



# СТАТИЧЕСКИЙ АНАЛИЗ ДЛЯ ВСЕХ И ДЛЯ КАЖДОГО

# ОБО МНЕ

2

**Артемий Сартаков**

Закончил ФИТ НГУ в 2017

5 лет в статическом анализе

Преподаю ООП на ФИТе



Я ЛЮБЛЮ  
~~МУЧИТЬ~~  
СТУДЕНТОВ

# ALT+ENTER КОД РЕВЬЮ

- » Берем проект студента
- » Открываем в IDEA
- » Ругаемся на желтый код

# ЧТО С ЭТИМ МЕТОДОМ НЕ ТАК?

5

```
public void checkCollisions(FlappyBird flappyBird) {  
    boolean found = false;  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
    // что-то делаем с found...  
}
```



# ЧТО С ЭТИМ МЕТОДОМ НЕ ТАК?

6

```
public void checkCollisions(FlappyBird flappyBird) {  
    boolean found = false;  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
            break;  
        }  
    }  
    // что-то делаем с found...  
}
```

# ИДЕЙКА ПРО ЭТО ЗНАЕТ

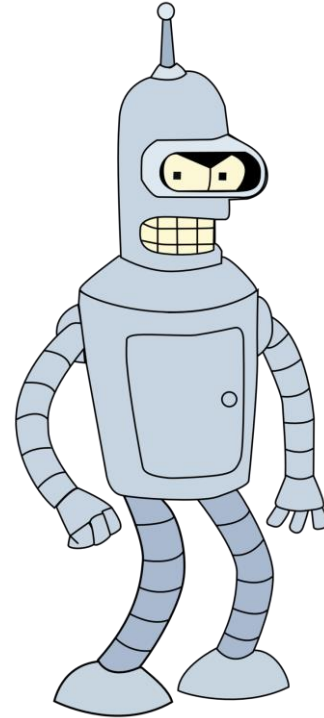
7

```
public void checkCollisions(FlappyBird flappyBird) {  
    boolean found = false;  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
    // что-то делаем с found...  
}
```

***Статический анализ*** – это анализ компьютерных программ, который производится без их запуска.



# Я, СТАТИЧЕСКИЙ АНАЛИЗАТОР



**Статический анализ** – это анализ компьютерных программ, который производится без их запуска. Обычно этот анализ производится **автоматически**.

# A KAK?

11

```
public void checkCollisions(FlappyBird flappyBird) {  
    boolean found = false;  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
    // что-то делаем с found...  
}
```

# ДАВАЙТЕ АВТОМАТИЗИРУЕМ

12

- » Ищем `for` в коде
- » Ищем внутри `if` (проверяем, что у него нет `else`). Больше ничего в цикле быть не должно
- » Смотрим, что внутри `if` есть только запись значения в переменную

# НО ЕСТЬ НЮАНС(Ы)

13

## Запись здорового человека

```
for (Pipe pipe : pipes) {  
    if (pipe.collission(flappyBird)) {  
        found = true;  
    }  
}
```

## Запись курильщика


```
for (Pipe pipe : pipes)  
    if (pipe.collission(flappyBird))  
    {  
        found = true;  
    }
```

# ДА ЛАДНО, ЧТО ТАМ ПИСАТЬ-ТО!

14

```
public int findForWithIfInside(String code, int start) {
    int forIdx = code.indexOf("for");
    if (forIdx == -1) {
        return -1;
    }
    // ищем конец (...)
    int clBraceIdx = code.indexOf(str: ")", forIdx);
    if (clBraceIdx == -1) {
        // weird, but ok...
        return -1;
    }
    // ищем ;
    int semicolonIdx = code.indexOf(str: ";", clBraceIdx);
    if (semicolonIdx == -1) {
        // weird, but ok...
        return -1;
    }
}
```

```
int ifIdx = code.indexOf(str: "if", clBraceIdx);
if (ifIdx == -1 || ifIdx > semicolonIdx) {
    // нету if'a внутри :(
    return -1;
}
// ищем конец условия if...
int ifClBraceIdx = code.indexOf(str: ")", ifIdx);
if (ifClBraceIdx == -1 || ifClBraceIdx > semicolonIdx) {
    // weird, but ok...
    return -1;
}
int eqIdx = code.indexOf(str: "=", ifClBraceIdx);
if (eqIdx == -1 || eqIdx > semicolonIdx) {
    // нету присваивания внутри :(
    return -1;
}
// тут я устал.....
```



