

제 목.	파이썬 프로젝트				
과목 명.	스크립트 프로그래밍				
담 당.	노은영		교 수 님		
제출 일.	2019	년	4	월	3 일
과학기술	대 학	컴퓨터 공학			과
3	학 년	2015211372 2015211386			학번
성 명.	한준호 류성진				

## 목차

1. 프로젝트 개요
2. 개발배경
3. 유사앱(프로그램)
4. 스토리 보드
5. 라이브러리
6. 역할 분담
7. 프로젝트 일정
8. 구현

## 1.프로젝트 개요

### 1-1.프로젝트 명

파이썬으로 구성하는 인터프리터 언어

### 1-2.프로젝트 기간

2019.03.27~2019.06.15

### 1-3.프로젝트 목적

파이썬으로 제작하여 자신이 원하는 변수를 제작하거나 문법을 수정하여 컴파일 하는 형식으로 제작

## 2.개발배경

- 기존의 언어들은 컴파일에서 지정된 언어와 문법을 준수하면서 프로그램을 개발해야 하고, 지정된 문법을 통해서만 프로그램을 제작해야하는 고정관념을 깨버린 언어를 개발 해 보고 싶다는 목적에 의해 개발 하게 되었습니다.
- 이 프로그램을 제작시에 확장성 폭이 넓어져 사용자가 각각의 원하는 프로그램 문법을 제작하여 스스로 원하는 스타일의 프로그램을 제작/ 공유를 하여 처리 자체를 다양하게 가능 하도록 할 수 있습니다.
- 이 프로그램을 제작하게 된다면 교육적인 측면에서도 편리하게 교육목적으로 미리 작성해둔 문법 파일을 통해 프로그램 기본을 교육 할 수 있고, 일반 개인이 사용 할 때에 기존에 타 언어의 문법을 제작하여 프로그램을 변환 하지 않고 사용 가능하여 확장성이 넓어집니다.

## 3.유사앱(프로그램)

- JavaScript - 웹기반 프로그래밍 언어
  - 장점 : HTML과 함께 사용하여 개발 속도가 빠르고 웹 브라우저 에서 구동 되기 때문에 운영체제에 제한이 없음
  - 단점 : 소스 코드가 브라우저에 노출되어있고 컴파일 되지 않은 언어이기에 복사하여 그대로 사용이 가능함.
- php - 웹 서버 스크립팅 언어
  - 장점 : 코드가 쉽고 문법이 간단하고 처리속도가 빠르다
- Python - 객체지향 인터프리터 언어

- 장점 : 문법이 간단하여 초보자도 쉽게 배울수 있다. 코드가 쉬운만큼 코딩속도도 빨라져 개발 기간을 단축 시킬수 있다. 텍스트 처리나 빅 데이터 처리가 간편하다.
- 단점 : 문법이 간단한 만큼 문법이 생략되어 있기 때문에 프로그램이 무겁고 동작시간이 느리다.

## 4.스토리 보드

- 1) 컴파일 문법 테이블 제작 [프로그램 기준이 되는 문법을 읽어 테이블을 생성함]
  - a) 어휘분석 -> 문법 주요토큰들을 분석하여 문법식을 기호화 시킴
  - b) 문법데이터 치환 -> 기호화 된 데이터를 기반으로 문법식을 해석 치환과정
    - i) 문법식 오류 -> 문법식의 오류가 있을 경우 오류위치를 기록함
    - ii) 문법식 알림 -> 해당위치의 문법의 에러를 사용자에게 로그로 출력.
    - iii) 중복 문법 -> 중복되는 문법에 대하여 사용자에게 알리고(경고) 로그에 기록
  - c) 치환식 테이블 삽입 -> 치환된 데이터를 문법식 테이블에 삽입
- 2) 어휘분석 [시스템의 주요 토큰(키워드) 테이블 생성]
  - a) 파일읽기 -> 파일을 읽으면서 문법테이블내용을 확인하여 해당하는 문법을 탐색(함수/조건식 우선 탐색)
    - i) 파일을 찾지 못함 -> 사용자에게 파일을 찾지 못함을 알림
    - ii) 문법 오류 -> 문법테이블에서 찾지 못한 문법이 나올 경우 사용자에게 해당하는 오류를 로그에 기록
  - b) 치환 -> 탐색된 데이터를 치환하여 문법테이블 기호를 사용하여 데이터를 변환
  - c) 저장 -> 어휘가 변환된 코드를 분석 테이블에 저장함.
- 3) 내부 코드생성(실행 테이블) -> 토큰을 기호화 시켜서 토큰테이블에 저장
  - a) 파일읽기 -> 파일을 읽어 어휘 분석에서 하지 않은 데이터를 읽음
    - i) 문법 오류 -> 문법테이블에서 찾지 못한 문법이 나올 경우 사용자에게 해당하는 오류를 로그에 기록
    - ii) 데이터 불일치 -> 데이터가 일치하지 않을경우 사용자에게 알리고 로그에 기록
  - b) 탐색 -> 문법테이블에서 특수 문법을 찾음
  - c) 치환 -> 탐색된 데이터를 문법테이블을 참조하여 기호화 코드를 생성
  - d) 저장 -> 치환된 코드를 내부코드에 저장
- 4) 구문 실행 -> 토큰테이블을 읽으면서 프로그램 실행처리

