|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №1  по дисциплине «Языки программирования»  Вариант 5 | | |
|  | | Работу выполнил  студент группы ПМИ-1,2-2023 2 курса  Волегов И.С.  «25» Декабря 2025 г. |
| Работу проверил  Ракина В.Д.  «\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Задание 2 4](#_Toc186055342)

[Текст задания 4](#_Toc186055343)

[Алгоритм решения 4](#_Toc186055344)

[Тестирование 4](#_Toc186055345)

[Код программы 4](#_Toc186055346)

[Задание 3 5](#_Toc186055347)

[Текст задания 5](#_Toc186055348)

[Алгоритм решения 5](#_Toc186055349)

[Тестирование 5](#_Toc186055350)

[Код программы 5](#_Toc186055351)

[Задание 4 6](#_Toc186055352)

[Текст задания 6](#_Toc186055353)

[Алгоритм решения 6](#_Toc186055354)

[Тестирование 6](#_Toc186055355)

[Код программы 6](#_Toc186055356)

[Задание 5 7](#_Toc186055357)

[Текст задания 7](#_Toc186055358)

[Алгоритм решения 7](#_Toc186055359)

[Тестирование 7](#_Toc186055360)

[Код программы 7](#_Toc186055361)

[Задание 6 8](#_Toc186055362)

[Текст задания 8](#_Toc186055363)

[Алгоритм решения 8](#_Toc186055364)

[Тестирование 8](#_Toc186055365)

[Код программы 8](#_Toc186055366)

[Задание 7 9](#_Toc186055367)

[Текст задания 9](#_Toc186055368)

[Алгоритм решения 9](#_Toc186055369)

[Тестирование 9](#_Toc186055370)

[Код программы 9](#_Toc186055371)

# Задание 2

## Текст задания

@Invoke. Разработайте аннотацию @Invoke, со следующими характеристиками: • Целью может быть только МЕТОД

• Доступна во время исполнения программы

• Не имеет свойств

Проаннотируйте какой-либо метод данной аннотацией..

## Алгоритм решения

1. **Создаем аннотацию @Invoke.**

**- Она применяется к методам и доступна во время выполнения.**

**2. В классе TestInvoke создаем методы:**

**- Один метод будет аннотирован аннотацией @Invoke, второй — нет.**

**3. В методе main:**

**- Создаем объект TestInvoke.**

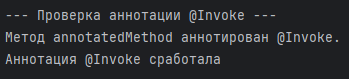
**- Получаем все методы класса с помощью рефлексии.**

**- Для каждого метода проверяем наличие аннотации @Invoke.**

**- Если аннотация найдена, вызываем метод.**

.

## Тестирование

.

## Код программы

Https://github.com/HintMSI/laba\_6

# Задание 3

## Текст задания

@Default. Разработайте аннотацию @Default, со следующими характеристиками: • Целью может быть ТИП или ПОЛЕ

• Доступна во время исполнения программы

• Имеет обязательное свойство value типа Class

Проаннотируйте какой-либо класс данной аннотацией..

## Алгоритм решения

* **Создается аннотация @Default с параметром value, который указывает тип значения по умолчанию.**
* **В классе TestDefault создается поле Field, которое аннотировано аннотацией @Default(value = Integer.class).**
* **В методе main создается объект класса TestDefault.**
* **Программа с помощью рефлексии (getDeclaredFields()) получает все поля класса.**
* **Для каждого поля проверяется наличие аннотации @Default с помощью метода isAnnotationPresent().**
* **Если поле аннотировано, программа выводит имя поля и тип, указанный в аннотации..**

## Тестирование

.

## Код программы

Https://github.com/HintMSI/laba\_6

# Задание 4

## Текст задания

@ToString. Разработайте аннотацию @ToString, со следующими характеристиками: • Целью может быть ТИП или ПОЛЕ

• Доступна во время исполнения программы

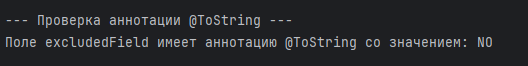
• Имеет необязательное свойство valuec двумя вариантами значений: YES или NO • Значение свойства по умолчанию: YES

Проаннотируйте какой-либо класс данной аннотацией..

