end       27     bgn      1      ujp      22     tjp      0      fjp     55
ldc       39     call     41     lod      363    lda      20      str     133
proc      27     sym      120

*****메모리 접근 횟수*****
메모리 접근 명령어 : ldi,sti,lod,lda,str
메모리 접근 횟수 : 605

*****총 명령어 실행 횟수*****
총 명령어 실행 횟수 : 1198

*****명령어 별 사용 비율*****
nop      1.5025%
ldi      3.5893%
sti      1.6694%
not      0%
neg      0%
inc      1.2521%
dec      0.5008%
dup      0%
add      7.0952%
sub      0.5843%
mult     0%
div      0%
mod      0%

```


QuickSort.lst - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

div 0 Cycle

mod 0 Cycle

gt 0 Cycle

lt 140 Cycle

ge 340 Cycle

le 820 Cycle

eq 0 Cycle

ne 140 Cycle

and 170 Cycle

or 0 Cycle

swp 0 Cycle

ldp 400 Cycle

ret 630 Cycle

retv 180 Cycle

end 0 Cycle

bgn 0 Cycle

ujp 220 Cycle

tjp 0 Cycle

fjp 550 Cycle

ldc 195 Cycle

call 1230 Cycle

lod 1815 Cycle

lda 100 Cycle

str 665 Cycle

proc 810 Cycle

sym 0 Cycle

*****명령어에 사용된 총 사이클 수*****

프로그램을 실행하는데 사용된 총 사이클 수 : 10066 Cycle

*****부

*****명령어 별 필요한 사이클 수*****

nop 5

inc 1

mult 50

ge 20

or 10

end 0

ldc 5

proc 30

ldi 10

dec 1

div 100

le 20

swp 10

bgn 0

call 30

sym 0

sti 10

dup 5

mod 100

eq 20

ldp 10

ujp 10

lod 5

not 5

add 10

gt 20

ne 20

ret 30

tjp 10

lda 5

neg 5

sub 10

lt 20

and 10

retv 30

fjp 10

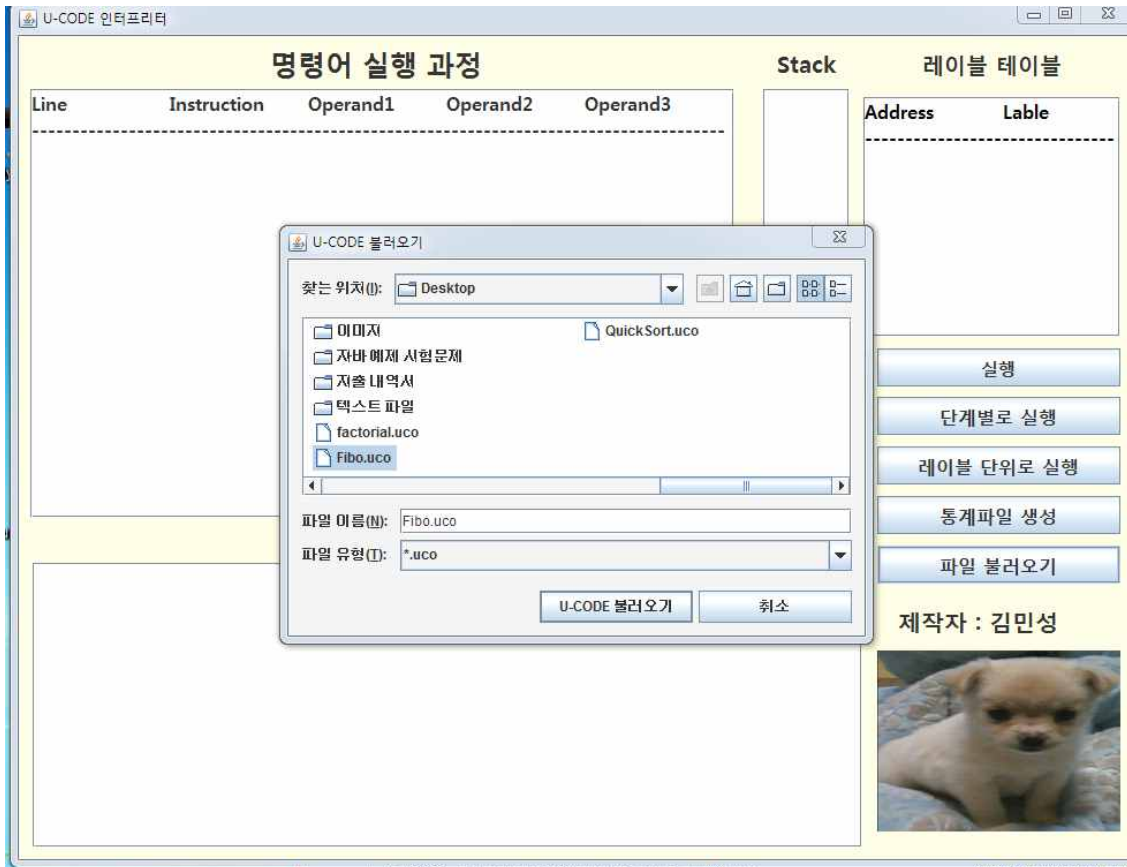
str 5

- 13 -

■Fibo.uco실행결과 (레이블 단위로 실행)

☺위의 실행화면과 인터페이스가 약간 다른 이유☺

수요일까지 결과발표일이 연장되면서, ‘레이블 단위로 실행’ 기능을 추가하였습니다.



U-CODE 인터프리터

명령어 실행 과정

301	eq		
302	fjp	\$\$0	
303	ldc	0	
304	retv		
305	nop		
306	call	Fibo	
307	add		
308	retv		
309	nop		
310	call	Fibo	
311	swp		
312	ldc	2	
313	sub		
314	end		
315	proc	1	1

Stack


2
1
5

레이블 테이블

Address	Lable
4	Fibo
14	\$\$0
22	\$\$1
34	\$\$2
36	main
40	\$\$3
52	\$\$4

실행
단계별로 실행
레이블 단위로 실행
통계파일 생성
파일 불러오기

제작자 : 김민성



실행 결과

0 1 1 2

U-CODE 인터프리터

명령어 실행 과정

1532	eq		
1533	nop		
1534	lod	1	0
1535	ldc	2	
1536	eq		
1537	fjp	\$\$1	
1538	ldc	1	
1539	retv		
1540	nop		
1541	call	Fibo	
1542	swp		
1543	ldc	2	
1544	sub		
1545	end		
1546	proc	1	1

Stack

1
1
4
5
5
8

레이블 테이블

Address	Lable
4	Fibo
14	\$\$0
22	\$\$1
34	\$\$2
36	main
40	\$\$3
52	\$\$4

실행
단계별로 실행
레이블 단위로 실행
통계파일 생성
파일 불러오기

제작자 : 김민성



실행 결과

0 1 1 2 3 5 8

U-CODE 인터프리터

명령어 실행 과정

1914	ldc	1
1915	retv	
1916	nop	
1917	call	Fibo
1918	add	
1919	retv	
1920	nop	
1921	call	Fibo
1922	add	
1923	retv	
1924	nop	
1925	call	Fibo
1926	add	
1927	retv	
1928	nop	

Stack

13

레이블 테이블

Address	Lable
4	Fibo
14	\$\$0
22	\$\$1
34	\$\$2
36	main
40	\$\$3
52	\$\$4

실행 결과

0 1 1 2 3 5 8

실행


단계별로 실행

레이블 단위로 실행

통계파일 생성

파일 불러오기

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정

5238	retv	
5239	nop	
5240	call	Fibo
5241	call	write
5242	lod	2 0
5243	inc	
5244	str	2 0
5245	nop	
5246	lod	2 0
5247	ldc	11
5248	lt	
5249	nop	
5250	call	lf
5251	ret	
5252	end	

Stack

레이블 테이블

Address	Lable
4	Fibo
14	\$\$0
22	\$\$1
34	\$\$2
36	main
40	\$\$3
52	\$\$4

실행 결과

0 1 1 2 3 5 8 13 21 34

프로그램이 정상적으로 종료되었습니다. End Main

실행


단계별로 실행

레이블 단위로 실행

통계파일 생성

파일 불러오기

제작자 : 김민성



U-CODE 인터프리터

명령어 실행 과정

