|  |  |  |
| --- | --- | --- |
| 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 №6  по дисциплине «Языки программирования»  Вариант 8 | | |
|  | | Работу выполнил  студент группы ПМИ-1,2-2023 2 курса  Хисматов Е.С.  «26» декабря 2024 г. |
| Работу проверил  Ракина В. Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1 4](#_Toc185336405)

[Текст задания 4](#_Toc185336406)

[Алгоритм решения 4](#_Toc185336407)

[Тестирование 5](#_Toc185336408)

[Код программы 5](#_Toc185336409)

[Задание 2 6](#_Toc185336410)

[Текст задания 6](#_Toc185336411)

[Алгоритм решения 6](#_Toc185336412)

[Тестирование 6](#_Toc185336413)

[Код программы 6](#_Toc185336414)

[Задание 3 7](#_Toc185336415)

[Текст задания 7](#_Toc185336416)

[Алгоритм решения 7](#_Toc185336417)

[Тестирование 7](#_Toc185336418)

[Код программы 7](#_Toc185336419)

[Задание 4 8](#_Toc185336420)

[Текст задания 8](#_Toc185336421)

[Алгоритм решения 8](#_Toc185336422)

[Тестирование 8](#_Toc185336423)

[Код программы 8](#_Toc185336424)

[Задание 5 9](#_Toc185336425)

[Текст задания 9](#_Toc185336426)

[Алгоритм решения 9](#_Toc185336427)

[Тестирование 9](#_Toc185336428)

[Код программы 9](#_Toc185336429)

[Задание 6 10](#_Toc185336430)

[Текст задания 10](#_Toc185336431)

[Алгоритм решения 10](#_Toc185336432)

[Тестирование 10](#_Toc185336433)

[Код программы 10](#_Toc185336434)

[Задание 7 11](#_Toc185336435)

[Текст задания 11](#_Toc185336436)

[Алгоритм решения 11](#_Toc185336437)

[Тестирование 11](#_Toc185336438)

[Код программы 11](#_Toc185336439)

[Задание 8 12](#_Toc185336440)

[Текст задания 12](#_Toc185336441)

[Алгоритм решения 12](#_Toc185336442)

[Тестирование 12](#_Toc185336443)

[Код программы 12](#_Toc185336444)

# Задание 1

## Текст задания

Кэширование.

Разработайте обобщенный утилитный метод cache. Метод принимает объект и возвращает такую версию данного объекта, у которого вызовы всех методов без параметров (доступных по текущей ссылке) кэшированы. При вызове кэшированного метода необходимо проанализировать состояние объекта:

• Если метод вызывается впервые после создания кэшированного объекта - то он вызывается обычным образом.

• Если метод вызывается повторно и с момента прошлого вызова внесены изменения в состояние объекта – то он вызывается обычным образом.

• Если метод вызывается повторно и объект не был изменен с момента прошлого вызова – то вместо вызова метода необходимо вернуть то же значение, что возвращал метод при предыдущем вызове.

При разработке метода необходимо учесть следующее:

• Метод принимает объект обобщенного типа T и возвращает объект того же типа.

• Если объект приходит по интерфейсной ссылке – то кэшируются все методы этой ссылки, а объект должен реализовывать этот интерфейс

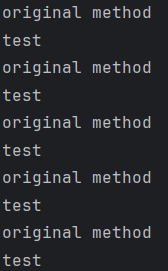
• Если объект приходит по классовой ссылке – то кэшируются только публичные методы, а объект должен быть подклассом исходного класса

