

Properties

Java Collections
1 уровень, 9 лекция

ОТКРЫТА

— Сегодня мы изучаем еще одну новую и интересную тему – **свойства (properties)**.

```
messages_ru.properties x
1 sample2.whatIsYourName=Здравствуйте, как Вас зовут?
2 sample2.enterYourName=Введите имя...
3 sample2.save=Сохранить
4 sample2.hello=Здравствуйте
```

В Java принято делать программы гибкими и легко настраиваемыми. Иногда еще говорят легко конфигурируемыми (от конфигурация).

Например, твоя программа раз в час копирует файлы из некоторой директории, архивирует их и отправляет тебе на email. Для этого программе надо знать директорию, откуда брать файлы и email, куда их посылать. Такие данные принято хранить не в коде программы, а в отдельных файлах свойств – **properties**-файлах.

Данные в таком файле хранятся в виде пар ключ-значение, разделенные знаком равно.

| Пример | |
|--------|-----------------------------|
| 1 | Файл data.properties |
| 2 | directory=c:/text/downloads |
| 3 | email=zapp@javarush.ru |

Слева от знака равно – имя (ключ), справа – значение.

— Т.е. это что-то типа текстового представления HashMap?

— В общем-то, да.

Для удобной работы с такими файлами в Java есть специальный класс – Properties. Класс Properties унаследован от Hashtable<Object,Object>. Его даже можно рассматривать как HashTable, который умеет загружать себя из файла.

Вот его методы:

| Метод | Описание |
|----------------------------------|---|
| void load(Reader reader) | Загружает свойства из файла, представленного объектом Reader |
| void load(InputStream inStream) | Загружает свойства из файла, представленного объектом InputStream |
| void loadFromXML(InputStream in) | Загружает свойства из XML-файла |

| | |
|---|---|
| <code>String getProperty(String key)</code> | Возвращает значение свойства (строку) по ключу |
| <code>String getProperty(String key, String defaultValue)</code> | Возвращает значение свойства по ключу или defaultValue, если такого ключа нет |
| <code>Set<String> stringPropertyNames()</code> | Возвращает список всех ключей |

Т.е. фактически тебе нужно выполнить всего две операции – загрузить в объект **Properties** данные из какого-нибудь файла, а затем получить эти свойства с помощью метода **getProperty()**. Ну и не забывай, что можешь пользоваться объектом **Properties** как **HashMap**.

Вот тебе пример:

| Код | |
|-----|--|
| 1 | <code>//файл, который хранит свойства нашего проекта</code> |
| 2 | <code>File file = new File("c:/data.properties");</code> |
| 3 | |
| 4 | <code>//создаем объект Properties и загружаем в него данные из файла.</code> |
| 5 | <code>Properties properties = new Properties();</code> |
| 6 | <code>properties.load(new FileReader(file));</code> |
| 7 | |
| 8 | <code>//получаем значения свойств из объекта Properties</code> |
| 9 | <code>String email = properties.getProperty("email");</code> |
| 10 | <code>String directory = properties.getProperty("directory");</code> |
| 11 | |
| 12 | <code>//получаем числовое значение из объекта Properties</code> |
| 13 | <code>int maxFileSize = Integer.parseInt(properties.getProperty("max.size", "10000"));</code> |

— Ага. Т.е. мы создаем объект Properties, затем передаем в него файл. В метод load, а затем просто вызываем getProperty. Так?

— Ага.

— А ты еще говорил, что им можно пользоваться как HashMap? Что ты имел в виду?

