Java Virtual Machine (Obfuscation and Java)

Пименов Александр

2004

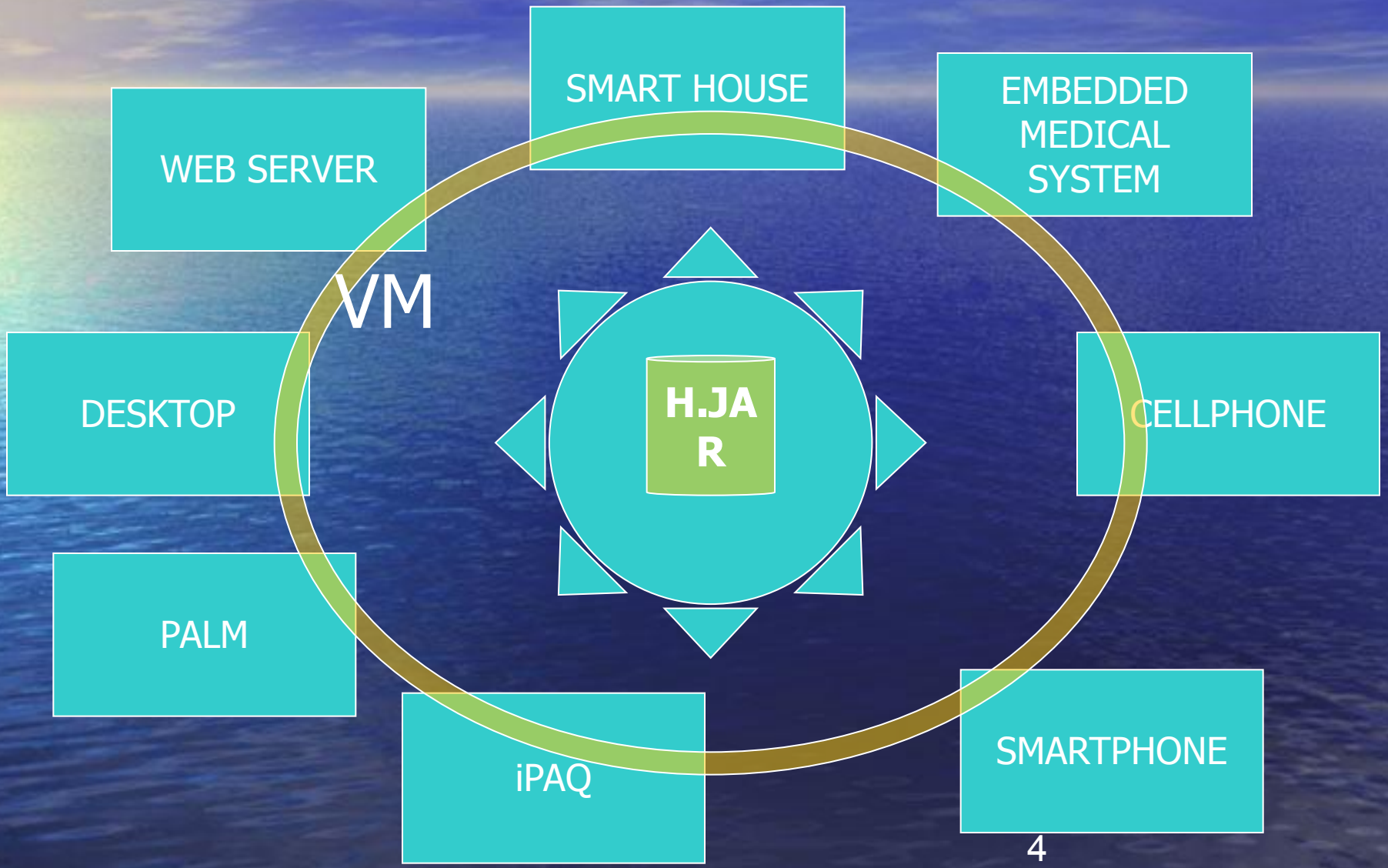
What it is all about?