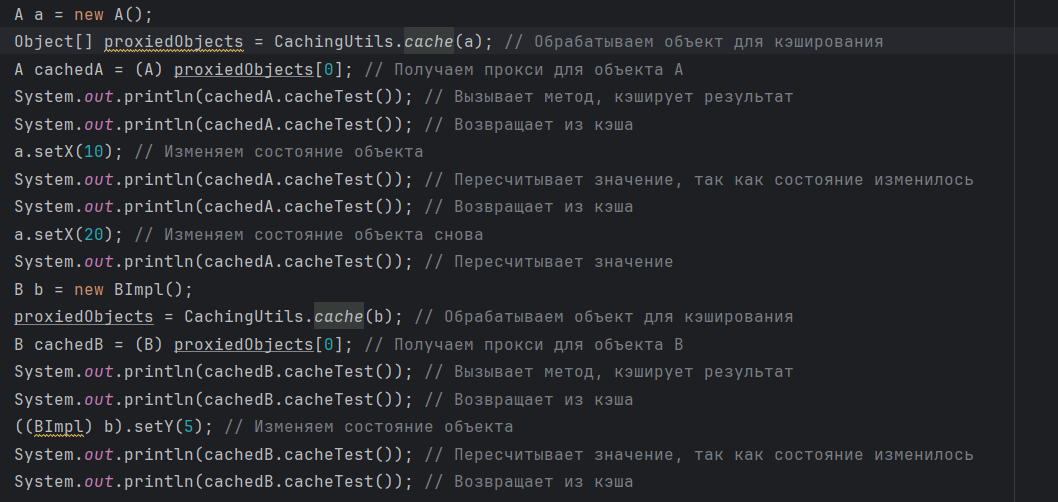
• Кэшированию подлежат только методы без входных параметров

## Алгоритм решения

Создаем прокси-объект для исходного объекта, который перехватывает вызовы его методов. Если метод вызывается впервые или состояние объекта изменилось, метод вызывается напрямую и результат сохраняется в кэш. Если метод вызывается повторно и состояние объекта не изменилось, возвращается закэшированный результат. Для отслеживания состояния объекта сохраняются значения его полей, которые проверяются перед каждым вызовом метода. Кэшируются только методы без параметров, и если объект реализует интерфейс, кэшируются все методы этого интерфейса; если объект передается по классовой ссылке, кэшируются только публичные методы. Кэширование настраивается через аннотацию @Cache.

## Тестирование





## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 2

## Текст задания

@Invoke.

Разработайте аннотацию @Invoke, со следующими характеристиками:

• Целью может быть только МЕТОД

• Доступна во время исполнения программы

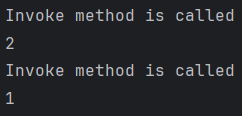
• Не имеет свойств

Проаннотируйте какой-либо метод данной аннотацией

## Алгоритм решения

Создание аннотации @Invoke, которая будет доступна только для методов и не имеет свойств. Аннотация должна быть доступна во время выполнения с помощью @Retention(RetentionPolicy.RUNTIME). Аннотируйте какой-либо метод с аннотацией @Invoke.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 3

## Текст задания

@Default.

Разработайте аннотацию @Default, со следующими характеристиками:

• Целью может быть ТИП или ПОЛЕ

• Доступна во время исполнения программы

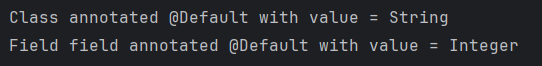
• Имеет обязательное свойство value типа Class

Проаннотируйте какой-либо класс данной аннотацией.

## Алгоритм решения

Создание аннотации @Default, которая может быть использована на типах (классах) и полях. У аннотации должно быть обязательное свойство value, которое будет хранить класс (тип). Аннотация должна быть доступна во время выполнения. Аннотируйте какой-либо класс и поле с аннотацией @Default.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# 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

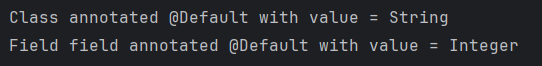
• Значение свойства по умолчанию: YES

Проаннотируйте какой-либо класс данной аннотацией.

## Алгоритм решения

Создание аннотации @ToString, которая может быть использована на типах (классах) и полях. У аннотации должно быть необязательное свойство value с двумя возможными значениями: YES и NO. Значение по умолчанию — YES. Обработайте аннотированные классы или поля, генерируя строковое представление в зависимости от значения value.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 5

## Текст задания

@Validate.

