**6. Интерпретация. Шаблонная JIT компиляция. Оптимизирующая JIT и AOT компиляция.**

*Интерпретация* - пооператорный (покомандный, построчный) анализ, обработка и тут же выполнение исходной программы или запроса (в отличие от компиляции, при которой программа транслируется без её выполнения).

Интерпретация работает по принципу конвейера, возникают конфликты, это приводит к простоям (чтобы уменьшить простои необходимо увеличить линейные участки кода).

В интерпретации поддерживается исполнение супер инструкций — характерных последовательностей инструкций.

*Динамическая(Just in Time) компиляция*

Компиляция IR(Intermediate Representation, Промежуточное представление) в машинный код происходит непосредственно во время работы программы

При вызове незагруженного класса JIT компилятор выполняет частичную компиляцию, а затем загружает нужный класс (не выполняется для super-class).

*Шаблонная компиляция* — сборка бинарного кода из небольших фрагментов кода.

*Оптимизирующая компиляция* — используются различные методы получения более оптимального программного кода при сохранении его функциональных возможностей.

Оптимизирующая JIT-компиляция

* По сравнению с интерпретатором и шаблонным JIT компилятором
  + Достоинства
    - Ускорение выполнения кода в 4 и более раз
    - Производительность кода сопоставимая с компилируемыми языками (C, C++, Fortran, Ada)
  + Недостатки
    - Затраты ресурсов на компиляцию
    - Увеличение времени старта
    - Затруднение отладки пользовательского кода
    - Высокая вероятность ошибки и нарушения семантики пользовательского кода

Виды:

* Peephole-оптимизации рассматривают несколько соседних (в терминах одного из графов представления программы) инструкций, чтобы увидеть, можно ли с ними произвести какую-либо трансформацию с точки зрения цели оптимизации. В частности, они могут быть заменены одной инструкцией или более короткой последовательностью инструкций.
* Локальная оптимизация- рассматривается только информация одного базового блока за один шаг. Так как в базовых блоках нет переходов потока управления, эти оптимизации требуют незначительного анализа (экономя время и снижая требования к памяти), но это также означает, что не сохраняется информация для следующего шага.
* Внутрипроцедурные оптимизации — глобальные оптимизации, выполняемые целиком в рамках единицы трансляции (например, функции или процедуры) При наличии глобальных переменных может быть затруднена.
* Оптимизации циклов
* Межпроцедурные оптимизации

*Ahead of Time(AOT)* — компиляция перед исполнением.

Это — статическая компиляция.

Экономит память, нет подвисаний во время исполнения программы.

Между сессиями часть runtime environment можно сохранить, это экономит память, время запуска программы.

Harmony Jet - наивный шаблонный компилятор.

Harmony Interp - наивный интерпретатор