- Идеология Java
- Classfile
  - Константный пул
- Структура и идеология машины
- Идеология системы команд байт-кода
- Обфускация и Java
- Q&A

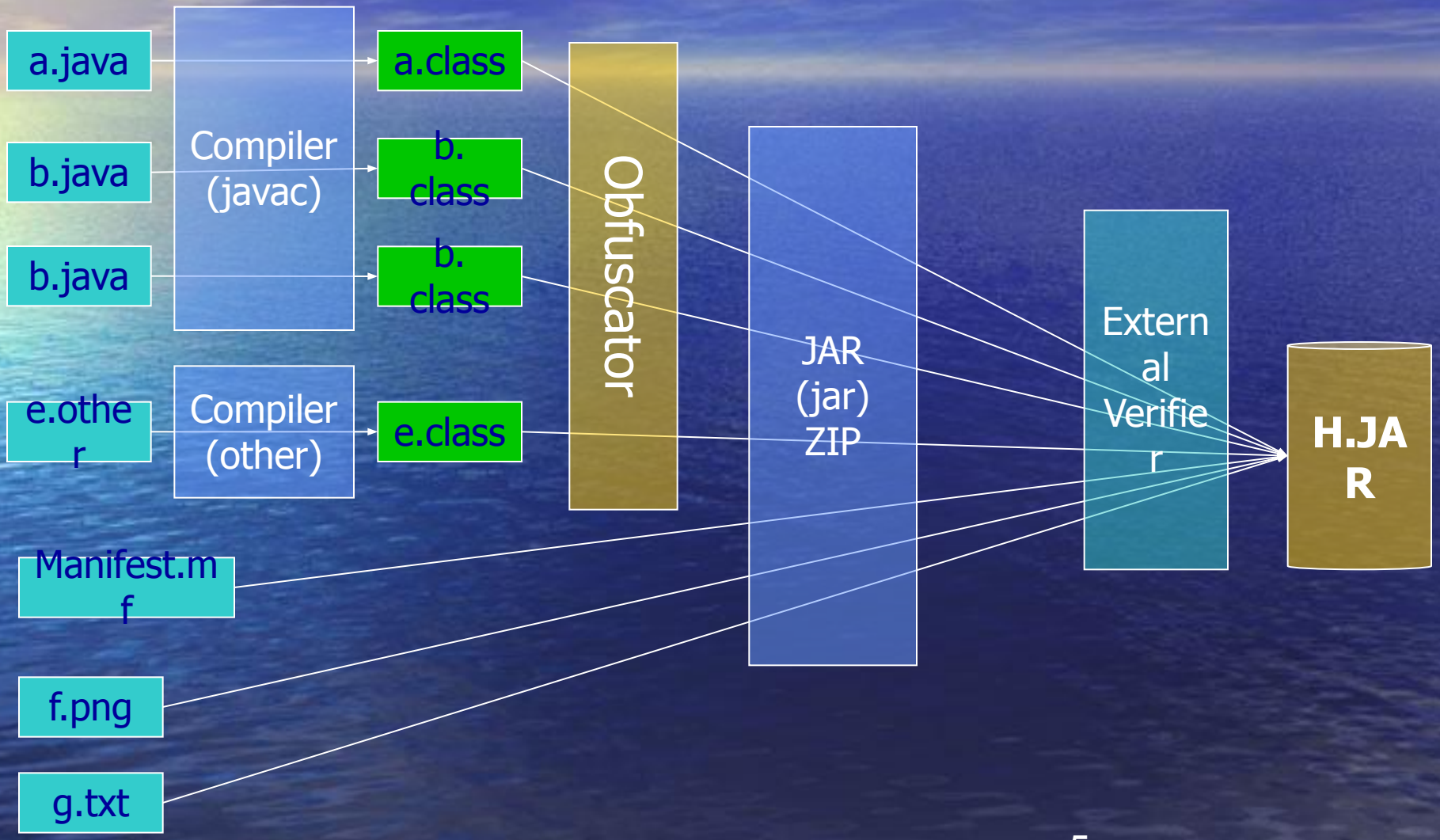
Java Technology (Введение)

- Java предложена фирмой SUN Microsystems
  - 1991 г. James Gosling – язык Oak
  - 1995 г. Изменение названия на JAVA и выпуск первой реализации - JDK 1.0
  - 1998 г. Платформа Java 2 (SE, EE, ME)
- Java код:
  - Переносим
  - Динамичен
  - Предсказуем
  - Объектно-ориентированан