— Класс Properties унаследован от Hashtable, а это – тот же HashMap, просто все методы его синхронизированы. Вот так можно просто вывести на экран все значения из файла свойств:

| Код | |
|-----|--|
| 1 | <code>//получаем файл со свойствами</code> |
| 2 | <code>File file = new File("c:/data.properties");</code> |
| 3 | |
| 4 | <code>//создаем объект Properties и загружаем в него данные из файла.</code> |
| 5 | <code>Properties properties = new Properties();</code> |
| 6 | <code>properties.load(new FileReader(file));</code> |
| 7 | |
| 8 | <code>//проходимся по всем ключам и печатаем все их значения на консоль</code> |
| 9 | <code>for (String key : properties.stringPropertyNames())</code> |
| 10 | <code>{</code> |
| 11 | <code> System.out.println(properties.get(key));</code> |
| 12 | <code>}</code> |

НАЧАТЬ ОБУЧЕНИЕ

−

+132

+

Комментарии (84)

популярные

новые

старые

JavaCoder

Введите текст комментария

Иlia lenskii

Уровень 22, Санкт-Петербург

3 июля, 14:30

...

Что касается примеров кода, то есть вот это [getResourceAsStream](#) + там же ссылка на docs.oracle.com

1

MyClass.getResourceAsStream("my.properties");

Где *MyClass* имя вашего класса, далее *class* и вызов непосредственно *getResourceAsStream*

У меня это выглядит вот так вот

1

InputStream inpPror = Myprop.class.getResourceAsStream("../prp.properties");

2

if(inpPror != null) {

3

Properties properties = new Properties();

4

properties.load(inpPror);

5

}

Экономия на спичках

Ответить

−

0

+

Артём

Уровень 24, Омск, Россия

21 февраля, 18:28

...

а чем это отличается от сериализации?

Ответить

−

0

+

LuneFox

инженер по сопровождению в BIFIT

EXPERT

19 декабря 2021, 14:44

...

Как мёд для ушей (глаз), я тоже все настраиваемые параметры всегда хранил в отдельных классах, чтобы не выискивать их в коде (сам додумался), а тут даже специальный файлик с ними забубенили.

Ответить

−

+2

+

Alukard

Vampire hunter в The Hellsing

EXPERT

24 декабря 2021, 16:13

...

Ты что тут забыл, иди на работу уже)

Ответить

−

+4

+

LuneFox

инженер по сопровождению в BIFIT

EXPERT

24 декабря 2021, 18:06

...

Сарказм принят)

Ответить

−

0

+

Роман Кончалов

Уровень 28, Россия

EXPERT

19 января, 17:00

...

Какие, например, параметры?

Ответить

−

0

+

LuneFox

инженер по сопровождению в BIFIT

EXPERT

20 января, 01:59

...

Да какие угодно. Например, дефолтный размер окна, или шрифт, который использует интерфейс, или путь к папке, откуда программа по умолчанию берёт файлы для работы.

Ответить

−

+1

+

Роман Кончалов

Уровень 28, Россия

EXPERT

20 января, 11:18

...

Но ведь если эти параметры используются только в одно месте, то достаточно объявить их в начале целевого класса как private final. Или были какие-то параметры, которые в нескольких местах использовались?

Я про то и говорю, что все параметры, которые могли бы быть записаны, как `private final` в куче разных классов я просто беру и выношу в отдельный класс с `public static final` полями, чтобы потом не искать их внутри кода в разных местах. Например, когда писал игры здесь, все строки, которые выводятся на экран, хранил в отдельном классе `Strings`. Например:

```
1 public static final String GAME_OVER_MESSAGE = "Вы проиграли!";
```

И если где-то вдруг хотел поправить описание или фразу, я просто лез в `Strings`, а не искал их в куче функционального кода с механикой игры.

Никто не мешает хранить разные настраиваемые параметры разных классов точно так же.

Ответить

-

+3

+

Сергей Уровень 22 9 февраля, 05:58

...

а еще он хранится отдельно от собранного проекта, как можно хранить и иные конфо файлы. И поэтому их можно менять в любое время, если у тебя не хардкодед конечно же кругом в проекте.

Ответить

-

+1

+

Hidden #213 Уровень 13 (Forever&Ever) 26 февраля, 17:41

...