## 5.라이브러리

- 1) [re](#) - 파이썬 정규식 처리 라이브러리
- 2) [pydoc](#) 프로젝트파일 문서화
- 3) 이후 기본라이브러리는 <https://docs.python.org/> 를 참조
- 4) struct Python 문자열로 표현된 C struct 와 Python Value 간 변환 - 파일 바이너리 데이터 저장

## 6.역할 분담

한준호

기초 문법 구현, 내부 프로그램 실행 구현, 각종 데이터 테이블 구현  
류성진  
어휘분석, 실행파일 내부코드 생성, 바이너리 데이터 구현

## 7. 프로젝트 일정

4~6주차 프로그램 문법정의/ 어휘분석 부분 제작  
6~8주차 프로그램 실행 구현/ 실행파일 내부코드 생성  
8~13주차 바이너리 데이터 구현/ 문법 테이블 바이너리 형식 저장 구현  
13~14주차 프로그램 실행 및 디버깅  
15주차 발표

## 8. 구현

### 8-1. 프로그래밍 언어 기본구조

조건식 = [식의 이름] [조건] [식]  
함수 =  
“def” [명사] (변수...) {식}  
“function” [명사] (변수...) {식}  
접근-지정자 변수 [명사] (변수...) {식}  
객체 = “class” [명사] {식}  
과 같이 구현되어 있기 때문에  
최소한의 제작으로 구현이 가능함.

### 8-2. 문법 기본변수(예시)

1. ? = [0~9][?] 0부터 9까지의 상수 / 0~∞개의 상수 표현
2. Var[?~?] = 변수 메모리에 저장되는 데이터 ex) [0~255] 0~255까지 범위데이터  
\*[] 를 생략시 익명의 데이터
3. Ep? = 식
4. Con? = 조건
5. “ = 이름(고유명사지정시작/끝)

6. ( ) = 문법 처리시 실행식

### 8-3. 문법 기본 연산자

- [ ] = L/R SQB - 배열 액세스 | 주소 액세스
- { } = L/R BRACE - 그룹지정자
- + = PLUS
- - = MINUS
- \* = STAR
- / = SLASH
- % = PERCENT
- > = GREATER - 조건식
- >= = EQGREATER
- < = LESS
- <= = EQLESS
- == = EQEQUAL
- = = EQUAL - 우항 연산 결과를 좌항에 넣음 | 좌항에 변수가 없을 경우 출력문
- | = VBAR - or연산
- & = AMPER - and 연산
- ~ = TILDE

기본문법 (예시)

if(조건){식}

Con\_if = "if" "(" Con0 ")" "{" Ep0 "}" (시스템에서 처리)

while(조건){식}

Con\_while = "while" "{" "(" Con0 ")" "{" Ep0 "}" (시스템에서 처리)

\*if문과 while문은 최 하위 문법으로 지정하여 고정함.

if(조건){식}else{식}

Con\_if\_else = Con\_if "else" "{" Ep1 "}" (if(Con\_if){Ep0}else{Ep1})

for(식;조건;식){식}

Con\_for = "for" "(" Ep0 ";" Con0 ";" Ep1 ")" "{" Ep2 "}" (Ep0 while(Con0){Ep1 Ep2})

int 별칭;

Var\_integer = "int" Var ";" (Var)

char 별칭;

Var\_char = “char” Var “,” (Var)

새로운 문법 예

nf(조건){식}

Con\_nf = “nf”“(” Con0 “)”“{” Ep0 “}” (if(Con0){}else{Ep0})

Con\_nf\_else = Con\_nf “else” “{” Ep1 “}” (if(Con0){Ep1}else{Ep0})

#### 8-4. 데이터 테이블 구조

##### 1. 심볼테이블

토큰형태로 읽어들이는 데이터를 저장함

{[변수혹은 함수이름][종류][데이터(배열데이터 혹은 일반  
데이터)]함수인수데이터}[변수|함수의 주소/위치][점 위치(코드테이블)]}

##### 2. 코드 테이블

코드를 관리함

{[종류][문자열][심볼데이터]}

##### 3. 메모리 테이블

{[전역변수][지역변수][함수주소]}

## 메모

### 2.구성

C++언어 프로그램을 사용한 컴파일 언어 처리과정과,  
Python프로그램을 통한 프로그램을 실행하는 형식의 인터프리터 언어 구현.

### 3.구체적인 계획

C언어를 통한 데이터 처리와 데이터를 컴파일 하여 특정 언어로 컴파일 시킨후, 컴파일된 데이터를 기반으로 Python에서 실행시킬 계획.

### 4.

### 4 장점

파이썬의 장점인 언어적 구조와 데이터처리를 통하여 큰 소스 파일도 분석하여 데이터로 처리함과 동시에 컴파일 시켜 전용 실행프로그램으로 생성가능(차후 프로그램변동과정이 없다면 빠른 실행 가능한 프로그램으로 제작)

C언어의 실행을 통하여 프로그램을 실행 시켜, 인터프리터를 동작 시킬 계획.

큰 데이터를 동시 컴파일 하게 되면서 많은 양의 데이터를 처리하게 되고

C언어를 통한 프로그램을 실행하게 되면서 처리를 하여 조금 더 세세한 부분까지 프로그램이 구성되게 된다.