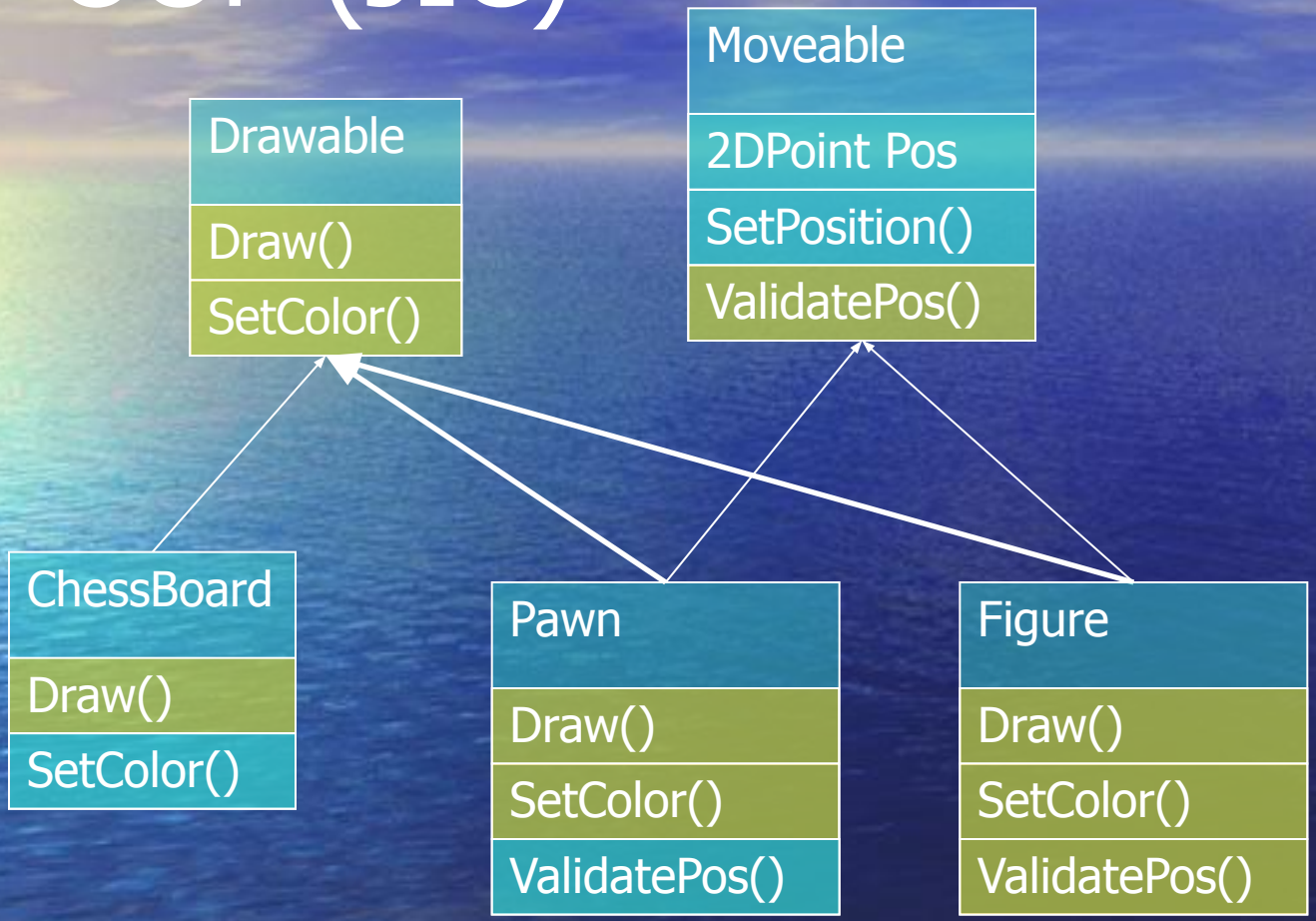
What is this all for?



Путь кода в JAVA



OOP (JIC)


Drawable
Draw()
SetColor()
Moveable
2DPoint Pos
SetPosition()
ValidatePos()
ChessBoard
Draw()
SetColor()
Pawn
Draw()
SetColor()
ValidatePos()
Figure
Draw()
SetColor()
ValidatePos()

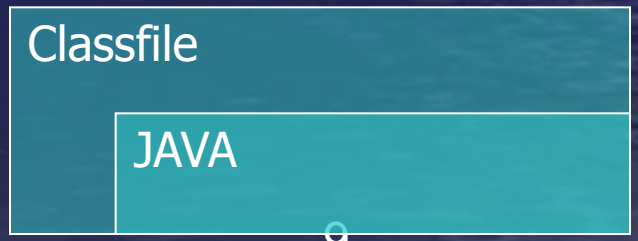
CLASS FILE

Classfile structure (идеалогия)