**лол, я это лапками  
быстрее напишу**

**используй  
паттерны**

**ты забыл скобку  
на 34 строке  
обработать**

**регексы  
используй**

# РЕГЕКСЫ ДОМА...

16

```
for\s*\(.*\)\s*if\s*\(.*\)\s*\{\s*+.*=.*\s.*\}
```

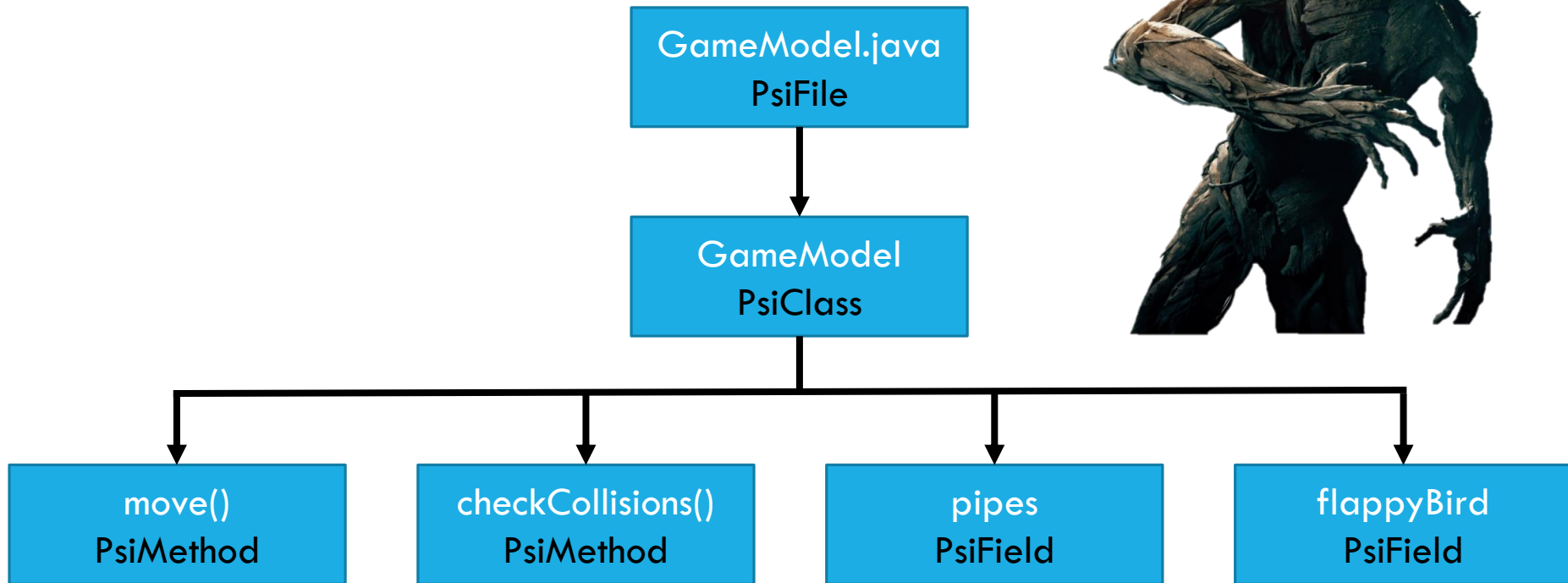


# КЛАСС – ЭТО ДЕРЕВО

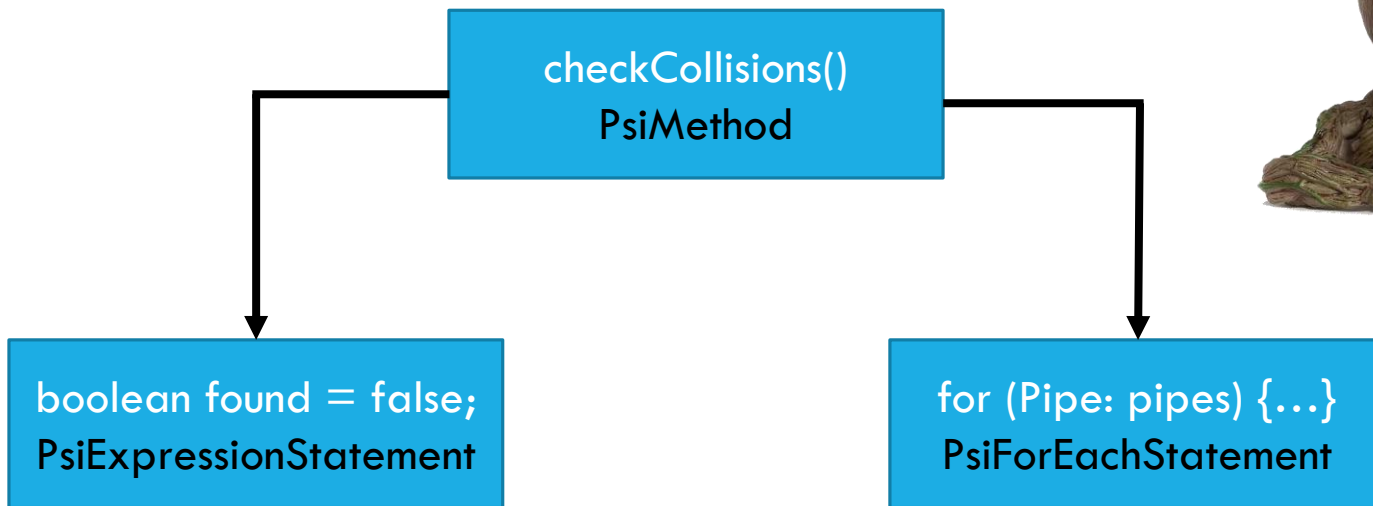
17

```
public class GameModel {  
  
    1 usage  
    Pipe[] pipes;  
    FlappyBird flappyBird;  
  
