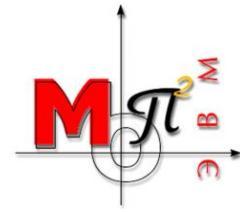




МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

} |
ИКТИБ



Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

ВПК-8. Проект «Прикладное решение на языке Python»

по направлению 09.04.04 – Программная инженерия (Мобильная и веб-разработка)
КТмо2-16

ОПТИМИЗАЦИЯ И РЕФАКТОРИНГ JAVA ПРИЛОЖЕНИЙ В ТГ БОТЕ НА PYTHON

Руденко Константин Дмитриевич

Преподаватель - доц., к. т. н. Лапшин Вячеслав Сергеевич

2025 г.

ПОСТАНОВКА ЗАДАЧИ

Цель:

В рамках темы

Оптимизация и рефакторинг Java-приложений

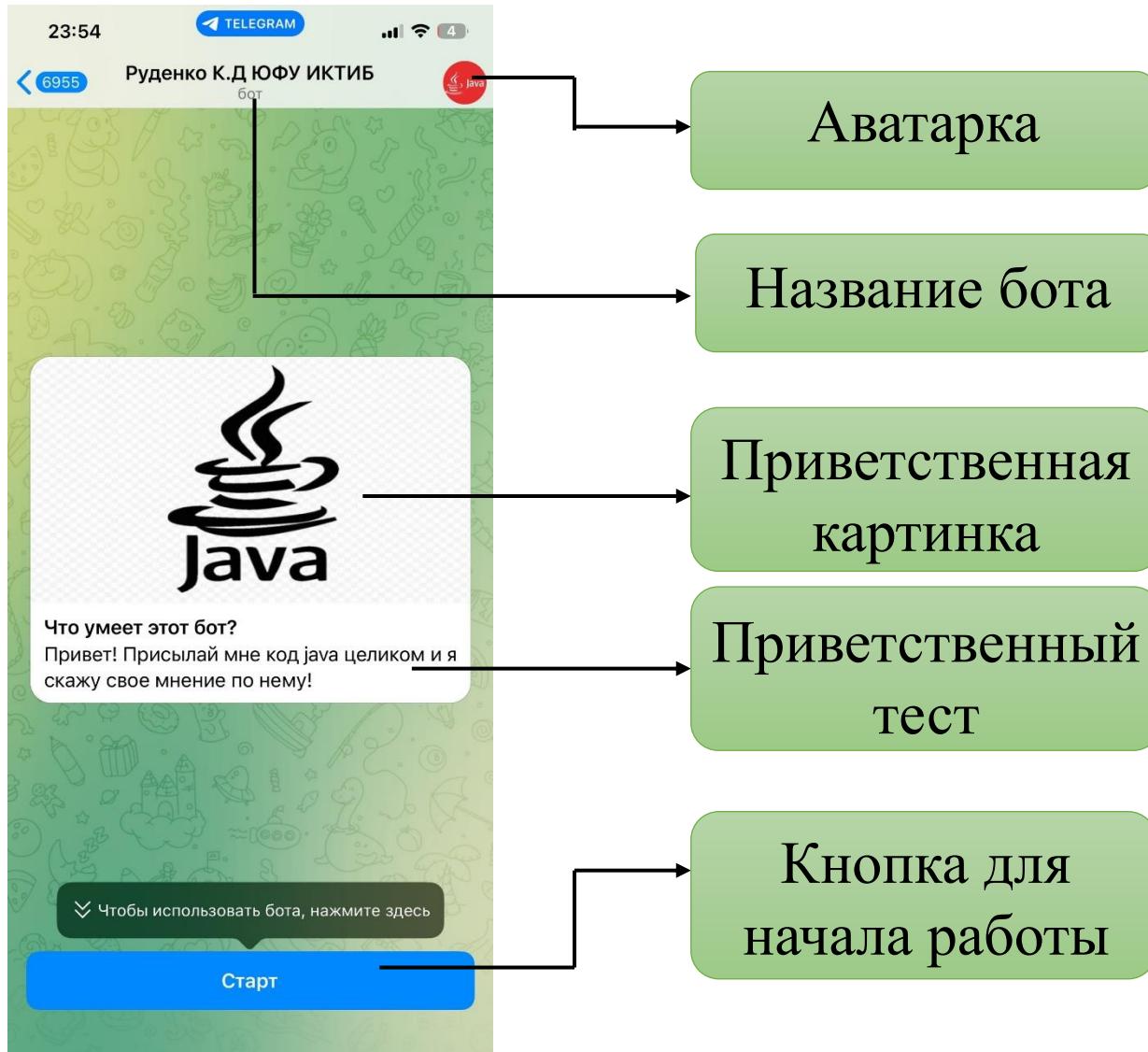
разработать чат-бот со следующим функционалом:

