

The background is a dark blue gradient with a starry texture. On the left side, there are several overlapping circular elements. A prominent one is a large arc with a scale from 140 to 260 in increments of 10. Other circles are partially visible, some with dashed lines and arrows, suggesting a technical or scientific theme.

# LUA

ПОДГОТОВИЛ: СТУДЕНТ ИО-15 КОЧЕТКОВ С.С.

НАУЧНЫЙ РУКОВОДИТЕЛЬ: ИСАЕВА Г.Н.

# ОСНОВНОЕ

Lua (лу́а, с порт. — «луна») — скриптовый язык программирования, разработанный в подразделении Tecgraf (*Computer Graphics Technology Group*) Католического университета Рио-де-Жанейро (Бразилия). Интерпретатор языка является свободно распространяемым, с открытыми исходными текстами на языке Си.

По возможностям, идеологии и реализации язык ближе всего к JavaScript, однако Lua отличается более мощными и гораздо более гибкими конструкциями. Хотя Lua не содержит понятия класса и объекта в явном виде, механизмы объектно-ориентированного программирования, включая множественное наследование, легко реализуются с использованием *метатаблиц*, которые также отвечают за перегрузку операций и т. п. Реализуемая модель объектно-ориентированного программирования — прототипная (как и в JavaScript).



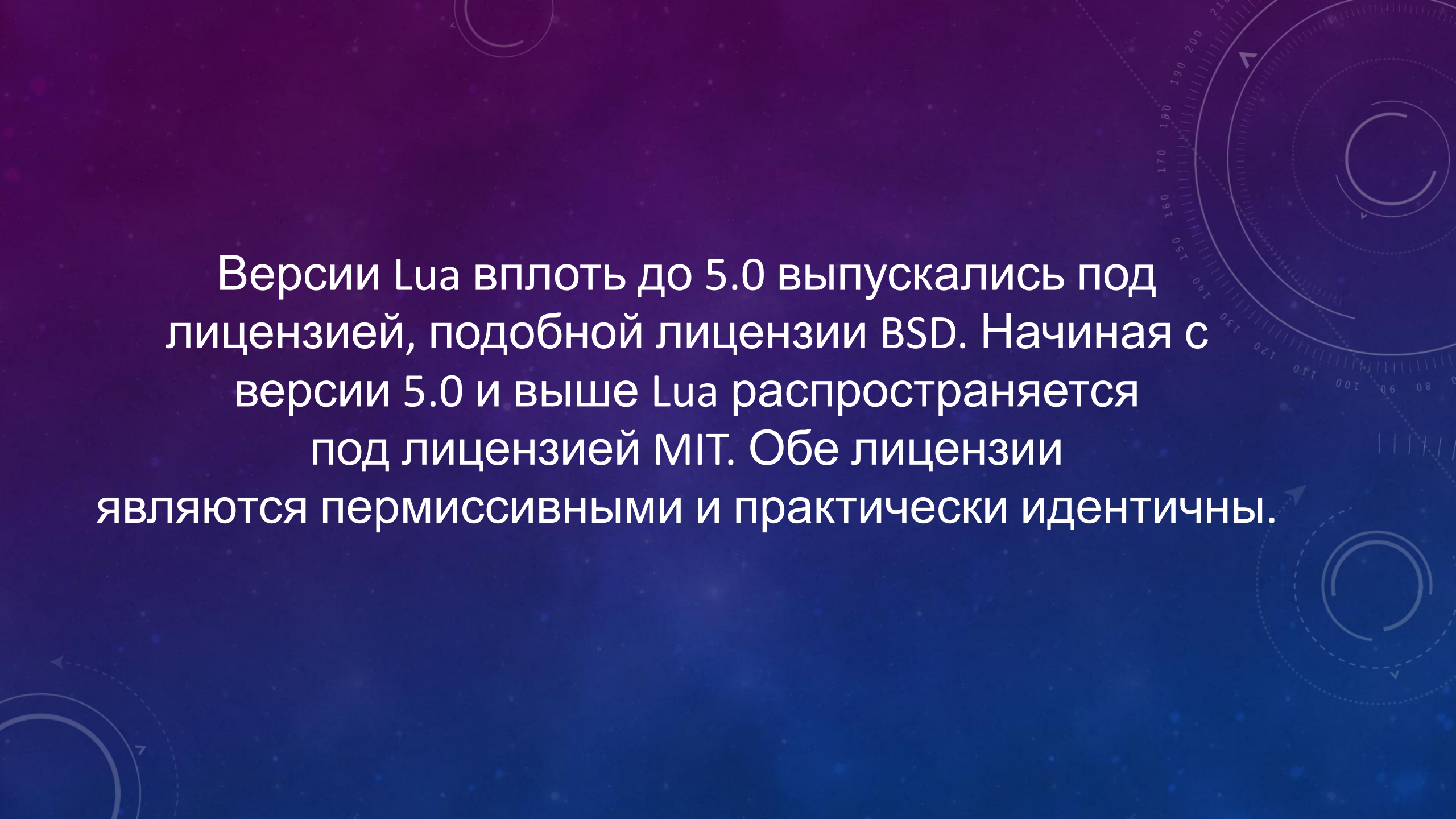
Язык широко используется для создания тиражируемого программного обеспечения (например, на нём написан графический интерфейс пакета Adobe Lightroom). Также получил известность как язык программирования уровней и расширений во многих играх (в том числе World of Warcraft и Garry's Mod) из-за удобства встраивания, скорости исполнения кода и лёгкости обучения.