    public void move() {}  
  
    private void checkCollisions(FlappyBird flappyBird) {...}  
}
```

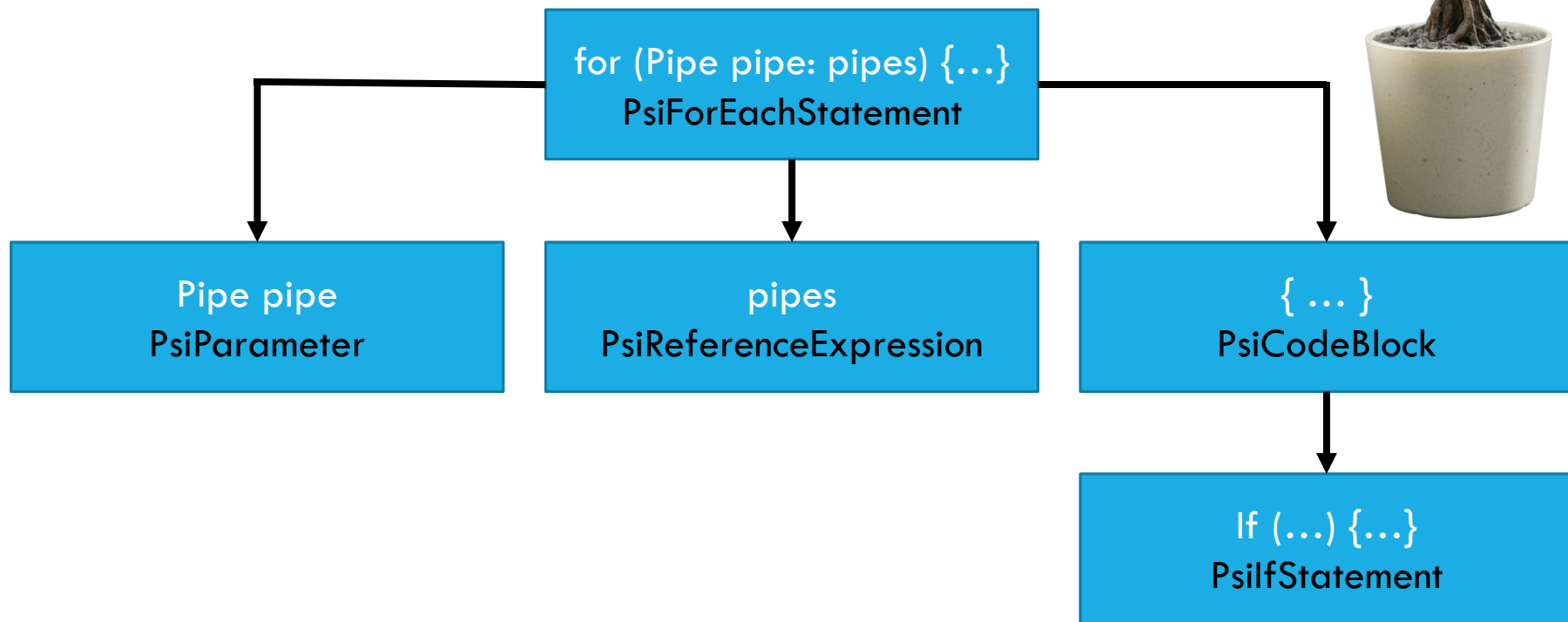
# ФАЙЛ – ЭТО ДЕРЕВО



# МЕТОД – ЭТО ДЕРЕВО



# FOR – ЭТО ДЕРЕВО



# ИНСПЕКЦИИ

21

- » Обходят дерево ([PsiFile](#)) вглубину, используют паттерн визитор
- » Помечают проблемный элемент и предлагают автоматические исправления (квик-фиксы)
- » На основе пометок подсвечивается код в редакторе