## Алгоритм решения

* **В коде создается аннотация @ToString, которая может быть использована для пометки классов и полей.**
* **В классе TestToString определяются два поля: одно аннотировано аннотацией @ToString("NO"), а второе — нет.**
* **В методе main создается объект TestToString.**
* **Программа с помощью рефлексии (getDeclaredFields()) получает все поля класса.**
* **Для каждого поля проверяется наличие аннотации @ToString.**
* **Если поле аннотировано, программа выводит имя поля и значение, указанное в аннотации..**

## Тестирование



## Код программы

Https://github.com/HintMSI/laba\_6

# Задание 5

## Текст задания

@Validate. Разработайте аннотацию @Validate, со следующими характеристиками: • Целью может быть ТИП или АННОТАЦИЯ

• Доступна во время исполнения программы

• Имеет обязательное свойство value, типа Class[]

Проаннотируйте какой-либо класс данной аннотацией..

## Алгоритм решения

1. **Аннотация @Validate**:

- Аннотация @Validate может быть применена как к типам (классам), так и к другим аннотациям. Она содержит массив классов, который будет использоваться для хранения значений.

2. **Класс TestValidate**:

- Класс TestValidate аннотирован аннотацией @Validate, и в параметре value указаны два класса: String и Integer. Эти классы будут выведены на экран.

3. **Получение аннотации**:

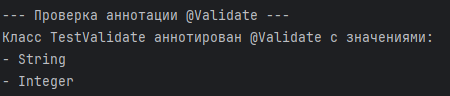
- В методе main получаем аннотацию @Validate с помощью рефлексии, используя TestValidate.class.getAnnotation(Validate.class).

- Если аннотация существует, программа выводит имена классов, указанных в аннотации.

4. **Цикл вывода классов**:

- В цикле выводим имена классов, используя метод getSimpleName(), чтобы отобразить только имя класса, а не полное имя с пакетом.

## Тестирование

.

## Код программы

Https://github.com/HintMSI/laba\_6

# Задание 6

## Текст задания

@Two. Разработайте аннотацию @Two, со следующими характеристиками:

• Целью может быть ТИП

• Доступна во время исполнения программы

• Имеет два обязательных свойства: first типа String и second типа int Проаннотируйте какой-либо класс данной аннотацией..

## Алгоритм решения

1. **Аннотация @Two:**

* **Аннотация @Two имеет два обязательных параметра:**

**a)first — строка.**

**b)second — целое число.**

**- Она применяется к типам (классам), поскольку указано ElementType.TYPE.**

**2. Класс TestTwo:**

**- Класс TestTwo аннотирован аннотацией @Two, где параметр first равен "Hello", а параметр second равен 42.**

**3. Получение аннотации:**

**- В методе main с помощью метода getAnnotation(Two.class) получаем аннотацию @Two для класса TestTwo.**

**- Если аннотация существует (не равна null), программа извлекает значения параметров first и second с помощью методов аннотации first() и second().**

**4. Вывод значений:**

**- Если аннотация присутствует, выводим значения параметров аннотации на экран.**

## Тестирование



## Код программы

Https://github.com/HintMSI/laba\_6

# 

# Задание 7

## Текст задания

@Cache. Разработайте аннотацию @Cache, со следующими характеристиками:

• Целью может быть ТИП

• Доступна во время исполнения программы

• Имеет необязательное свойство value, типа String[]

• Значение свойства по умолчанию: пустой массив

Проаннотируйте какой-либо класс данной аннотацией..

## Алгоритм решения

1. **Аннотация @Cache:**

**- Аннотация @Cache имеет один параметр value, который является массивом строк. По умолчанию этот массив пуст.**

**- Она применяется к типам (классам), так как указан ElementType.TYPE.**

**2. Класс TestCache:**

**- Класс TestCache аннотирован аннотацией @Cache, где параметр value содержит три значения: {"user", "data", "cache"}.**

**3. Получение аннотации:**

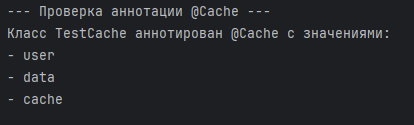
**- В методе main с помощью метода getAnnotation(Cache.class) получаем аннотацию @Cache для класса TestCache.**

**- Если аннотация существует (не равна null), программа извлекает массив строк value с помощью метода value().**

**4. Вывод значений:**

**- Если аннотация присутствует, выводим значения из массива value на экран.**

## Тестирование

.

## Код программы

Https://github.com/HintMSI/laba\_6