- Программные компоненты Java распространяются в виде набора classfile-ов (или их архивов JAR)
  - Наследование структуры JAVA
    - Инкапсуляция
  - Полная платформонезависимость
  - Позднее связывание
  - Динамическая загрузка программных компонентов

Classfile structure (метафора)

- Каждый classfile – представляет данные одного класса или интерфейса
- Он не обязан содержать никаких данных о символическом представлении своего класса или интерфейса
- Classfile – это полностью стандартизированный поток байт (big-endian)
- До критического момента classfile – это просто данные



Classfile structure (что содержит)

- Пул констант
- Тип класса (доступ, наследование итп)
- Ссылку на класс-предок (суперкласс)
- Набор поддерживаемых интерфейсов
- Описания полей
  - Атрибуты полей
- Описания методов
  - Атрибуты методов
- Описания атрибутов

Classfile structure (константный пул)

- Набор записей позволяющих разрешать ссылки при связывании и выносить констнты из кода
- Экономит место объединяя одинаковые константы
- Хранит
  - Константы (int, float, long, double, строки UTF8)
  - Ссылки на символические названия внешних объектов (методов, функций, классов)
- Максимальный размер констпула 2^16
- Константный пул разрешают в процессе исполнения

Пример

CONSTANT_Methodref_info

CONSTANT_Class_info

TYPE	10
class_index	12
name_and_type_index	18

TYPE	7
name_index	14

COCONSTANT_NameAndType_info

TYPE	1
length	9
bytes	test/test1

TYPE	12
name_index	22
descriptor_index	26

TYPE	1
length	7
bytes	callIfFailed

CONSTANT_Utf8_info

TYPE	1
length	9
bytes	(IDLtest/test2;)Ljava/lang/Object

Attributes

- SourceFile - описание исходника
- ConstantValue – описание константы
- Code –описание кода
- Exceptions – список исключений кидаемых методом
- InnerClasses – список внутрених классов из других пакетов
- Synthetic – отметка о том что член класса отсутствовал в исходнике
- LineNumberTable – список ссылок на номера строк
- LocalVariableTable – список имен локальных переменных
- Deprecated – отметка о том что член или класс вытеснен (перегружен)

Methods and attributes

```
attribute_info
{
u2 attribute_name_index;
u4 attribute_length;
u1 info[attribute_length];
}
```

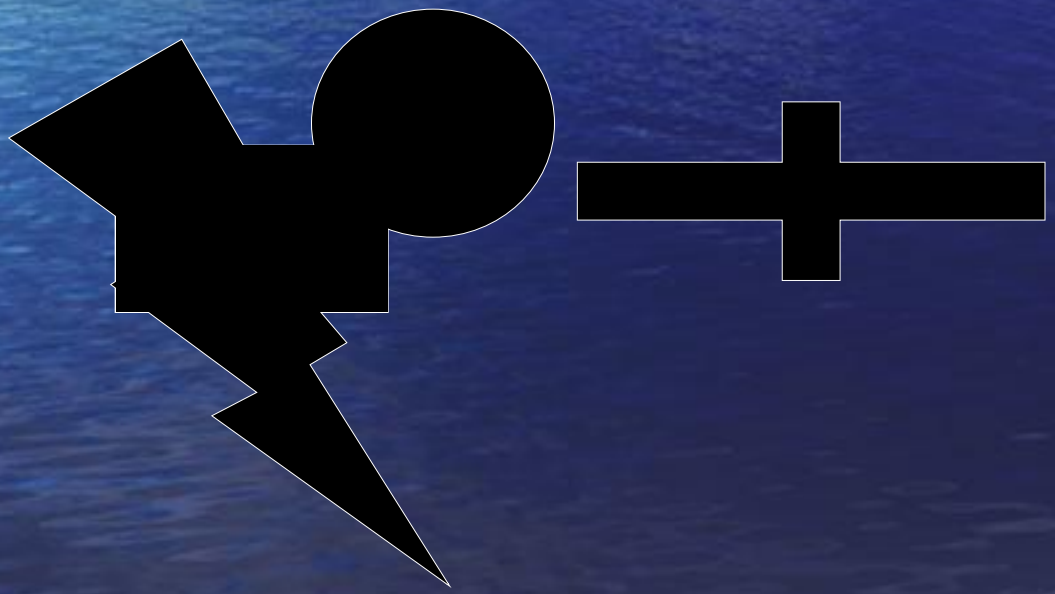
```
method_info
{
u2 access_flags;
u2 name_index;
u2 descriptor_index;
u2 attributes_count;
attribute_info
attributes[attributes_count];
}
```

```
Code_attribute
{
u2 attribute_name_index;
u4 attribute_length;
u2 max_stack;
u2 max_locals;
u4 code_length;
u1 code[code_length];
u2 exception_table_length;
{
        u2 start_pc;
        u2 end_pc;
        u2 handler_pc;
        u2 catch_type;
} exception_table[exception_table_length];
u2 attributes_count;
attribute_info
attributes[attributes_count];
}
```