- погружение в методы рефакторинга
- анализатор кода
- обратная связь с разработчиком бота

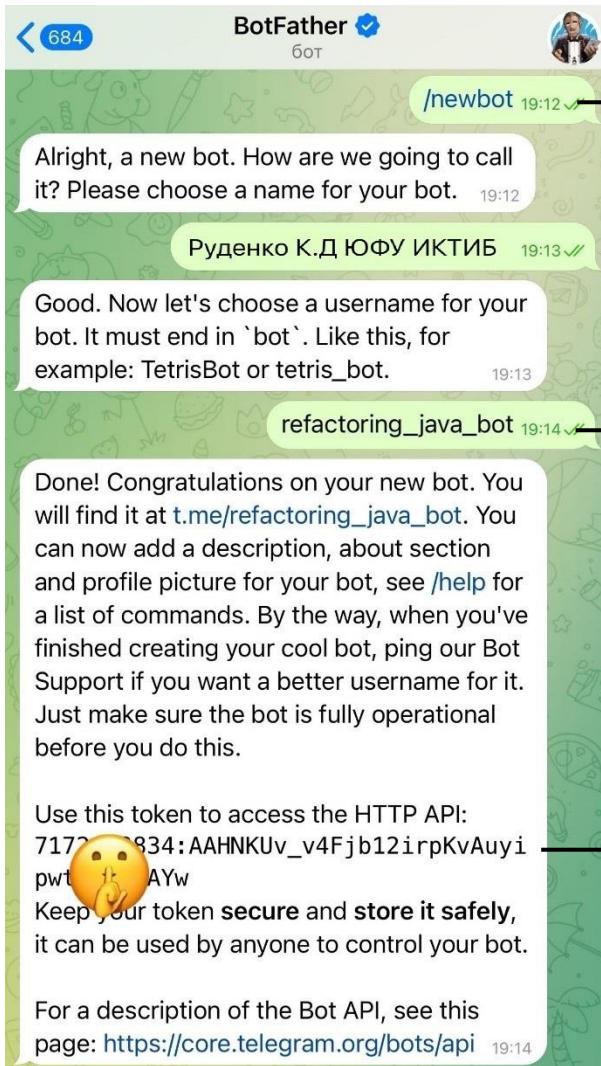
Провести тестирование чат-бота.

