

# 23

## Python 변환을 지원하는, 한글 프로그래밍 언어 코베이직의 구현

소속 정보컴퓨터공학부

분과 B

팀명 금정골사과

참여학생 이석원, 유수민, 김정한

지도교수 우균

### 과제 개요

#### 과제 배경

- 사용되는 프로그래밍 언어 대부분은 영어로 되어있어 한국인이 배우기 어려움
- 이러한 사실은 초보자를 대상으로 하는 코딩교육에서 커다란 걸림돌로 작용
- 한글을 사용하는 초보자가 쉽게 접근할 수 있는 프로그래밍 언어를 제공하고자 함

#### 과제 목적

- 교육용 프로그래밍 언어인 BASIC을 기반으로 한 한글언어 코베이직을 제안
- BASIC 문법 및 특징을 유지하며 한글로 프로그래밍하는 것을 목표로 함
- 코베이직을 파이썬으로 변환을 통해 프로그래밍 활용도를 높이는 것을 목표로 함

### 과제 내용

#### 코베이직 설계

##### 어휘 분석기

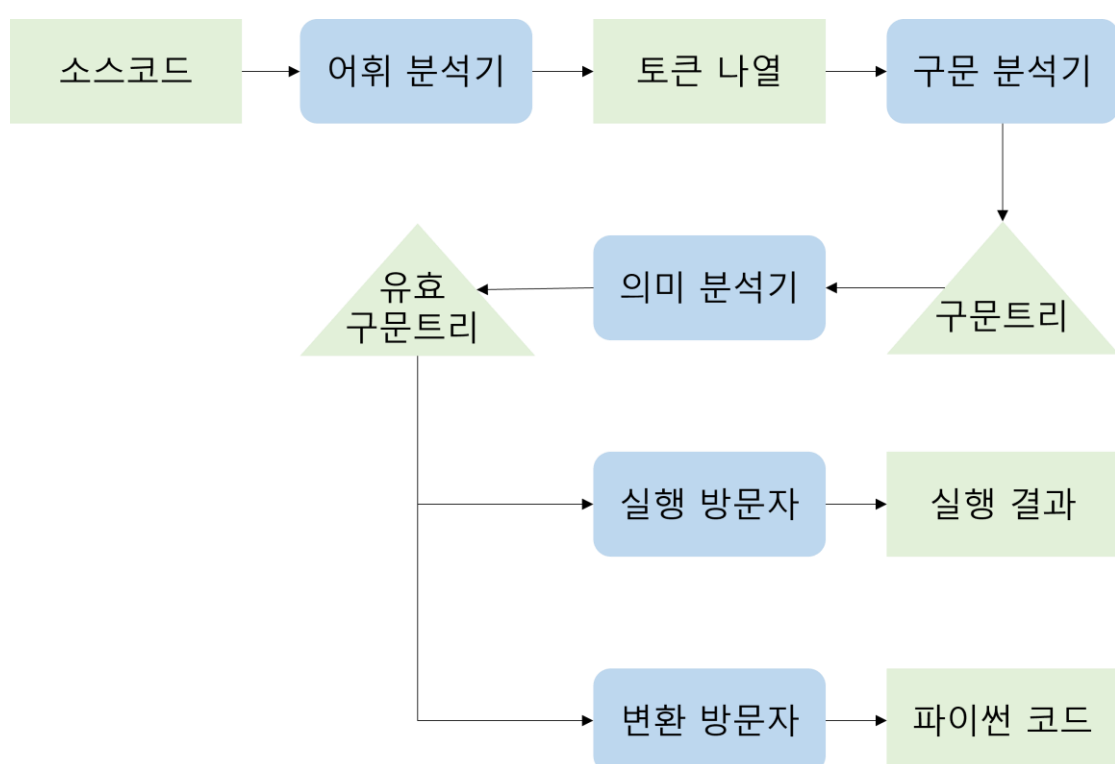
- 소스코드를 토큰으로 분해
- 토큰은 코드의 기본 구성 요소로, 변수, 연산자, 예약어 등과 같은 어휘요소로 구별

##### 구문분석기

- 분해된 토큰들의 구조와 관계를 파악함
- 문법 규칙 준수 확인 및 추상 구문 트리 생성

##### 의미분석기

- 프로그램의 의미를 이해, 검증
- 변수 선언과 사용 등의 검사 수행
- 코드의 의미적 일관성을 보장



#### 코베이직 구현

- C++와 Qt를 이용하여 구현
- 한글 변수명과 함수명을 처리하기 위해 Unicode 문자열 처리 기능 사용
- 키보드 입력 및 출력을 위한 문자 인코딩 문제를 고려
- 한글 주석 및 문법 강조 기능을 구현하여 코드 가독성을 높임

```

10 배열 정수(10)
15 자료 2, -7, 9, 3, -1, 5, -4, 6, 8, -0
20 반복 가 = 0 .. 10
25 읽기 정수(가)
30 출력 정수(가), " ";
35 다음 가
40 출력 "정렬 전"
45 순회 100
50 반복 가 = 0 .. 10
55 출력 정수(가), " ";
  
```

```

def stmt100(next=stmt200):
    print(" | | | ")
    return next

def stmt200(next=stmt300):
    print("+++"+"+++")
    return next
  
```

#### 실험 및 평가

- 테스트벡터(변수대입, 산술연산, 조건문, 반복문, 흐름제어, 함수, 주석 등 수록)를 활용하여 성능 평가
- 한베이식과 비교하였을 때 키워드 수는 20%, 제어명령어 수는 64% 까지 줄어듦을 확인

구분		한베이식	새싹	코베이직
키워드 수	개수	84	21	17
	비교(%)	100	25	20
제어명령어 수	개수	14	19	9
	비교(%)	100	135	64

### 기대 효과

- 코베이직은 한글로 프로그래밍 할 수 있게 하여 한국인이 쉽게 코딩을 배울 수 있게 도움
- 영어가 익숙하지 않은 아이들도 한글로 코딩을 접하면서, 논리적인 사고를 갖추도록 도움
- 간결한 인터페이스가 제공하는 사용성을 통해 도구 자체를 배우는데 시간을 많이 들이지 않고 핵심논리를 배우는데 집중 할 수 있는 환경을 제공
- 코베이직 코드를 python코드로 변환하는 기능을 통해 학습 연계성 강화를 기대