

The JetBrains logo is located in the top right corner. It consists of a stylized, colorful shape resembling a diamond or a square with rounded corners, composed of overlapping bands in shades of pink, orange, and yellow. Inside this shape is a black square containing the text "JET BRAINS" in white, uppercase letters, with a horizontal white line below it.

JET
BRAINS

Kotlin Compiler

Саввинов Дмитрий

1. Non-local effects

Преамбула

Работает:

```
fun foo(x: Any) {  
    if (x is String) {  
        x.length  
    }  
}
```

Не работает :(

```
fun isString(x: Any): Boolean {  
    return x is String  
}  
  
fun foo(x: Any) {  
    if (isString(x)) {  
        x.length  
    }  
}
```

Преамбула

Ура, работает!

```
fun isString(x: Any): Boolean {  
    contract { returns(true) implies (x is String) }  
    return x is String  
}
```

```
fun foo(x: Any) {  
    if (isString(x)) {  
        x.length  
    }  
}
```



Проблема

- Следующий statement не знает про эффекты предыдущего

```
fun openFile(fileName: String) {  
    contract { throwsChecked(FileNotFoundException) }  
}
```

```
fun foo(x: Any) {
```

```
    ...
```

```
    openFile("foobar.txt")
```

```
    ...
```

```
}
```

IDE сообщает о том, что может полететь exception

- Другие use-case: context-restricted warnings (“не вызывайте writeAction вне EDT”, “не тупите внутри критической секции”), purity, const-ness, etc.
- Механизм распространения эффектов это почти что **вывод эффектов**

Задачи

- Разобраться с имеющейся кодовой базой
- Разобраться со стандартными техниками анализа кода (data/control flow analysis)
- Придумать, как реализовать нелокальные эффекты:
 - Основной челлендж: сделать не слишком мозговыносно, но при этом достаточно гибко для будущих расширений
- (*) Подумать, какие еще прикольные use case существуют
- (*) Попробовать допилить до полноценного вывода эффектов

Profit?

- Опыт работы с крупным и сложным проектом
- Опыт работы с компиляторами
 - В частности, с анализом кода
- Летняя стажировка **по этой теме**: скорее нет, чем да
- Дипломная работа **по этой теме**: может быть

2. Smart code statistics

Проблема

- Нам, как разработчикам языка, очень часто хочется знать, **какой** код пишут наши пользователи
 - Синтаксические конструкции искать легко:
 - Количество классов, функций, etc. - ОК
 - Есть даже автоматический crawler, который ходит по гитхабу и анализирует его
 - Семантические конструкции искать невозможно без полноценного запуска компилятора
 - “Вызовы, которые используют nested-класс из некоего предка, такого, что...”
- Текущее решение: собрать себе кастомный компилятор и опробовать его на десятке больших проектов
 - Долго, муторно, плохая выборка
- Хочется, чтобы было быстро, легко, точно.

Задачи

- Изучить кучу весьма разнородных “хотелок”
- Понять, как покрыть максимальное их количество и не сойти с ума
- Разработать некоторую систему (плагин к компилятору, подсистему компилятора, допилить crawler, etc.), которая бы смогла эти запросы удовлетворять
- Интеграция решения в production -- out of scope

Profit?

- Опыт решения задач, когда заказчик сам не знает, чего хочет :)
- Опыт работы с компиляторами
 - В частности, с устройством языка -- resolve, inference, IRs
- Летняя стажировка **по этой теме**: очень может быть!
 - Нам правда хочется иметь такую рабочую систему
- Дипломная работа **по этой теме**: почти наверняка нет
 - Недостаточно академично*

3. Kotlin Decompiler

Преамбула

- Компилятор -- программа, которая переводит язык высокого уровня в язык более низкого уровня
 - *javac* -- переводит код на языке Java в байткоды JVM
 - *kotlinc* -- переводит код на языке Kotlin в JVM (еще в JS и LLVM-IR)
- Декомпилятор -- программа, выполняющая обратное преобразование
- Зачем нужны декомпиляторы?
 - Просмотр кода библиотек без приложенных исходников
 - For science!

Проблема

—

- Его нет.



Задачи

- Научиться декомпилировать какое-нибудь подмножество языка
- Понять, насколько нереально декомпилировать более-менее весь язык

Profit?

- Опыт решения нерешаемых задач
- Очень много знаний о JVM-платформе, формате классфайлов, байткодах, etc.
- Много знаний о Котлине
- Летняя стажировка **по этой теме**: может быть
 - Единственная проблема, что работа может упереться в неразрешимые проблемы
- Дипломная работа **по этой теме**: может быть

**Thank you
for your attention**