TELEGRAM БОТ

РЕФАКТОРИНГ НА JAVA



ПОЛУЧЕНИЕ ТОКЕНА И ЭКРАН ПРИВЕТСТВИЯ



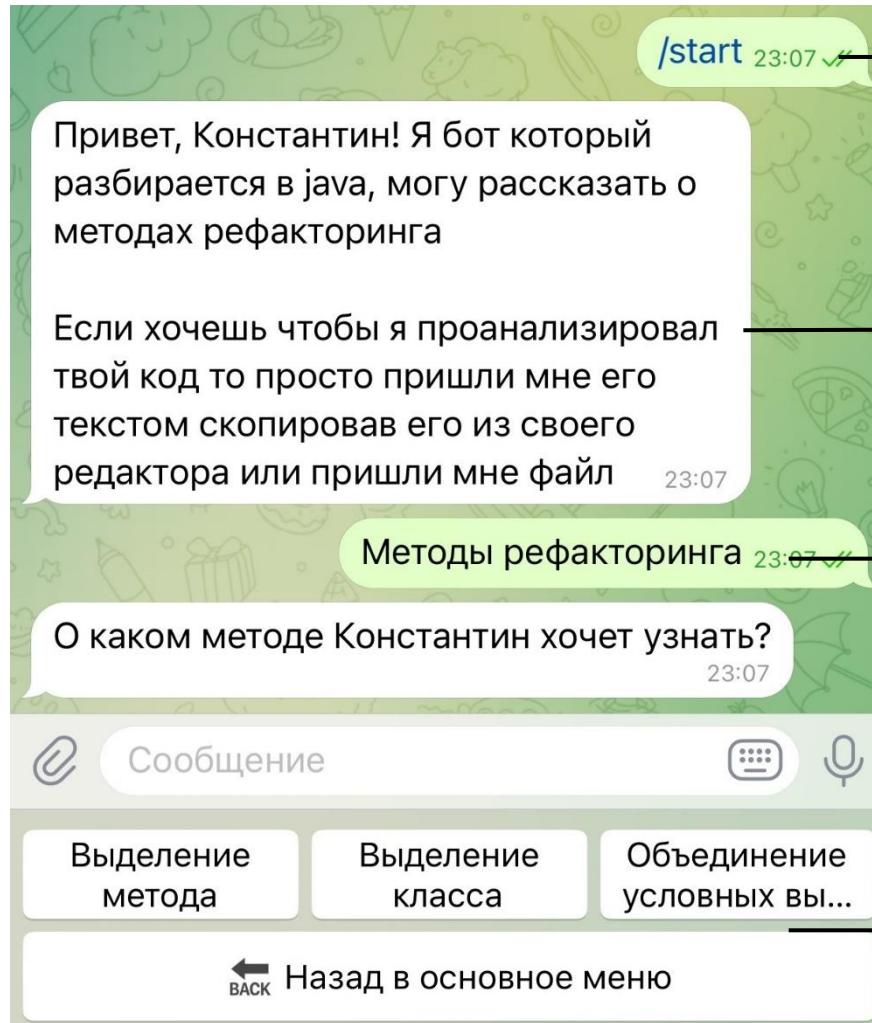
Бот администратор

Команда для создания нового бота

Короткая ссылка

Секретный токен

МЕТОДЫ РЕФАКТОРИНГА В TELEGRAM БОТЕ



Команда для начала пользования ботом

Что я умею!

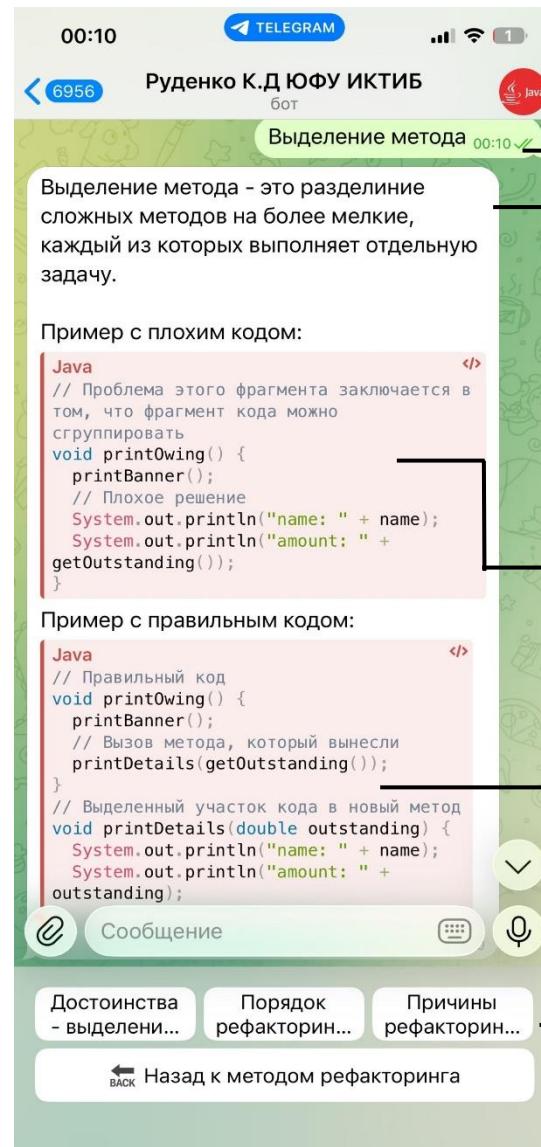
Нажатие на кнопку

Всплывающие
кнопки

ОСНОВНЫЕ МЕТОДЫ РЕФАКТОРИНГА

- Выделение класса
- Выделение метода
- Передача всего объекта
- Отказ от наследования
- Замена временной переменной запросом
- Объединение дублирующего кода
- Введение переменной
- Упрощение условных выражений
- Замена числовых значений константами
- Инкапсуляция полей
- Разделение больших классов