JVM

Спецификация JVM

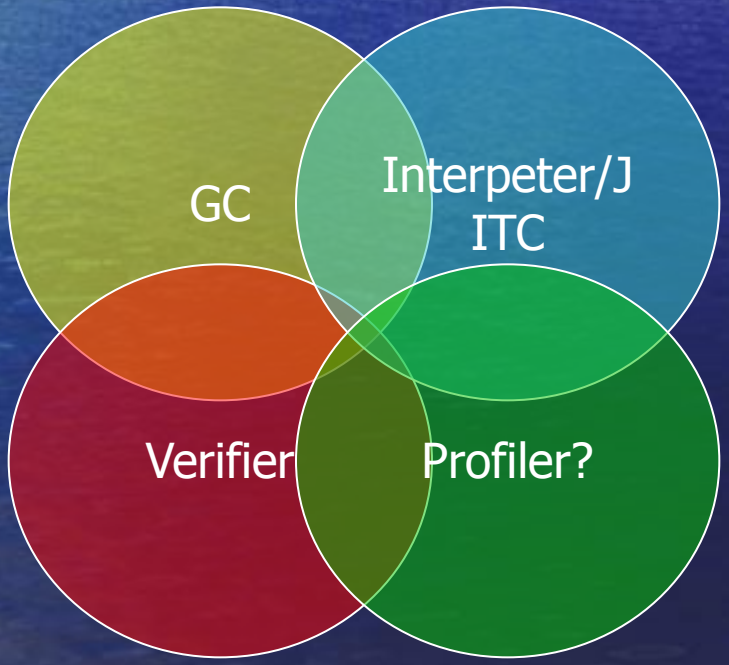
- В всей полноте описывает «что?»
- Не описывает «как?»



JVM Subsystems

Собирает мусор

Исполняет



Ведет статическую предпроверку

Ищет узкие места

JVM
(структура JAVA машины)

- Многопоточность
- Поддержка исключений
- Модель абстрактной памяти построенной на ссылках на объекты
- Автоматическая сборка мусора
- Строгая типизированость предполагается

JVM (структура JAVA машины)