```

5238      retv
5239      nop
5240      call      Fibo
5241      call      write
5242      lod        2          0
5243      inc
5244      str        2          0
5245      nop
5246      lod        2          0
5247      ldc       11
5248      lt
5249      nop
5250      call      lf
5251      ret
5252      end

```

list파일 생성

?

통계파일(*.lst)을 생성하시겠습니까?

예(Y)

아니오(N)

Stack

레이블 테이블

Address	Lable
4	Fibo
14	\$\$0
22	\$\$1
34	\$\$2
36	main
40	\$\$3
52	\$\$4

실행

단계별로 실행

레이블 단위로 실행

통계파일 생성

파일 불러오기


실행 결과

```

0 1 1 2 3 5 8 13 21 34
프로그램이 정상적으로 종료되었습니다. End Main

```

제작자 : 김민성



```

Fibo.lst - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
5244          str 2 0          % ++i
5245    $$3      nop          % Begin Of For
5246          lod 2 0
5247          ldc 11
5248          lt              % i < 11
5249    $$4      nop          % End Of For
5250          call 1f          % '\n'
5251          ret
5252          end

*****실행 결과*****
0 1 1 2 3 5 8 13 21 34

*****명령어 별 실행 횟수*****
nop      134      ldi      0      sti      0      not      0      neg      0
inc       10      dec      0      dup      133     add      133     sub      266
mult      0      div      0      mod      0      gt       0      lt       11
ge        0      le       0      eq       497     ne       0      and      0
or        0      swp      133     ldp      11     ret       1      retv     276
end       276     bgn      1      ujp      153     tjp      0      fjp      508
ldc      918     call     288     lod      661    lda       0      str      287
proc      277     sym      277

*****메모리 접근 횟수*****
메모리 접근 명령어 : ldi,sti,lod,lda,str
메모리 접근 횟수 : 948

*****총 명령어 실행 횟수*****
총 명령어 실행 횟수 : 5251

*****명령어 별 사용 비율*****
nop      2.5519%
ldi      0%
sti      0%
not      0%
neg      0%
inc      0.1904%

```