ЛЕГКОЕ ПОГРУЖЕНИЕ В ПРИНЦИПЫ. ВЫДЕЛЕНИЕ МЕТОДА



Запрос

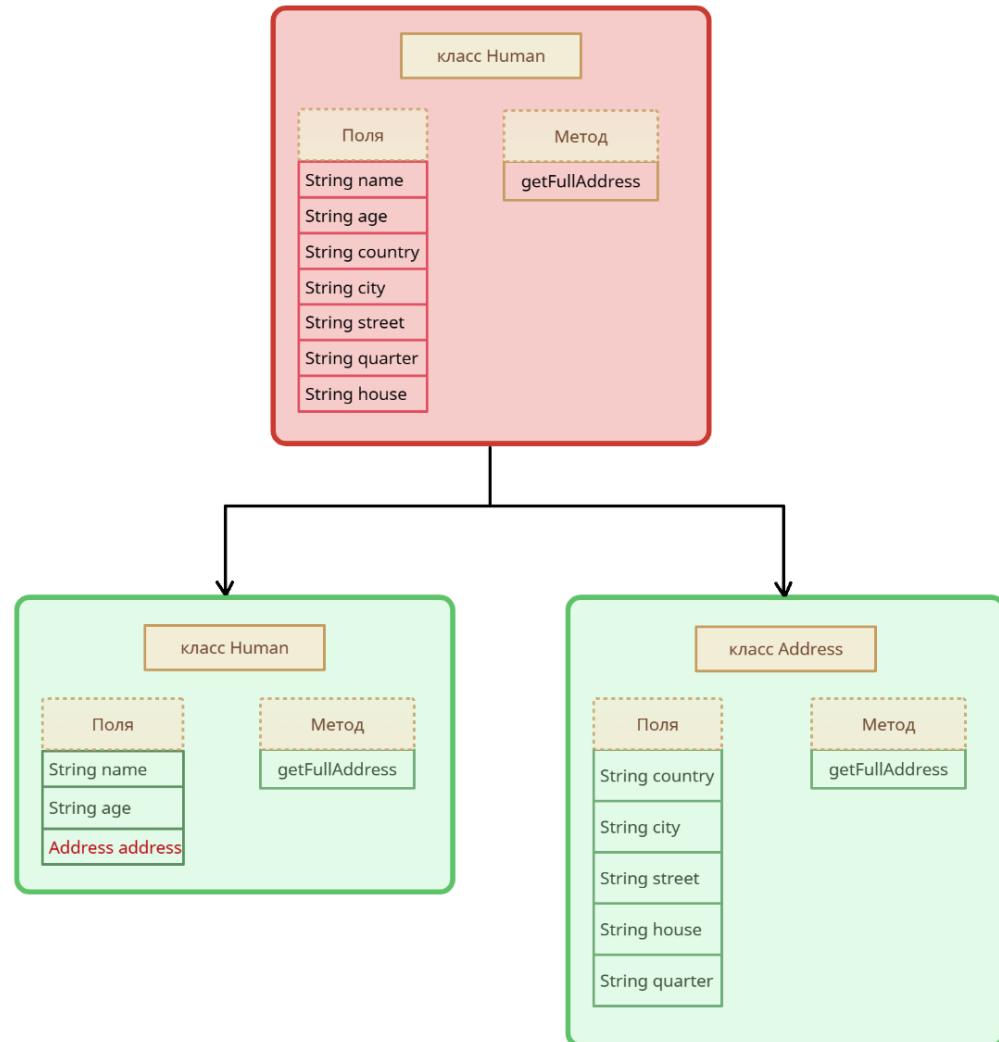
Определение

До рефакторинга

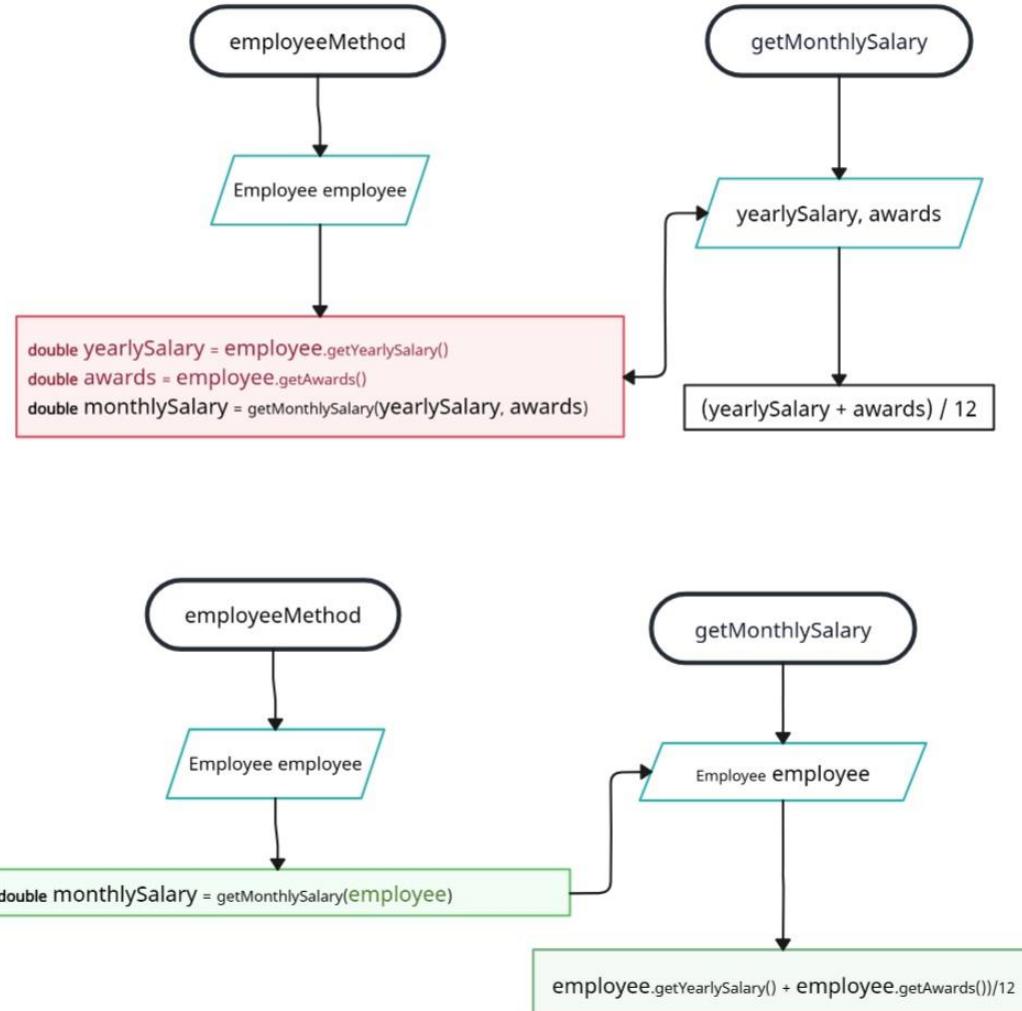
Оптимизированная
версия

Углубленное
изучение

ПОГРУЖЕНИЕ В МЕТОДЫ РЕФАКТОРИНГА. ВЫДЕЛЕНИЕ КЛАССА



ПОГРУЖЕНИЕ В МЕТОДЫ РЕФАКТОРИНГА. ПЕРЕДАЧА ВСЕГО ОБЪЕКТА.



ПОГРУЖЕНИЕ В МЕТОДЫ РЕФАКТОРИНГА. ОБЪЕДИНЕНИЕ УСЛОВНЫХ ВЫРАЖЕНИЙ.



