Летняя практика JetBrains-НГУ 2017

Экспериментальная лаборатория
ФИТ НГУ

Иртегов Д.В.

Темы проектов

1) Kotlin (консультант: Андрей Бреслав)

- Сравнение средств статического анализа, встроенных в различные популярные языки, включая различные специальные аннотации
- ANTLR-грамматика для языка Kotlin

2) IntelliJ IDEA (консультант: Рустам Вишняков)

- Импорт цветовых настроек для консоли из одного из популярных форматов в текущую цветовую схему (color scheme) Intellij IDEA

3) Marketing Research (консультант: Аркадий Калакуцкий)

- SurveyGizmo: утилита для выгрузки результатов опросов, проверки ответов на соответствие логике и сохранения результатов в определенном формате

Сравнение средств статического анализа

Стати́ческий ана́лиз ко́да — анализ программного обеспечения, производимый (в отличие от динамического анализа) без реального выполнения исследуемых программ.

Выполняется компиляторами или отдельными инструментами (напр. Lint для языка C)

Типы статического анализа

- Lint-like
  - Поиск распространенных ошибок
    - Неинициализированные переменные
    - Некорректные преобразования типов
    - Константные условия и недостижимые ветви кода
    - Вычисления, результат которых не используется
    - «Плохие» значения: деление на 0, нулевой указатель, выход индекса за границу массива
    - Непарные операции: open/close, lock/unlock, malloc/free

Типы статического анализа

- Формальная верификация
  - Анализ алгоритма на соответствие решаемой задаче
    - Требуют описания решаемой задачи
    - Описание задается внешним файлом или аннотациями в коде
    - Надо писать одну программу два раза: на целевом языке и на языке описания задачи
    - Профит: если вероятность допустить ошибку =P, то вероятность допустить одну ошибку при написании разными людьми на разных языках =P^2

Цель проекта

- Написать обзор (реферат), с конечной целью найти решения и приемы анализа, которые НЕ делаются компилятором Kotlin, но которые могли бы быть полезны
- Писать рефераты (литобзоры) вам в жизни пригодится, т.к. даже в бакалаврском дипломе иметь литобзор полезно

ANTLR-грамматика языка Kotlin

- ANTLR (от англ. ANother Tool for Language Recognition — «ещё одно средство распознавания языков») — генератор нисходящих анализаторов для формальных языков.
- Использует описание языка в формате РБНФ (расширенный формализм Бэкуса-Науэра)
- Генерирует лексический и синтаксический анализатор на различных языках (Java, C#, Python, JS)

Где применяется

- Разработка компиляторов
- В средах разработки
  - реализовать правильное синтаксическое подчеркивание в редакторе при помощи одних регулярных выражений бывает невозможно, надо все-таки разобрать синтаксис.
- Везде, где вам надо разобрать сложный язык
  - Интерпретаторы
  - Статические анализаторы
  - Отладчики

Пример программы на ANTLR

```
// Definition of an expression
statement: INTEGER (PLUS^ INTEGER)*;
// Here is the Lexer
PLUS: '+';
DIGIT: ('0'..'9');
INTEGER: (DIGIT)+;
```

Что из нее получается

```
{
            root_0 = (Object)adaptor.nil();

            INTEGER1=(Token)match(input,INTEGER,FOLLOW_INTEGER_in_statement12);
            INTEGER1_tree = (Object)adaptor.create(INTEGER1);
            adaptor.addChild(root_0, INTEGER1_tree);

            // Add.g:4:21: ( PLUS INTEGER )*
            loop1:
            do {
                int alt1=2;
                int LA1_0 = input.LA(1);

                if ( (LA1_0==PLUS) ) {
                    alt1=1;
                }


                switch (alt1) {
                case 1 :
                   // Add.g:4:22: PLUS INTEGER
                   {
                   PLUS2=(Token)match(input,PLUS,FOLLOW_PLUS_in_statement15);
                   PLUS2_tree = (Object)adaptor.create(PLUS2);
                   root_0 = (Object)adaptor.becomeRoot(PLUS2_tree, root_0);

                   INTEGER3=(Token)match(input,INTEGER,FOLLOW_INTEGER_in_statement18);
                   INTEGER3_tree = (Object)adaptor.create(INTEGER3);
                   adaptor.addChild(root_0, INTEGER3_tree);


                   }
                   break;

               default :
                   break loop1;
                }
            } while (true);

 }
```

Что такое Kotlin

Kotlin (Ко́тлин) — статически типизированный язык программирования, работающий поверх JVM и разрабатываемый компанией JetBrains.

```
fun sayHello(maybe : String?, neverNull : Int) {
    // use of elvis operator
    val name : String = maybe ?: "stranger"
    println("Hello $name")
}
```

Зачем ANTLR грамматика для Kotlin?

- Разработчики языка надеются, что свободно распространяемая ANTLR грамматика облегчит реализацию поддержки Kotlin в других проектах
  - Среды разработки
  - Отладчики
  - Анализаторы/верификаторы
  - …

SurveyGizmo

Software As Service платформа для проведения онлайн-опросов

Позволяет создавать веб-анкеты и получать собранные данные.