# ПИШЕМ ИНСПЕКЦИЮ



FALSE POSITIVE



# FALSE-POSITIVE

24

```
for (Pipe pipe : pipes) {  
    if (pipe.collission(flappyBird)) {  
        found = true;  
    }  
}
```



# FALSE-POSITIVE

25

```
for (Pipe pipe : pipes) {  
    if (pipe.collission(flappyBird)) {  
        found = true;  
    }  
}
```



***Сайд-эффект***— что-то, что производит наблюдаемое  
изменение состояния программы

# СТАНДАРТНЫЕ САЙД ЭФФЕКТЫ

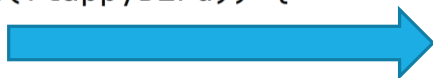
27

- » Выброс исключения
- » Логирование / `System.out.println`
- » Запись в поле

# САЙД-ЭФФЕКТЫ

28

```
for (Pipe pipe : pipes) {  
    if (pipe.collission(flappyBird)) {  
        found = true;  
    }  
}
```



```
class Pipe {  
  
    1 usage  
    int nHits;  
  
    1 usage  
    boolean collission(FlappyBird flappyBird) {  
        System.out.println("Collision checked!");  
        nHits++;  
        MagicUtils.someMagicMethod();  
        return true;  
    }  
}
```

# САЙД-ЭФФЕКТЫ (IDEA)

29

- » Кидание исключения
- » Логирование / `System.out.println`
- » Запись в поле
- » Чтение из `volatile` поля
- » Больше одного вызова

FALSE NEGATIVE



# FALSE NEGATIVE

31

- » Поддержать разные виды циклов
- » Скобочки и присваивание к нескольким переменным
- » Поддержать неявные `if`'ы
- » Удалять `if`, если переменная не используется

# ДА... (ВИДЫ ЦИКЛОВ)

32

```
private void checkCollisions(FlappyBird flappyBird) {  
    boolean found = false;  
    for (int i = 0; i < pipes.length; i++) {  
        Pipe pipe = pipes[i];  
        if (pipe.collission(flappyBird)) {  
            found = true;  
        }  
    }  
    // делаем что-то с found...  
}
```

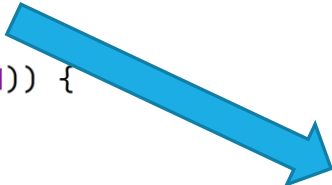




# НО СНОВА САЙД-ЭФФЕКТЫ...

33

```
private void checkCollisions() {  
    boolean found = false;  
    for (int i = 0; i < getLenSafe(i, pipes.length); i++) {  
        Pipe pipe = pipes[i];  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
}
```



```
private int getLenSafe(int i, int len) {  
    if (i > 10) {  
        throw new IllegalArgumentException("Too many pipes!");  
    }  
    return len;  
}
```

# ДА... (ЦЕПОЧКА ПРИСВАИВАНИЙ)

34

```
private boolean checkCollisions(FlappyBird flappyBird, boolean hasEvents) {  
    boolean found = false;  
    for (int i = 0; i < pipes.length; i++) {  
        Pipe pipe = pipes[i];  
        if (pipe.collision(flappyBird)) {  
            hasEvents = found = true;  
        }  
    }  
    // делаем что-то с found...  
    return hasEvents;  
}
```



# НО НУЖНО УЧЕСТЬ МАССИВЫ...

35

```
private static String[] bigOrSmall(int[] numbers) {  
    String[] strings = new String[numbers.length];  
    Arrays.fill(strings, val: "small");  
    for (int i : numbers) {  
        if (i > 10) {  
            strings[i] = "big";  
        }  
    }  
    return strings;  
}
```

# ДА... (НЕЯВНЫЙ IF)

36

```
boolean hasDefaultConstructor = false;  
for (MethodInfo constructor : state.getConstructors()) {  
    hasDefaultConstructor |=  
        (constructor.getParameters().isEmpty() && constructor.isPublic());  
}
```