Разработайте аннотацию @Validate, со следующими характеристиками:

• Целью может быть ТИП или АННОТАЦИЯ

• Доступна во время исполнения программы

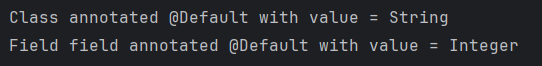
• Имеет обязательное свойство value, типа Class[]

Проаннотируйте какой-либо класс данной аннотацией.

## Алгоритм решения

Создание аннотации @Validate, которая может быть использована на типах и аннотациях. У аннотации должно быть обязательное свойство value, которое является массивом классов (Class[]). Аннотация должна быть доступна во время выполнения.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 6

## Текст задания

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

• Целью может быть ТИП

• Доступна во время исполнения программы

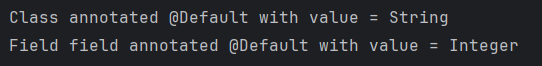
• Имеет два обязательных свойства: first типа String и second типа int

Проаннотируйте какой-либо класс данной аннотацией.

## Алгоритм решения

Создание аннотации @Two, которая может быть использована только на типах. Аннотация должна иметь два обязательных свойства: first типа String и second типа int. Аннотация должна быть доступна во время выполнения.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# 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

Создание аннотации @Cache, которая может быть использована только на типах. Аннотация должна иметь необязательное свойство value типа String[], которое содержит список методов для кэширования. Если свойство не задано, кэширование выполняется для всех методов без параметров. Обрабатывайте аннотированные классы в методе, где проверяете, какие методы нужно кэшировать.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 8

## Текст задания

Кэширование.

Измените утилитный метод cache из задачи 1.3. Теперь метод должен принимать произвольный набор объектов по ссылкам типа Object. Кэширование выполняется только для тех объектов, которые проаннотированы с помощью @Cache из предыдущей задачи, при этом:

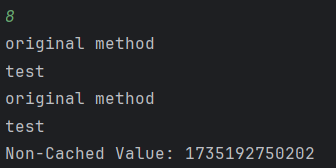
• Если для аннотации не заданы параметры, то кэширование выполняется как обычно.

• Если для аннотации задан набор строк, то их необходимо учитывать как названия методов подлежащих кэшированию. Остальные методы – не кэшируются.

## Алгоритм решения

Кэширование выполняется только для объектов, аннотированных с помощью @Cache. Если аннотация содержит список строк, то кэшируются только методы, чьи имена указаны в списке, в противном случае кэшируются все методы без параметров.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)

# Задание 9

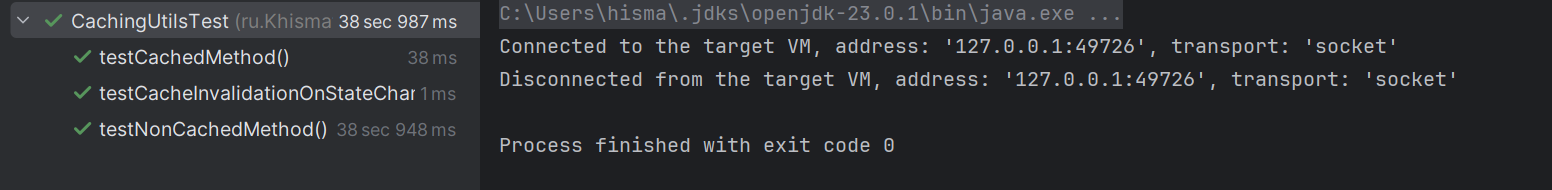
## Текст задания

Напишите тест, используя фреймворк juntil, к методу из задания 3. Пояснения к тесту и объекту тестирования отобразить в комментариях кода и в отчете.

## Алгоритм решения

Тестируем, что метод с аннотацией @Cache правильно кэшируется, что кэш сбрасывается при изменении состояния объекта, а также проверяем, что методы, не указанные в аннотации, не кэшируются. В тесте создается объект, обернутый в кэширующий прокси, и с помощью рефлексии вызываются тестируемые методы. Далее проверяется, что методы с аннотацией возвращают одинаковые результаты при повторных вызовах, а методы без аннотации — разные.

## Тестирование



## Код программы

[https://github.com/NTHrdd/Lab6](https://github.com/NTHrdd/Lab3)