РАЗРАБОТКА КОМПИЛЯТОРА ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ЯЗЫКА ОБЩЕГО НАЗНАЧЕНИЯ

Исполнитель:
Колмогорцев Егор Николаевич
Научный руководитель:
Корнев Дмитрий Васильевич

Цель работы

- Учебная задача
- Язык общего назначения
- Объектно-ориентированный
- Множественное наследование
- Виртуальные методы
- Неявное приведение типов

Фазы работы компилятора

- Лексический анализ
- Синтаксический анализ
- Семантический анализ
- Генерация промежуточного кода
- Машинно-независимые оптимизации
- Генерация кода
- Машинно-зависимые оптимизации

Фазы работы компилятора

- Лексический анализ
- Синтаксический анализ
- Семантический анализ
- Генерация промежуточного кода

Лексический анализ

- Поток символов -> поток токенов
- Flex
- Регулярное выражение / токен

Flex

```
digit [0-9]
%%
{digit}+\.{digit}+ {
yylval->dval = std::atof(yytext);
return token::DOUBLENUMBER;
}

{digit}+ {
yylval->ival = std::atoi(yytext);
return token::INTNUMBER;
}
```

Синтаксический анализ

- Токены -> AST
- Контекстно-свободные грамматики
- Yacc / Bison
- LALR

Грамматика языка

- выражения
- операторы
- объявления переменных и массивов
- функции
- объявления классов

Bison

```
expr
   : expr PLUS expr
   { $$ = new BinaryOpASTNode("+", $1, $3);

   | expr TIMES expr
   { $$ = new BinaryOpASTNode("*", $1, $3);}
   /* ... */
   | INTNUMBER
   { $$ = new IntNumberASTNode($1); }
   /* ... */
   ;
```

Конфликты, приоритеты

- Неоднозначная грамматика
- Конфликт перенос/свертка
- Решение - приоритеты операторов

```
%left PLUS MINUS;
%left TIMES SLASH;
```

LLVM

- LLVM
  - трехадресный код
  - SSA-представление
  - блоки
- оптимизации
- целевой язык компилятора

Дизайн языка

- переменные
  - строгая типизация
  - lvalue / rvalue
- массивы
- ввод / вывод

Простая программа

```
def main() of int
{
   sum of int;
   sum = 0;
   i of int;
   for(i = 0; i < 10; i = i + 1) {
      sum = sum + i;
   }
   printi(sum);
   return 0;
}
```

ООП

- Множественное наследование
  - Объект - структура из полей
  - При наследовании - родительские структутры + свои поля
- Приведение типов
  - Хранится система типов
  - Смещение указателя на начало обьекта
- Виртуальные методы
  - При создании объекта устанавливаются указатели на таблицу виртуальных методов
  - При наследовании возможно несколько таблиц
  - При вызове upcasting и downcasting

Пример с наследованием

```
class A {
   a of int;
   virtual def a() of int {
      printi(1);
      return 0;
   }
}
```

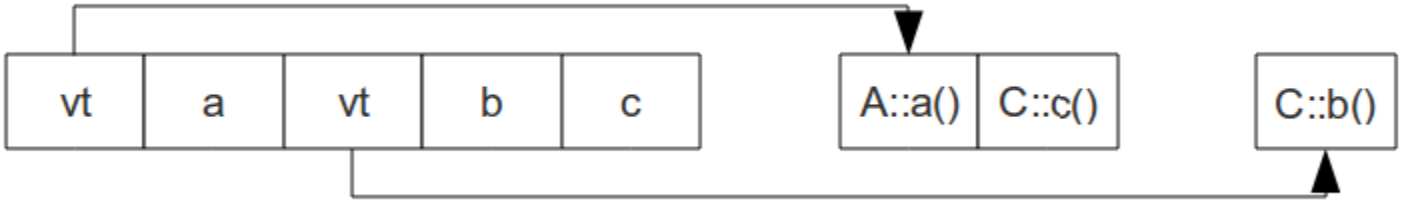
Пример с наследованием

```
class B {
   b of int;
   virtual def b() of int {
      printi(2);
      return 0;
   }
}
```

Пример с наследованием

```
class C : A, B {
   c of int;
   virtual def b() of int {
      printi(3);
      return 0;
   }
   virtual def c() of int {
      printi(4);
      return 0;
   }
}
```

Пример с наследованием




```
def main() of int
{
   c of C;
   b of B;
   b = c;
   b->b();
   return 0;
}
```