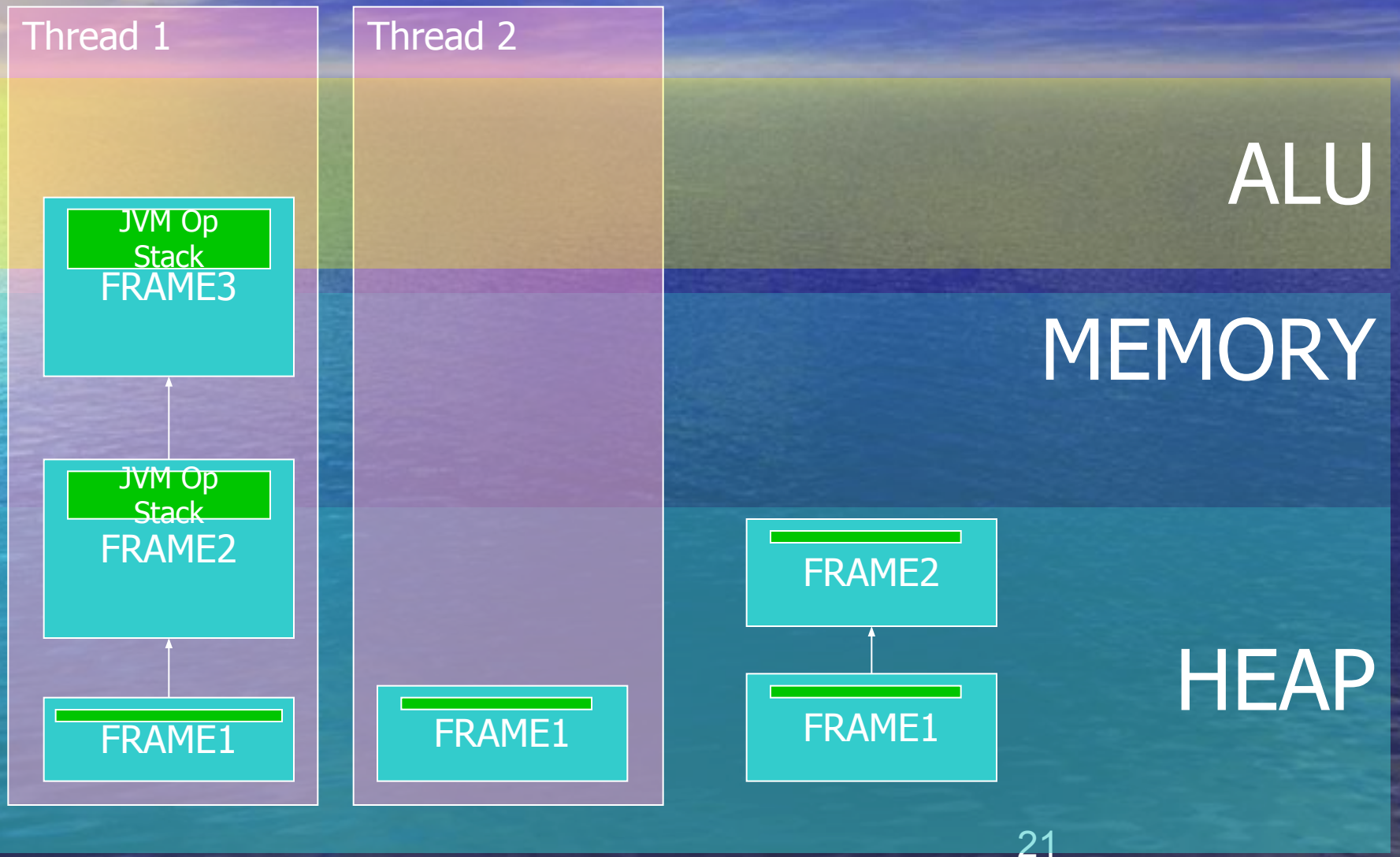
- JVM – абстактная стековая машина с локальными переменными
- Каждый поток в JVM имеет свой стек «вызов» заполняемый фреймами
- JVM оперирует с:
  - Стеком операндов (текушие подсчеты)
  - Локальными переменными + параметрами
  - Статическими переменными

JVM


Runtime constpool
Thread 1
Local Variables
Parameters
JVM Op Stack
Frame1
Local Variables
Parameters
JVM Op Stack
Frame2
Thread 2
Local Variables
Parameters
JVM Op Stack
Frame1
Local Variables
Parameters
JVM Op Stack
Frame2
Local Variables
Parameters
JVM Op Stack
Frame3

JVM

FRAME STACK (call stack)



BYTE code (Концепции)

- Байткоды имеют опкод в 8 бит и расширяются байтово
- Байткоды имеют структуру кеширующю пространство
- Есть свободные байткоды
- Есть сложные и простые байткоды (IS неоднороден)
  - Со всеми вытекающими
- Машина работает с данными размером 8, 32, 64 бита

BYTE code (Типы данных)

- byte - 8-bit знаковое целое
- short - 16-bit знаковое целое
- int - 32-bit знаковое целое
- long - 64-bit знаковое целое
- char - 16-bit без знаковое для представления UNICODE
- float
- double

BYTE code (типы)

- Работа с константами
- Push байт-коды
- Сохранение и загрузка локальных переменных
- Pop байт-коды и работа со стеком
- Арифметические операции
- Операции сравнения и условных переходов, операции вызовов, операции перехода по таблице
- Операция безусловного перехода
- Операции загрузки статических и динамических полей
- Создания новых объектов, массивов, проверка приведения типов
- Броски исключений
- Захват и освобождение мониторов
- Быстрая проверка на ноль
- Длинные переходы и длинные загрузки