Пример с плохим кодом:

```
Java
/*
Проблема
У вас есть несколько условных операторов,
ведущих к одинаковому результату или
действию.
*/
double disabilityAmount() {
    if (seniority < 2) {
        return 0;
    }
    if (monthsDisabled > 12) {
        return 0;
    }
    if (isPartTime) {
        return 0;
    }
    // И так далее
    // ...
}
```

До
рефакторинга

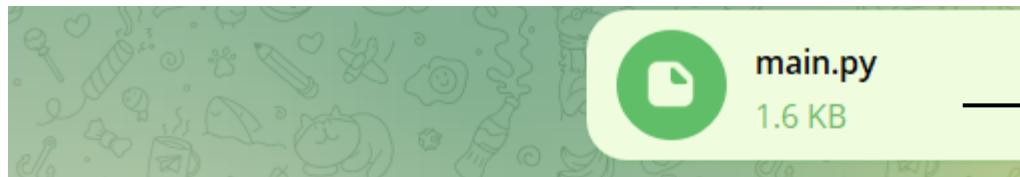
Пример с правильным кодом:

```
Java
/*
Решение
Объедините все условия в одном условном
операторе.
*/
double disabilityAmount() {
    if (seniority < 2) || (monthsDisabled >
12) || (isPartTime){
        return 0;
    }

    // Продолжение кода
    // ...
}
// Или вынести в отдельный метод
double disabilityAmount() {
    if (isNotEligibleForDisability()) {
        return 0;
    }
    // Продолжение кода
    // ...
}
```

После
ОПТИМИЗАЦИИ

ОТПРАВКА ФАЙЛОВ



ЯП другой

Константин Руденко
main.py

Файл не проанализирован, я понимаю только файлы на ЯП java
23:51



Правильный ЯП

Константин Руденко
Main.java

Имя проанализированного файла: [Main.java](#)

[Main.java:1: NoPackage](#): All classes, interfaces, enums and annotations must belong to a named package

[Main.java:1: UseUtilityClass](#): This utility class has a non-private constructor

[Main.java:5: SimplifyBooleanReturns](#): This if statement can be replaced by `return false;`

АНАЛИЗ КОДА

```
public class Main {  
    public static boolean maxi( int x , int y ){  
        if( x > y ){  
            return true ;  
        } else if ( x == y ) {  
            return false ;  
        } else {  
            return false ;  
        }  
    }  
    public static void main(String[] args) {  
        System.out.println(maxi(2, 3));  
    }  
}
```

23:51 ✓



Текст на java

Константин Руденко

```
public class Main { public static boolean maxi( int x , int y ) { . . .  
file.java:1: NoPackage: All classes, interfaces, enums and  
annotations must belong to a named package  
file.java:1: UseUtilityClass: This utility class has a non-private  
constructor  
file.java:5: SimplifyBooleanReturns: This if statement can be  
replaced by `return false;`
```

Ответ на английском
языке

Перевод на русский язык:

```
file.java:1: норепакедж: Все классы, интерфейсы, перечисления и  
аннотации должны принадлежать именовано  
file.java:1: useUtilityclass: этот утилита имеет непредоходную  
конструктор  
file.java:5: umidifybooleanreturns: это оператор может быть  
заменен на `вернуть false;` ``
```

23:

Ответ на русском
языке

АНАЛИЗАТОР КОДА

```
public class Example {  
    // Неиспользуемая переменная  
    private int unusedVariable;  
    public void performOperations() {  
  
        // Избыточное создание объектов  
        String redundantString = new String("Redundant");  
        // Неправильное использование строк и операций с ними  
        String inefficientString = "";  
        for (int i = 0; i < 1000; i++) {  
            inefficientString += "a";  
        }  
        // Использование небезопасных API  
        File file = new File("example.txt");  
        FileReader fr = null;  
        try {  
            fr = new FileReader(file);  
            BufferedReader br = new BufferedReader(fr);  
            String line;  
            while ((line = br.readLine()) != null) {  
                System.out.println(line);  
            }  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        } catch (IOException e) {  
            e.printStackTrace();  
        } finally {  
            // Утечка ресурсов: не закрыты FileReader и  
            // BufferedReader  
            // Неправильное управление памятью  
        }  
    }  
    // Несоответствие соглашениям о наименовании  
    public void BADNAMINGCONVENTION() {  
        System.out.println("Bad naming convention");  
    }  
}
```