Позволяет задавать логику опроса (например, если указан род занятий «Студент», то не надо спрашивать о сфере деятельности компании, в которой человек работает)

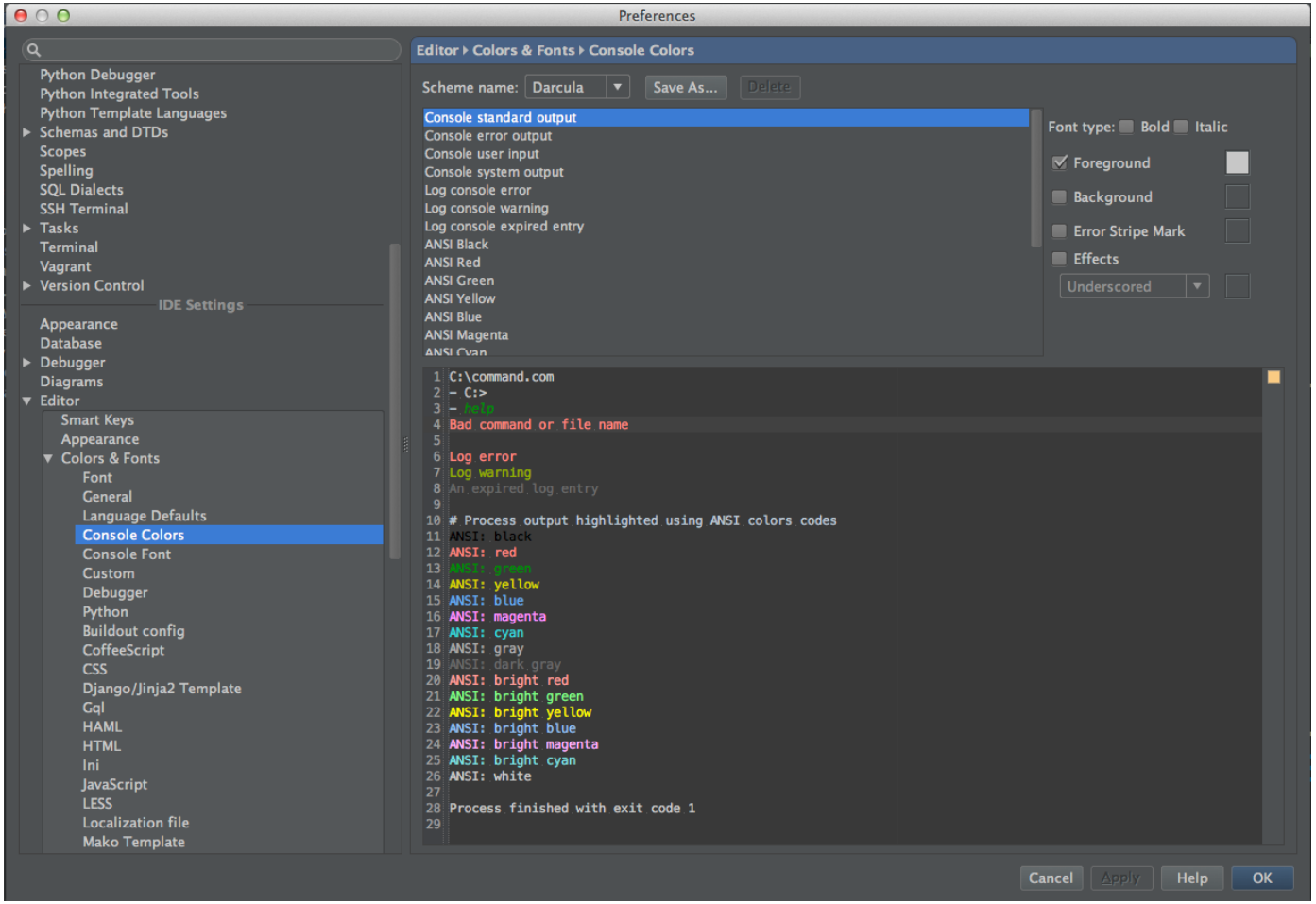
Проверка ответов на соответствие логике

- Иногда SurveyGizmo почему-то отдает ответы, не соответствующие логике (студент сообщает, что работает в компании с 500 сотрудниками)
- Нужно научиться их отфильтровывать
- Задача сложнее, чем кажется
  - У сервиса есть API для получения данных, но нет API для получения логики
  - Логика задается только через Web GUI сервиса
  - Надо разрабатывать собственный формат (язык?) для описания этой логики

Импорт цветовых настроек для консоли Intellij Idea

- Feature request от пользователей Idea
- Если вы это сделаете, возможно, ваш код будет реально использоваться в составе Idea

Что такое цветовые настройки



Откуда их надо импортировать

- Пользователи просят:
  - .colorscheme (Konsole)
  - .itermcolors (iTerm)
  - .reg (PuTTY)
  - .terminal (Terminal)
  - .config (Terminator)
- Есть целые коллекции готовых цветовых схем, напр.
  https://github.com/mbadolato/iTerm2-Color-Schemes

Пример: Putty