Интерпретатор языка Mython

В предыдущем блоке вы познакомились с таблицей виртуальных методов и тем, как с помощью неё реализуется динамический полиморфизм в C++. Важной особенностью наших курсов является закрепление теоретических знаний на практике. Нам было нелегко придумать, на какой практической задаче можно закрепить понимание механизма таблицы виртуальных методов. Всё-таки это деталь реализации и непосредственно ею пользоваться в повседневной практике не приходится. И тут мы вспомнили, что один из лучших способов что-то понять и освоить — это сделать самому! Поэтому далее мы предлагаем реализовать на C++ интерпретатор языка программирования mython, который представляет собой подмножество Python'a (mython — это mini-Python). В языке mython есть классы и наследование, а все методы классов — виртуальные, так что в процессе реализации интерпретатора вам самостоятельно придётся реализовать таблицу виртуальных методов :)

Далее наша работа будет построена следующим образом. Ниже приведено полное описание языка программирования mython, который представляет собой подмножество Python'a. Затем, после этого материала, следует видео лекция, рассказывающая об основных принципах построения трансляторов. Наконец, после лекции идут две задачи по программированию. В первой из них вам предстоит реализовать лексический анализатор языка mython, а во второй — его интерпретатор.

Итак, поехали! Мы говорили, что «Чёрный пояс» ставит своей целью закрепить и развить ваше умение применять C++ на практике, и сейчас вас ждёт очень много практики :)

Описание языка mython

Числа

В языке mython используются только целые числа. С ними можно выполнять обычные арифметические операции: сложение, вычитание, умножение, деление (целочисленное).

Строки

Строковая константа в языке mython — это последовательность произвольных символов, размещающаяся на одной строке и ограниченная либо символами ", либо символами '. Примеры строк в mython:

- "hello"
- 'world'
- 'long string with a double quote " inside'
- "another long string with single quote ' inside"

- '', "" — пустые строки

Строки в языке mython являются неизменяемыми.

Логические константы и None

Помимо строковых и целочисленных значений язык mython поддерживает логические значения True и False, а также специальное значение None, которое является аналогом nullptr в C++. Обратите внимание, что в отличие от C++ логические константы пишутся с большой буквы.

Идентификаторы

Идентификаторы в mython используются для обозначения имён переменных, классов и методов (см. далее). Идентификаторы в mython формируются точно так же, как и в большинстве других языков программирования: они начинаются со строчной или заглавной латинской буквы либо с символа подчёркивания, за которым следует произвольная последовательность, состоящая из цифр, букв и символа подчёркивания.

Примеры правильных идентификаторов: x, _42, do_something, int2str. Примеры неправильных идентификаторов:

- 4four — начинается с цифры
- one;two — содержит символ не являющийся цифрой, буквой или знаком подчёркивания

Классы

Также в языке mython есть возможность определить свой тип, создав класс. Так же, как и в C++, класс имеет поля и методы, однако в отличие от C++ поля не надо объявлять заранее. Объявление класса начинается с ключевого слова class, за которым следует идентификатор имени и объявление методов класса.

Важные отличия классов в mython от классов в C++:

- специальный метод __init__ играет роль конструктора — он автоматически вызывается при создании нового объекта класса;
- неявным параметром всех методов является специальный параметр self; он является аналогом указателя this в C++ и ссылается на текущий объект класса;
- поля не объявляются заранее, а добавляются в объект класса при первом присваивании, как следствие обращения к полям всегда надо начинать с self., чтобы отличать их от локальных переменных;
- все поля объекта являются публичными.

Новый объект класса создаётся так же, как в C++, — указанием имени класса, за которым в скобках идут параметры, передаваемые методу __init__. Например, Rect(10, 5) — это новый

объект класса Rect, при вызове метода __init__ параметр w будет иметь значение 10, а параметр h — 5.

Типизация

В отличие от C++ mython — это язык с динамической типизацией, то есть тип каждой переменной определяется во время исполнения программы и может меняться в ходе её работы. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». Как следствие при первом использовании переменной не надо указывать её тип.

Операции

В языке mython определены:

- арифметические операции для целых чисел (деление выполняется нацело)
- операция конкатенации строк; пример — s = 'hello, ' + 'world'
- операции сравнения строк и целых чисел на «равно/не равно», «больше/меньше», «больше или равно/меньше или равно»; при этом сравнение строк выполняется лексикографически

Функция str

Функция str преобразует переданный ей аргумент в строку. Если аргумент является объектом класса, она вызывает у него специальный метод __str__ и возвращает его результат. Если метода __str__ в классе нет, она возвращает строковое представление адреса объекта в памяти. Примеры:

- str('Hello') вернёт строку Hello
- str(100500) вернёт **строку** 100500
- str(False) вернёт **строку** False
- str(Rect(3, 4)) вернёт адрес объекта в памяти, например '0x2056fd0'

Команда print

Специальная команда print принимает набор аргументов, разделённых запятой, печатает их в стандартный вывод и дополнительно выводит перевод строки.

Обратите внимание, что команда print вставляет пробел между выводимыми значениями. Если в команду print не передать аргументов, она просто выведет перевод строки.

Для преобразования каждого своего аргумента в строку команда print вызывает для него функцию str. Таким образом команда print Rect(20, 15) выведет в stdout строку Rect(20x15).

Условный оператор

условие> — это произвольное выражение, за которым следует символ двоеточия. Если условие истинно, то выполняются действия под веткой if, если ложно — действия под веткой else. Ветка else может отсутствовать. <условие> может содержать сравнения, а также логические операции and, or и not. Истинность условия определяется в зависимости от типа значения, являющегося результатом его вычисления:

- если результатом вычисления условия является значение логического типа, то для проверки истинности условия используется именно оно (например, if x > 0:, if s != 'Jack' and s != 'Ann':)
- если результатом вычисления условия является число, то условие истинно тогда и только тогда, когда это число не равно нулю (как в C/C++), например, if x + y:
- если результатом вычисления условия является строка, то условие истинно тогда и только тогда, когда эта строка имеет ненулевую длину
- если результатом вычисления условия является объект класса, то условие истинно
- если результатом вычисления условия является None, то условие ложно

Обратите внимание, что действия в ветках if и else набраны с отступом в два пробела. В отличие от C++, в котором блоки кода обрамляются фигурными скобками, в языке mython команды объединяются в блоки с помощью отступов. Один отступ равен двум пробелам

Наследование

В языке mython у класса может быть один родительский класс. Если он есть, он указывается в скобках после имени класса и до символа двоеточия.

Наследование в mython работает так же, как и в C++ — все методы родительского класса становятся доступны классу-потомку. При этом все методы являются публичными и виртуальными.

Методы

Команда return завершает выполнение метода из возвращает из него результат вычисления своего аргумента. **Если исполнение метода не достигает команды return, метод возращает None**.

Семантика присваивания

Как было сказано выше, mython — это язык с динамической типизацией, поэтому операция присваивания имеет семантику не копирования значения в область памяти, а связывания имени переменной со значением. Как следствие, переменные только ссылаются на значения, а не содержат их копии. Говоря терминологией C++, можно сказать, что все переменные в mython являются указателями (аналогом nullptr является значение None). Таким образом, код ниже выведет "2", так как переменные x и y ссылаются на один и тот же объект.