0:05

Константин Руденко

```
public class Example { // Неиспользуемая переменная pri...  
file.java:1: NoPackage: All classes, interfaces, enums and  
annotations must belong to a named package  
file.java:3: UnusedPrivateField: Avoid unused private fields such as  
'unusedVariable'.  
file.java:7: UnusedLocalVariable: Avoid unused local variables such  
as 'redundantString'.  
file.java:9: UnusedLocalVariable: Avoid unused local variables such  
as 'inefficientString'.  
file.java:25: IdenticalCatchBranches: 'catch' branch identical to  
'FileNotFoundException' branch  
file.java:27: EmptyControlStatement: Empty finally clause  
file.java:33: MethodNamingConventions: The instance method  
name 'BADNAMINGCONVENTION' doesn't match '[a-z][a-zA-Z0-9]*'
```

Перевод на русский язык:

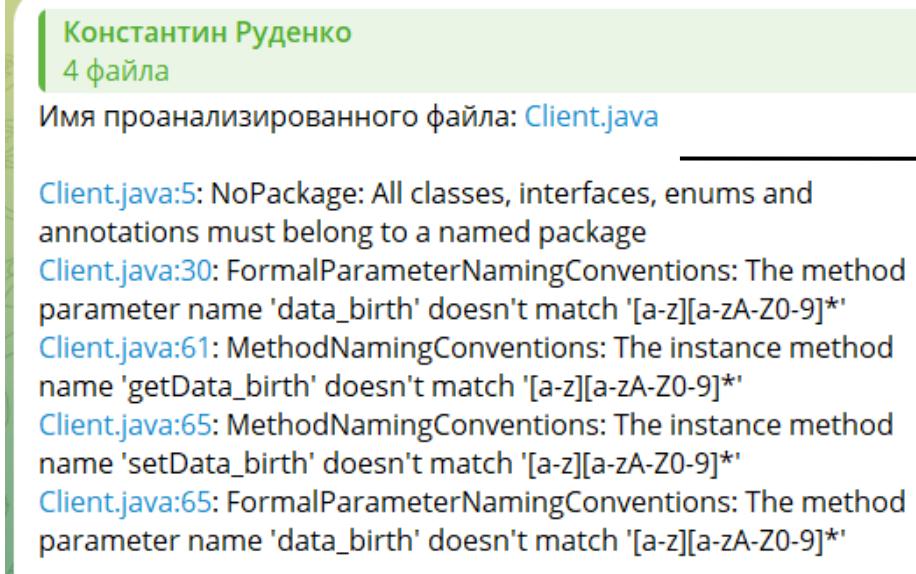
file.java:1: **напаккейс**: Все классы, интерфейсы, перечисления и
аннотации должны принадлежать именовано
file.java:3: **Unaseprivatefield**: Избегайте неиспользованных
частных полей, таких как «неиспользованный».
file.java:7: **неиспользованный классарвируемый**: избегайте
неиспользованных локальных переменных, таких как «избыточная
тренажера».
file.java:9: **неиспользованный классарвируемые**: избегайте
неиспользованных локальных переменных, таких как
«неэффективные».
file.java:25: **идентичные CATCHBRANCHES**: «Поймайте» ветвь,
идентичный «FileNotFoundException»
file.java:27: **EmptyControLlestatement**: Пустое, наконец, пункт
file.java:33: **MethodnamingConventions**: Имя метода экземпляра
«BadnamingConvention» не соответствует '[a-z] [a-zA-Z0-9]*' 0:05