# ДА...(НЕЯВНЫЙ IF)

37

```
boolean hasDefaultConstructor = false;
for (MethodInfo constructor : state.getConstructors()) {
    if (hasDefaultConstructor | (constructor.getParameters().isEmpty() && constructor.isPublic())) {
        hasDefaultConstructor = true;
    }
}
```



# НО ЕЩЕ НЕ СДЕЛАНО...

38

IDEA-254535 Created by Сергей Цыпанов 2 years ago Updated by Anna Kutarba 6 months ago

Visible to issue readers

Inspection 'Loop can be terminated when condition is met' fails to detect relevant snippet



Consider the method:

```
1 boolean hasDefaultConstructor = false;
2 for (MethodInfo constructor : state.getConstructors()) {
3     hasDefaultConstructor |= (constructor.getParameters().isEmpty() && constructor.isPublic());
4 }
```

Java

# BIGBRAIN ИНСПЕКЦИИ



# УДАЛЯТЬ IF ЦЕЛИКОМ

40

```
void checkCollisions() {  
    boolean found = false;  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
}
```



# ПЕРЕМЕННАЯ НЕ ИСПОЛЬЗУЕТСЯ

41

- » Есть объявление, без инициализатора
- » Есть только одно присваивание внутри `if` внутри цикла

```
boolean found;  
for (Pipe pipe : pipes) {  
    if (pipe.collission(flappyBird)) {  
        found = true;  
    }  
}
```



# ПЕРЕМЕННАЯ НЕ ИСПОЛЬЗУЕТСЯ

42

» Есть только записи, но нет чтения

```
void checkCollisions(boolean b) {  
    boolean found;  
    if (b) {  
        found = true;  
    }  
    for (Pipe pipe : pipes) {  
        if (pipe.collision(flappyBird)) {  
            found = true;  
        }  
    }  
}
```



# ПЕРЕМЕННАЯ НЕ ИСПОЛЬЗУЕТСЯ

43

- » Есть переприсваивания в другие переменные
- » Для всех этих переменных нет другого вида чтения

```
void checkCollisions(boolean b) {  
    boolean found = false;  
    if (b) {  
        found = true;  
    }  
    for (Pipe pipe : pipes) {  
        if (pipe.collison(flappyBird)) {  
            found = true;  
        }  
    }  
    boolean another = found;  
}
```



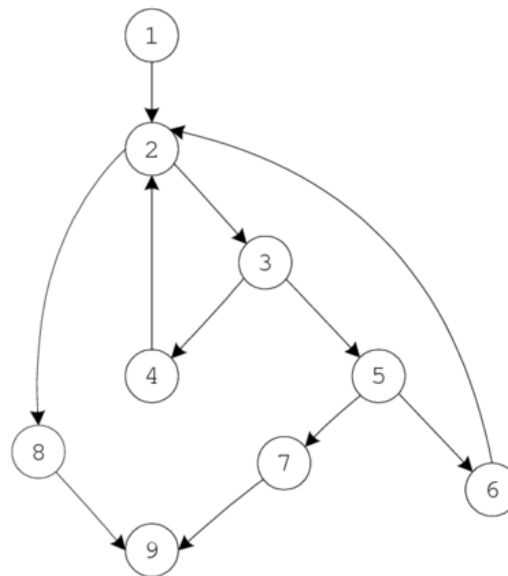
# CONTROL FLOW ANALYSIS

44

## Source Program:

```
int binsearch(int x, int v[], int n)
{
    int low, high, mid;
    1 | low = 0;
      | high = n - 1;
      | while (low <= high) | 2
      | {
      |     3 | mid = (low + high)/2;
      |     | if (x < v[mid])
      |     |     high = mid - 1; | 4
      |     5 | else if (x > v[mid])
      |     |     low = mid + 1; | 6
      |     7 | else return mid;
      | }
      | return -1; | 8
    } | 9
```

## CFG:



# ГДЕ ЕЩЕ СТАТИЧЕСКИЙ АНАЛИЗ

45

- » Редактор
- » Дебаггер
- » Коммиты
- » Поиск
- » To be continued...

# ИТОГ

46

- » Статический анализ – это автоматический анализ кода без его запуска
- » Очень часто в основе анализа лежит AST – представление кода в виде дерева
- » Написать базовую проверку вашего кода просто
- » Написать проверку для всех случаев бывает сложно, но весело

# Q&A



arty\_s



artfly

