

Тестовое задание для Senior Developer в Compiler Team

Смотреть будем на:

- корректность и полноту реализации,
- качество кода (представим себе, что нам этот проект придется развивать и поддерживать еще долго-долго),
- тесты,
- удобство сборки/запуска/использования.

Процесс:

Если у Вас есть вопросы по заданию, не стесняйтесь их задавать.

Перед тем как писать код, нужно прислать нам краткое описание общего подхода к реализации, в частности если в задании что-то не указано, Вы принимаете решение как доопределить поведение, и сообщаете нам об этом (целый ряд вещей намеренно не определен, чтобы дать Вам простор для творчества). Если по ходу реализации всплывут еще какие-то подобные моменты, тоже сообщите нам как можно скорее.

Описание

Требуется реализовать поиск избыточных проверок на null в байткоде JVM. “Проверка на null” - это сравнение некоего выражения с null. Такая проверка считается избыточной, если это сравнение всегда возвращает true или всегда возвращает false. Случаи, когда это так, приведены в примерах ниже. Можно полагаться на сохранение семантики final полей. Если у вас останется время, можете подумать над собственными примерами.

Входом является class-файл. Можно считать, что этот class-файл получен в результате компиляции исходников на Java с использованием javac (для простоты исходный код в примерах дан на языке Java).

Выходом является перечень обнаруженных избыточных проверок на null. Формат диагностического сообщения - на ваше усмотрение. По крайней мере, следует указать имя файла исходного кода и номер строки, в которой происходит избыточная проверка на null.

Пример №1

```
package test;

public class Example1 {
    public static void test1() {
        Object x = null;
        if (x == null) { // *
            System.out.println("x is null");
        }
    }
}
```

```

    public static void test2() {
        Object x = new Object();
        if (x == null) { // *
            System.out.println("x is null");
        }
    }

    public static void test3() {
        Object x = "Hello, world!";
        if (x == null) { // *
            System.out.println("x is null");
        }
    }
}

```

Пример №2

```

package test;

public class Example2 {
    public static void test1(Object x) {
        if (x == null) return;
        if (x != null) { // *
            System.out.println("x != null");
        }
    }

    public static void test2(Object x) {
        System.out.println(x.hashCode());
        if (x == null) { // *
            System.out.println("x == null");
        }
    }
}

```

Пример №3

```

package test;

public class Example3 {
    private final Object x;

    public Example3() {
        x = new Object();
    }

    public void test() {
        if (x == null) { // *
            System.out.println("x == null");
        }
    }
}

```

Пример №4

```
package test;

public class Example4 {
    public final Object foo() {
        return new Object();
    }

    public void test() {
        if (foo() != null) { // *
            System.out.println("foo() != null");
        }
    }
}
```

Пример №5

```
package test;

public class Example5 {
    private static final String MESSAGE = "Hello, world!";

    public void test() {
        if (MESSAGE != null) { // *
            System.out.println(MESSAGE);
        }
    }
}
```