13/17

ТЕСТИРОВАНИЕ АНАЛИЗАТОРА В TELEGRAM БОТЕ

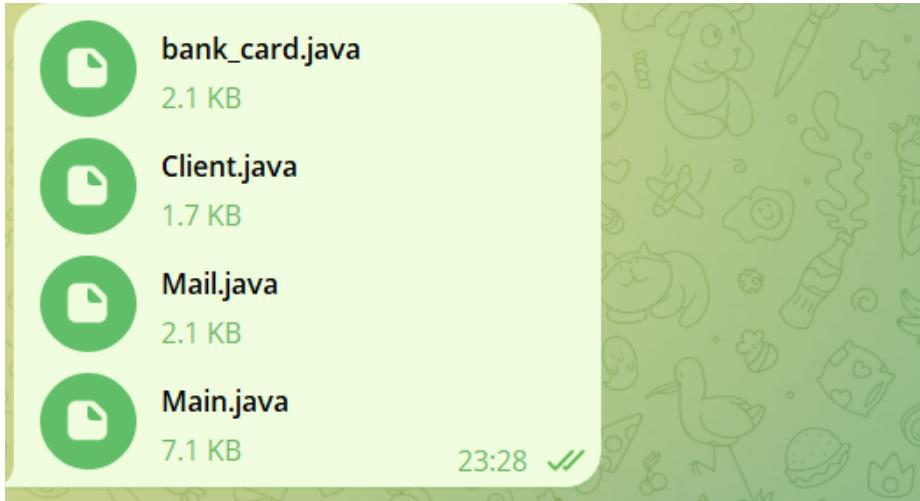


Отправка проекта



Ответ анализатора

ТЕСТИРОВАНИЕ АНАЛИЗАТОРА В TELEGRAM БОТЕ



Нарушение о
наименовании методов

Константин Руденко
4 файла

Имя проанализированного файла: [Client.java](#)

[Client.java:5](#): NoPackage: All classes, interfaces, enums and annotations must belong to a named package

[Client.java:30](#): FormalParameterNamingConventions: The method parameter name 'data_birth' doesn't match '[a-z][a-zA-Z0-9]*'

[Client.java:61](#): MethodNamingConventions: The instance method name 'getData_birth' doesn't match '[a-z][a-zA-Z0-9]*'

[Client.java:65](#): MethodNamingConventions: The instance method name 'setData_birth' doesn't match '[a-z][a-zA-Z0-9]*'

[Client.java:65](#): FormalParameterNamingConventions: The method parameter name 'data_birth' doesn't match '[a-z][a-zA-Z0-9]*'

ТЕСТИРОВАНИЕ АНАЛИЗАТОРА В TELEGRAM БОТЕ

Константин Руденко
4 файла

Имя проанализированного файла: [Main.java](#)

Main.java:7: NoPackage: All classes, interfaces, enums and annotations must belong to a named package
Main.java:7: UseUtilityClass: This utility class has a non-private constructor
Main.java:22: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:29: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:34: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:53: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:60: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:65: SimplifyBooleanReturns: This if statement can be replaced by `return {condition};`
Main.java:128: LocalVariableNamingConventions: The local variable name 'b_c' doesn't match '[a-z][a-zA-Z0-9]*'
Main.java:145: UselessParentheses: Useless parentheses.
Main.java:145: UselessParentheses: Useless parentheses.
Main.java:146: UselessParentheses: Useless parentheses.
Main.java:146: UselessParentheses: Useless parentheses.
Main.java:147: UselessParentheses: Useless parentheses.
Main.java:147: UselessParentheses: Useless parentheses.
Main.java:148: UselessParentheses: Useless parentheses.
Main.java:148: UselessParentheses: Useless parentheses.
Main.java:149: UselessParentheses: Useless parentheses.
Main.java:149: UselessParentheses: Useless parentheses.
Main.java:150: UselessParentheses: Useless parentheses.
Main.java:150: UselessParentheses: Useless parentheses.
Main.java:158: UselessParentheses: Useless parentheses.
Main.java:158: UselessParentheses: Useless parentheses.
Main.java:158: UselessParentheses: Useless parentheses.
Main.java:159: UselessParentheses: Useless parentheses.
Main.java:159: UselessParentheses: Useless parentheses.
Main.java:159: UselessParentheses: Useless parentheses.

Лишние скобки

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

В рамках данной работы

«Оптимизация и рефакторинг Java-приложений»

разработан чат-бот со следующим функционалом:

- погружение в методы рефакторинга
- анализатор кода
- обратная связь с разработчиком бота

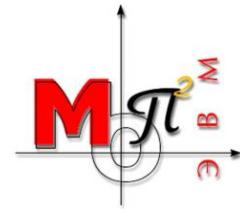
Чат-бот прошел тестирование на коде, написанном при прохождении производственной практики в СберТех.



МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

} |
ИКТИБ



Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

ВПК-8. Проект «Прикладное решение на языке Python»

по направлению 09.04.04 – Программная инженерия (Мобильная и веб-разработка)
КТмо2-16

ОПТИМИЗАЦИЯ И РЕФАКТОРИНГ JAVA ПРИЛОЖЕНИЙ В ТГ БОТЕ НА PYTHON

Руденко Константин Дмитриевич

Преподаватель - доц., к. т. н. Лапшин Вячеслав Сергеевич

2025 г.