## Учреждение образования

## Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Тема

Изучение соглашение по оформлению кода - Java Code Conventions

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Выполнил студент

ФИТ 4 курс 7 группа Бобрович Г.С.

Проверил

кандидат технических наук

Сулим П.Е.

Отчет по лабораторной работе

защищен с отметкой баллов

Минск 2024

***Цель* *работы*:** написать Java-code по правилам Java Code Conventions.

***Описание работы:***

Необходимо ознакомится с Java Code Conventions, привести примеры Java-кода с пояснениями по оформлению.

1. [Обеспечение доступа к экземпляру и переменным класса](https://www.magnumblog.space/java/translating-java-code-conventions)

Одним из популярных модификаторов в ООП является модификатор доступа. Он определяет уровень доступности членов класса для других классов и объектов. В языке Java, например, существуют следующие модификаторы доступа:

public: член класса доступен из любого места программы

protected: член класса доступен только внутри пакета и для наследников класса

private: член класса доступен только внутри самого класса

default: член класса доступен только внутри пакета

Другим популярным модификатором является модификатор static. Он позволяет определять члены класса, которые принадлежат самому классу, а не отдельным объектам этого класса. Такие члены доступны без создания экземпляра класса и могут быть общими для всех объектов данного класса.

1. [Обращение к переменным и методам класса](https://www.magnumblog.space/java/translating-java-code-conventions)

Для обращения к переменным и методам класса необходимо использовать имя класса

classMethod();

AClass.classMethod();

1. [Константы](https://www.magnumblog.space/java/translating-java-code-conventions)

Кроме переменных, в Java для хранения данных можно использовать константы. В отличие от переменных константам можно присвоить значение только один раз. Константа объявляется также, как и переменная, только вначале идет ключевое слово final

final int LIMIT = 5;

System.out.println(LIMIT); // 5

// LIMIT=57; // так мы уже не можем написать, так как LIMIT – константа

Как правило, константы имеют имена в верхнем регистре.

Константы позволяют задать такие переменные, которые не должны больше изменяться. Например, если у нас есть переменная для хранения числа pi, то мы можем объявить ее константой, так как ее значение постоянно.

1. [Примеры присваивания значение переменным](https://www.magnumblog.space/java/translating-java-code-conventions)

Инициализация — это присвоение начального значения переменной. То есть, если Вы только создали переменную и не задали ей еще никакого значения, то данная переменная не инициализирована. Так что, если услышите:

"Необходимо проинициализировать переменную" — это всего лишь имеется ввиду, что "необходимо присвоить начальное значение переменной"

"Мы проинициализировали переменную" — это значит "мы присвоили начальное значение переменной"

Пример 1:

if ((c++ = d++) != 0) {

...

}

Пример 2:

a = b + c;

d = a + r;

1. [Круглые скобки](https://www.magnumblog.space/java/translating-java-code-conventions)

Как правило, хорошей практикой является использование круглых скобок в выражениях, содержащих различные операторы, чтобы избегать проблем с приоритетом операторов.

if ((a == b) && (c == d))

1. [Примеры кода](https://www.magnumblog.space/java/translating-java-code-conventions)

public class Test {// объявляется класс Test

public static void main(String[] args) {// так включаются программы,

// просто запомнить

System.out.println("Hello world");

System.out.println("5 раз");

}

}

Вывод: было изучено соглашение по оформлению кода Java Code Conventions, рассмотрены отдельные случаи и примеры.