

# ***Experiment in Compiler Construction***

*Phân tích ngữ nghĩa (2)*

**Nguyễn Hữu Đức**

Bộ môn Hệ thống thông tin

Khoa Công nghệ Thông tin

Đại học Bách khoa Hà nội

# Xây dựng bảng ký hiệu trong KPLC

---

- Khởi tạo và giải phóng
- Khai báo hằng
- Khai báo kiểu
- Khai báo biến
- Khai báo hàm, thủ tục
- Khai báo tham số hình thức

# Khởi tạo và giải phóng bảng ký hiệu(1)

---

```
int compile(char *fileName) {  
    ...  
    // Khởi tạo bảng ký hiệu  
    initSymTab();  
    // Dịch chương trình  
    compileProgram();  
    // In chương trình để kiểm tra kết quả  
    printObject(symtab->program, 0);  
    // Giải phóng bảng ký hiệu  
    cleanSymTab();  
    ...  
}
```

# Khởi tạo chương trình

---

- Đối tượng chương trình được khởi tạo tại hàm

```
void compileProgram(void);
```

- Sau khi khởi tạo chương trình phải chuyển vào block chính bằng hàm `enterBlock()`
- Sau khi duyệt xong toàn bộ chương trình, ra khỏi khối bằng hàm `exitBlock()`

# Khai báo hằng

---

- Các đối tượng hằng số được tạo ra và khai báo ở hàm `compileBlock()`
- Giá trị của hằng số được lấy từ quá trình duyệt giá trị hằng qua hàm

`ConstantValue* compileConstant(void)`

- Nếu giá trị hằng là một định danh hằng, phải tra bảng ký hiệu để lấy giá trị tương ứng
- Sau khi duyệt xong một hằng số, phải đăng ký vào block hiện tại bằng hàm `declareObject`

# Khai báo kiểu người dùng định nghĩa

---

- Các đối tượng kiểu được tạo ra và khai báo ở hàm `compileBlock2()`

- Kiểu thực tế được lấy từ quá trình duyệt kiểu bằng hàm

`Type* compileType(void)`

- Nếu gặp định danh kiểu thì phải tra bảng ký hiệu để lấy kiểu tương ứng
- Sau khi duyệt xong một kiểu người dùng định nghĩa, phải đăng ký vào block hiện tại bằng hàm `declareObject`

# Khai báo biến

---

- Các đối tượng biến được tạo ra và khai báo ở hàm `compileBlock3()`
- Kiểu của biến được lấy từ quá trình duyệt kiểu bằng hàm

`Type* compileType(void)`

- Lưu trữ phạm vi hiện tại vào danh sách thuộc tính của đối tượng biến để phục vụ mục đích sinh mã sau này
- Sau khi duyệt xong một biến, phải đăng ký vào block hiện tại bằng hàm `declareObject`

# Khai báo hàm

---

- Các đối tượng hàm được tạo ra và khai báo ở hàm `compileFuncDecl()`
- Các thuộc tính của đối tượng hàm sẽ được cập nhật bao gồm:
  - danh sách tham số: `compileParams`
  - kiểu dữ liệu trả về: `compileType`
  - phạm vi của hàm
- Lưu ý đăng ký đối tượng hàm vào block hiện tại và chuyển block hiện tại sang block của hàm trước khi duyệt tiếp các đối tượng cục bộ



# Khai báo thủ tục

---

- Các đối tượng thủ tục được tạo ra và khai báo ở hàm `compileProcDecl()`
- Các thuộc tính của đối tượng thủ tục sẽ được cập nhật bao gồm:
  - danh sách tham số: `compileParams`
  - phạm vi của thủ tục
- Lưu ý đăng ký đối tượng thủ tục vào block hiện tại và chuyển block hiện tại sang block của hàm trước khi duyệt tiếp các đối tượng cục bộ

# Khai báo tham số hình thức

---

- Các đối tượng tham số hình thức được tạo ra và khai báo ở hàm `compileParam()`
- Các thuộc tính của đối tượng tham số hình thức bao gồm:
  - Kiểu dữ liệu cơ bản
  - Tham biến (PARAM\_VALUE) hoặc tham trị (PARAM\_REFERENCE)
- Lưu ý: đối tượng tham số hình thức nên được đăng ký vào đồng thời vào cả thuộc tính `paramList` của hàm/thủ tục hiện tại, cả vào danh sách đối tượng trong phạm vi hiện tại

# Nhiệm vụ thực hành

---

- Tìm hiểu lại cấu trúc của bộ parser (có thay đổi)
- Bổ xung các đoạn code vào những hàm có đánh dấu TODO để thực hiện các công việc đăng ký đối tượng
- Biên dịch và thử nghiệm với các ví dụ mẫu