Lua разработана подразделением Tecgraf (группа технологий компьютерной графики) Католического университета Рио-де-Жанейро в Бразилии, история языка ведёт отсчёт с 1993 года. Авторы языка — Роберто Иерусалимский, Луиш Энрике ди Фигейреду (*Luiz Henrique de Figueiredo*) и Валдемар Селиш (*Waldemar Celes*). Lua распространяется свободно, с открытыми исходными текстами на языке Си.

Историческими родителями Lua были языки конфигурирования и описания данных *SOL* (Simple Object Language) и *DEL* (Data-Entry Language). Они были независимо разработаны в Tecgraf в 1992—1993 годах для добавления некоторой гибкости в два отдельных проекта (оба были интерактивными графическими приложениями для конструкторских нужд в компании Petrobras). В *SOL* и *DEL* отсутствовали какие-либо управляющие конструкции, и Petrobras чувствовал растущую необходимость в добавлении к ним полноценного программирования.



Lua 1.0 была спроектирована таким образом, что её конструкторы объектов, тогда чуть отличающиеся от текущего лёгкого и гибкого стиля, включали в себя синтаксис языка SOL (отсюда название Lua: по-португальски *sol* — «солнце», *lua* — «луна»). Управляющие конструкции Lua в основном заимствованы из Modula (*if*, *while*, *repeat/until*), хотя на них также повлияли CLU (параллельное присваивание, множественное возвращаемое значение функции как более простая альтернатива вместо передачи параметров по ссылке или явных указателей), C++ («отличная идея объявлять локальные переменные лишь тогда, когда они нужны»), SNOBOL и AWK (ассоциативные массивы). Создатели Lua также признают, что единый вездесущий механизм структурирования данных в LISP и Scheme (связный список) оказал большое влияние на их решение о выборе таблиц в качестве основной структуры данных для Lua.

The background features a dark blue gradient with a subtle pattern of white stars and faint technical diagrams. On the right side, there are several circular gauges or dials with numerical scales (e.g., 160, 170, 180, 190, 200, 210) and arrows. On the left, there are partial circular diagrams with arrows. The overall aesthetic is technical and futuristic.

Версии Lua вплоть до 5.0 выпускались под лицензией, подобной лицензии BSD. Начиная с версии 5.0 и выше Lua распространяется под лицензией MIT. Обе лицензии являются перmissive и практически идентичны.



- Lua обычно называют мультипарадигменным языком. Он обеспечивает небольшой набор базовых механизмов, которые могут быть расширены для решения различных задач, а не набор сложных жёстких спецификаций, обеспечивающих программирование в единой парадигме.
- Так, в Lua отсутствует явная поддержка наследования, однако оно легко реализуется с помощью *метатаблиц*.
- В общем, Lua стремится обеспечить гибкие метафункции, которые могут быть расширены по мере необходимости, а не поставлять набор функций, специфичных для одной парадигмы программирования. Как результат, основа языка легка и легко адаптируема к большинству приложений.

Lua — это динамически типизированный язык, предназначенный для использования в качестве расширения или в качестве скриптового языка, и при этом достаточно компактный, чтобы поместиться на различных исполняющих платформах. Он поддерживает небольшое количество структур данных, такие как логические значения, числа (по умолчанию — двойной точности с плавающей точкой), а также строки. Типичные структуры данных, такие как массивы, наборы, списки и записи, могут быть представлены с использованием родной структуры данных Lua — таблицы, которая по существу является гетерогенным ассоциативным массивом.

