

[Pony](#) — объектно-ориентированный язык программирования, построенный в первую очередь на модели акторов и прозрачной конкурентности. В его дополнительных плюсах трутся такие понятия, как — «Open-source, производительность, интересные идеи». Основной упор идет на многоядерный системы, современную разработку без мыслей о низком уровне.

Возможности:

- Типобезопасность. Подтвержденная математическим бекграундом.
- Безопасная работа с памятью. Хотя это и является следствием из типобезопасности, но нельзя упускать данный пункт. В нем вы не найдете повисших указателей, переполнение буфера или в самом частом случае — вам не придется знать, что такое null.
- Безопасная работа с экзепшенами. Все они определяются на семантическом уровне и никаких рантайм неожиданностей.
- Скажем нет гонкам данных! Никаких атомиков, мьютексов или таймеров. Система типов с проверками во время компиляции дают высокую гарантию сего понятия. Пишите конкурентный код и не переживайте за остальное.
- Никаких дедлоков. Следствие из отсутствия любых операций с блокировками.

Система типов:

- Класс, стандартен из ООП мира. Поля, конструкторы и функции в вашем распоряжении.
- Актор, идентичен классу, но имеет возможности определения асинхронных функций.
- Прimitив, тоже идентичен классу с 2-я различиями. Он не может иметь полей и в процессе работы программы создается только 1 инстанс на определенный вами примитив. Используется для многих вещей, но чаще, как особый вид дженерика с возможностью в None тип, чтобы проверять приходящие из FFI мира значения, не равняется ли оно пустоте.
- Трейт и интерфейс (Являются подтипами, однако определяются так же, как и класс). Многие ЯП имеют только 1 из них одновременно, но в Пони они задействованы оба. Различие в том, что трейт является условной проверкой принадлежности, а интерфейс проверяет соответствие структурно.

- Алиасы для типов. Не буду говорить тут про них много, мощная система и требует самостоятельного изучения.
- Кортеж есть последовательность типов, которые могут быть определены в одной переменной.
- Юнион очень похож на кортеж, только используется для обобщения возможных типов. Особый вид дженерика.
- Пересечение почти является противоположностью юниона и может описывать одно значение для нескольких типов одновременно. В примере ниже видно, как карта может содержать одновременно ключ двух разных типов.
- Все выше описанные выражения типов могут комбинироваться

Стандартные выражения:

- Это всем очень хорошо знакомые — «Переменные, знаки операций, проверки, циклы, методы, экзепшены, etc..». Оставляю на самостоятельное изучение.

Привлекательность языка

Сборщик мусора

На борту мы имеем очень крутой GC, который является полностью конкурентным и не имеет Stop the World. На самом деле там их 2, один является сборщиком ссылок для каждого созданного актора и один глобальный.

Скорость акторов

Благодаря такому парню, как Дима Вьюков, который внес огромный вклад в lock-free алгоритмы и Go — появилась база, на которую делал упор Pony при разработке общения акторов. И именно поэтому сейчас скорость передачи между акторами достигает 20кк в секунду, а создание может достигать 1кк в секунду.