

# **MyLang Compiler**

——编译原理大作业



小组成员：

5120379023 陈龙

5120379064 丁卓成

5120379074 彭博

# 整体架构

- ◆ 使用flex和bison进行词法和语法分析
- ◆ 生成LLVM汇编代码，可以用ll解释器解释执行或编译成本地代码执行
- ◆ 语法树的生成和后端汇编生成用C++实现
- ◆ 前台界面用QT实现

# 基本功能

——可以编译并运行快速排序和八皇后问题的程序

- ◆ 实现变量，包括函数参数和局部变量，两者不可重名
- ◆ 实现基本类型（整数和布尔类型）的运算、赋值，对于赋值，表达式区分左值和右值，右值不可被赋值
- ◆ 实现数组类型，包括多维数组，数组不可赋值与被赋值，也不可为函数返回值，数组作为函数参数时按引用传递
- ◆ 实现函数调用及返回，函数名可以和参数或局部变量相同而不冲突
- ◆ 实现控制结构，包括if elif else, while do, repeat until和foreach do
- ◆ 实现print语句，它会被编译为printf函数调用

# 类相关功能

——可以编译并运行**example**程序

- ◆ 实现支持单继承（但不是单根继承）的类型，并有相应的类型检查
- ◆ 类型可以作为数组成员、函数返回值和参数，作返回值和参数时按值传递
- ◆ 实现类的成员变量和成员函数的访问（“.”操作符），在类作用域内可直接访问该类成员
- ◆ 成员变量和成员函数不可重名，但可以和成员函数的参数及局部变量重名
- ◆ 若发生重名，成员变量和函数将被覆盖，需使用this.v的方式访问类的成员。this不可当做指针使用，只能用来取类的成员

# 类相关功能

——可以编译并运行**example**程序

- ◆ 父类通过 `%class.B = type { %class.A, i32 }` 的方式实现，类型无成员变量时定义为 `type { i8 }`
- ◆ 子类的方法会覆盖父类的方法。由于我们没有提供super关键字，子类的成员不可与父类成员重名，子类也无法访问到被覆盖的父类方法
- ◆ 类型成员函数的调用，在汇编中通过传递一个this指针实现，调用父类未被覆盖的函数时，先获得类型内部父类型的this指针，再调用相应函数（未实现static方法）

# 扩展功能

- ◆ 支持字符串，可以通过`print "hello world\n"`实现hello world程序，但字符串只能用于print语句，而不是一个可操作的类型，它也不会自动转型成布尔类型的true
- ◆ 实现了逻辑操作符（`or`，`and`）的部分执行语义，如`a() or b()`中若`a()`的值已经为true，则即使`b()`可能有副作用也不会执行，若为`b() or yes`，则仍会执行`b()`
- ◆ 实现了简单的控制流分析，对于无法执行到的代码不会生成，对于返回值非空的函数若控制流可能抵达其末尾而在其末尾没有return语句，将会报错

Thank You !