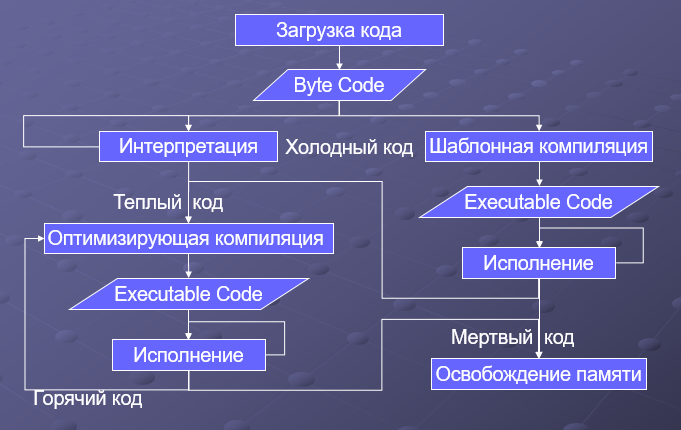
***29. Модель динамической перекомпиляции. Особенности JIT компилятора. Замена кода.***

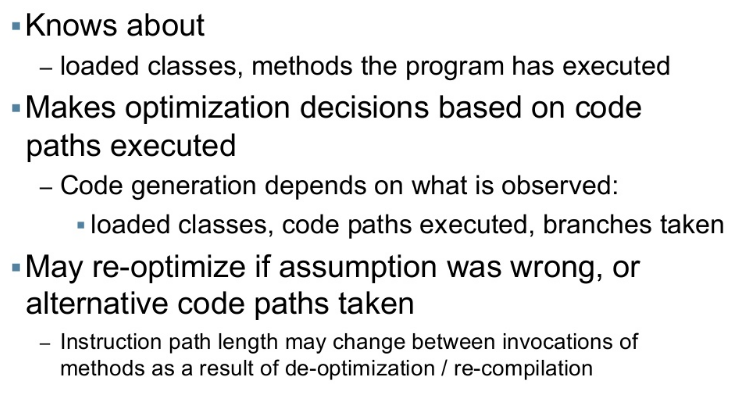
*Модель динамической перекомпиляции*

*Из-за необходимости интерпретации байт-код выполняется значительно медленнее машинного кода сравнимой функциональности, однако он более переносим (не зависит от операционной системы и модели процессора). Чтобы ускорить выполнение байт-кода, используется динамическая компиляция, когда виртуальная машина транслирует псевдокод в машинный код непосредственно перед его первым исполнением (и при повторных обращениях к коду исполняется уже скомпилированный вариант).*



*JIT-компиляция*

[*вики*](https://ru.wikipedia.org/wiki/JIT-%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%86%D0%B8%D1%8F#.D0.9E.D1.81.D0.BE.D0.B1.D0.B5.D0.BD.D0.BD.D0.BE.D1.81.D1.82.D0.B8_.D1.80.D0.B5.D0.B0.D0.BB.D0.B8.D0.B7.D0.B0.D1.86.D0.B8.D0.B8)

**

*Замена кода*

* Ленивая замена кода
  + Новый код исполняется при следующем вызове
  + Требует сохранения предыдущей версии кода если она исполняется в момент завершения компиляции
  + Требуется отслеживать ссылки на старую версию кода для освобождения памяти
  + В случае глубокой подстановки может привести к существенным потерям производительности из-за невозможности заменить долго исполняемый код
* Замена кода на стеке
  + Версия кода заменяется в момент завершения компиляции
  + Требует трансформации параметров на стеке в соответствие с новой версией кода
  + Сложность реализации