Size reduction

iconst_m1

push (byte)-1

iconst_n

n = if (x < 9 && x > 3) then x – 3 else OTHER_OPCODE

push (byte)n

bipush

push x

sipush

push x

ldc

push CONST_POOL[x]

ldc_w

push CONST_POOL[x]

Vierifier

- invoke virtual
  - Метод ДОЛЖЕН БЫТЬ этого класса или класса предка.
- А кто проверит?
  - verifier
  - В runtime

JIT

- Pros vs Cons
  - Time – 2x-40x faster
  - Memory Overhead – 5x-10x
- Compilation
  - JIT Just-In-Time
  - AOT Ahead-Of-Time
  - HI Hotspot Implementations

Garbage collection (общие мысли)

- Мусор в Америке – федеральная собственность
- А некоторые считают – это для склеротиков
- Нет нужды явно освобождать память... машина сделает это сама
- Представьте себе сервер и утечку памяти в нем...
- Если воспринимать исключения как необходимость, надо как необходимость воспринимать и сборку мусора

Obfuscation

Obfuscation and Java

- Pros
  - Она на самом деле необходима
  - Java машина очень распростанена
  - Обфускация экономит место
- Cons
  - Reflection может работать неверно
  - Идеалогия Java против обфускации

Decompilers

- Goto
- Бывает класс «for» и «synchronized» в байт-коде но не бывает таких классов в Java
- Что навсегда потеряно, того уж не вернешь. Старые имена и связи не восстановить, если они уничтожены

Decompilers

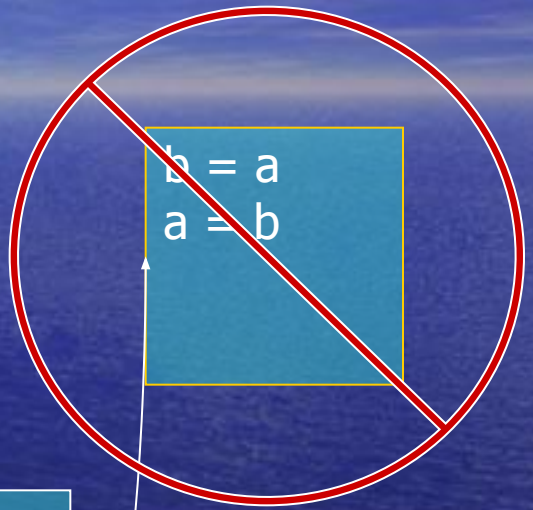
А бывает они даже разваливаются….

```
int tmp;
tmp = a;
a = b;
b = tmp;
```

```
iload_0
istore_2
iload_1
istore_0
iload_2
istore_1
```

Оптимизатор

```
iload_0
iload_1
istore_0
istore_1
```



Obfuscation - Names and profiles (Retroguard)

- GNU GPL
- Скриптовый

Names and profiles (JODE)

- GNU GPL
- Переименование классов, методов и полей
- Удаление отладочной информации
- Удаление «мертвого» кода (классов, методов, полей)
- Оптимизация выделения локальных переменных
- Есть декомпилятор

Names and profiles (SandMark)

- Работа Аризонского университета – выполняет watermarking, tamper-proofing и code obfuscation
- Основан на алгоритмах by Venkatesan, Collberg, Stern, and others

Names and profiles (DashO)

- Commercial
- Sun's choice
- Features
  - Package/Class/Method/Field renaming using our patented Overload-Induction(tm) renaming system
  - Unused Class/Method/Field and constant pool entry removal
  - Advanced Control Flow Obfuscation
  - String Encryption
  - Class and method level optimization to improve JIT performance

Names and profiles

	Сокрытие имен (name mangling)	Изменение потока исполения (code mangling)	Кодирование строк (strings enscription)
Retroguard			?
SandMark	?		?
DashO			
JODE			?

Where is more information available?

- Исчерпывающая информация о Java машине содержится в спецификации SUN и больше мало где:
  - http://sunsite.nstu.ru/java-stuff/vmspec/
- Информация о статистике и classfile
  - ftp://ftp.cs.arizona.edu/reports/2004/TR04-11.pdf
- Различные обфускаторы
  - http://www.retrologic.com/retroguard-docs.html
  - http://sandmark.cs.arizona.edu/publications.html
  - http://www.preemptive.com/products/dasho/Features.html

THE END

Спасибо за внимание

Q & A

Теперь совсем конец