Lua реализует небольшой набор дополнительных функций, такие как сборщик мусора, принудительное приведение типов и прочие. Включая минимальный набор типов данных, Lua пытается найти баланс между мощностью и размером.

Lua представляет собой язык с динамическим определением типов данных. Переменная языка может содержать значения любого типа. Все значения в Lua могут храниться в переменных, использоваться в качестве аргументов при вызове функций и возвращаться в виде результата их выполнения.



# В LUA ВОСЕМЬ ОСНОВНЫХ ТИПОВ ДАННЫХ:

- nil (неопределенный)
- boolean (логический)
- number (числовой, с версии 5.3 добавлена возможность явного определения целого или вещественного формата числа)
- string (строковый)
- function (функция)
- userdata (пользовательские данные)
- thread (поток)
- table (таблица)

# ПРИМЕРЫ ПРОГРАММ

- `print("Hello Word");`
- `--` Однострочный комментарий
- `--[[` Многострочный
- `Комментарий]]`

# ВЕТВЛЕНИЕ+ВЫЧИСЛЕНИЕ ФАКТОРИАЛА

```
function factorial(n)
  if n == 0 then
    return 1
  else
    return n * factorial(n - 1)
  end
end
```



## ЦИКЛ СО СЧЕТЧИКОМ

```
for i = 1,5 do  
    -- Statements  
end
```

Как и многие интерпретируемые языки программирования, реализация *Lua* имеет отдельно компилятор с исходного языка в исполняемый байт-код и виртуальную машину для исполнения сгенерированного байт-кода. Причём байт-код — это не команды стековой машины, а команды некоего виртуального процессора с несколькими регистрами, что повышает эффективность исполнения. В стандартной виртуальной машине *Lua* используется распределение памяти со сборкой мусора (аналогично Java или .NET).

*Lua* использует единый строковый пул, что позволяет снизить расходы памяти на хранение строк.

# СУДЬБА ЯЗЫКА

В настоящее время используется в различных проектах, где требуется встроить достаточно быстрый и нетрудный в освоении скриптовый язык программирования — например, в разработке игр, где *Lua* часто используется в качестве прослойки между игровым движком и данными для написания сценариев поведения/взаимодействия объектов.



- ❑ Первыми в разработку компьютерных игр язык *Lua* внедрила компания LucasArts, начиная с игры Grim Fandango.
- ❑ В 2003 году в результате опроса на сайте GameDev.net *Lua* был признан самым популярным скриптовым языком для разработки игр.
- ❑ Также используется в авиасимуляторе X-Plane, в движке X-Ray для S.T.A.L.K.E.R.
- ❑ Для популярной игры Minecraft созданы модификации ComputerCraft и его более совершенный аналог OpenComputers, которые добавляют компьютеры, программируемые на языке Lua.
- ❑ Зарубежная команда Croteam (разработчики Serious Sam и The Talos Principle) используют Lua в скриптах начиная с версии Serious Engine 3.5.

```
6c0e09e9-99ec-4e26-ba5f-cb3e269023dc 512 1048064 1% /mnt/6c0/
tmpfs 512 65024 1% /tmp/
tmpfs 512 65024 1% /mnt/d36/
openos 148630 0 100% /mnt/4b2/
/# ls
```

```
bin boot etc lib mnt tmp usr
```

```
init.lua
/# ls bin
address.lua alias.lua cat.lua cd.lua
clear.lua components.lua cp.lua date.lua
df.lua echo.lua edit.lua install.lua
label.lua ln.lua ls.lua lua.lua
man.lua mkdir.lua more.lua mount.lua
mv.lua primary.lua pwd.lua reboot.lua
redstone.lua resolution.lua rm.lua set.lua
sh.lua shutdown.lua umount.lua unalias.lua
unset.lua uptime.lua useradd.lua userdel.lua
which.lua
```

```
lua> lua
Lua 5.2.3 Copyright (C) 1994-2013 Lua.org, PUC-Rio
Enter a statement and hit enter to evaluate it.
Prefix an expression with '#' to evaluate it.
Press ctrl+q to quit the interpreter.
lua> file = io.open('mnt.modem', 'w')
lua> file:write('modem')
lua> file:close()
lua> os.sleep(1)
```

# СПАСИБО ЗА ВНИМАНИЕ!