Нет, тебе точно пора уже на работу)))
Я почти на том же уровне, но нифига не понимаю о чём вы говорите. Какие-то настраиваемые `final` параметры и специальный файл для них.

Что?! Зачем?! И ты ещё говоришь, что сам к этому пришёл, без этой лекции...

Ответить

-

0

+

LuneFox инженер по сопровождению в BIFIT EXPERT 26 февраля, 18:57

...

Дык чего тут особенного, тема лекции как раз про это самое :)
Пока я не мог пользоваться `properties`, их роль заменял отдельный класс со статическими переменными.

Например, в `properties` можно положить такие данные:

```
1 game.message.start=Игра началась!  
2 game.message.finish=Игра закончилась!
```

А потом подгружать этот файл и обращаться к полям `properties`.

Я делал так прямо в классе:

```
1 public static final String GAME_MESSAGE_START = "Игра началась!";  
2 public static final String GAME_MESSAGE_FINISH = "Игра закончилась!";
```

Ну, и обращался не так

```
1 printMessage(properties.get("game.message.start"));
```

А вот так

```
1 printMessage(Strings.GAME_MESSAGE_START);
```

Только и всего :)
А сам я к этому пришёл по нужде, потому что устал искать строки с текстом в куче кода, чтобы исправить опечатку или что-то перефразировать. Тем более если строк много в разных классах.

Ответить

-

+1

+

Ars Уровень 41 15 ноября 2021, 17:21

...

Инфа по задаче:
Среднее количество попыток для этой задачи 2.24. Всего эту задачу решили 7291 учеников.

Ответить

-

0

+

LuneFox инженер по сопровождению в BIFIT EXPERT 19 декабря 2021, 14:42

...

Всё хорошо, только это не задача, а лекция.

Ответить

-

+3

+

Ars Уровень 41 19 декабря 2021, 15:44

...

От блин.
Но в задачу копировать не буду. Уже не актуальная инфа.

Ответить

-

0

+

Anonymous #2620027 Уровень 47 13 ноября 2021, 17:46

...

Properties уже есть в Core. Use Collections, use Multithreading, генераторы не добавляю

НАЧАТЬ ОБУЧЕНИЕ

LuneFoxинженер по сопровождению в BIFITEXPERT19 декабря 2021, 14:42...

Я даже рад этому, потому что курс прохожу медленно, и всё забывается. Пока мой мозг чувствует сопротивление при решении задач, курс всё делает верно.

Ответить+2

ArsУровень 4119 декабря 2021, 15:48...

Чтобы не забывалось, это надо использовать. И/или конспектировать.
По идее так и заявлено, что в курсе много задачи и поэтому, всё хорошо запомнится. Только увы, на многие либо нет задач вообще, либо их пара штук. Вот если бы те же properties нет нет, да и всплывали бы в задачах, то они бы и не были забыты.
А вот такое повторение теории без прилагающейся практики не спасёт ситуацию. Всё равно забудется, если не использовать.

Ответить+1

LuneFoxинженер по сопровождению в BIFITEXPERT19 декабря 2021, 16:39...

Ещё кстати хорошая практика для запоминания - это чётко понимать, что и как происходит, тогда и заучивать насильно ничего не придётся. Не знаю, какой пример даже привести. Например, если однажды выучить последовательность цифр от 1 до 10, то не нужно для каждого этажа в 10-этажном доме запоминать, какой находится выше, а какой ниже, а ещё отдельно помнить, что ниже 1 этажа и выше 10 этажа этажей нет, ты просто пользуешься здравым смыслом. А забыть здравый смысл куда сложнее, чем таблицу с информацией :)

Ответить0

ArsУровень 4119 декабря 2021, 17:23...

Понимать - не значит запомнить.
Я могу прекрасно понять, как работает какой-нибудь LinkedHashMap, понять какие у него методы, чем он отличается от обычных HashMap и LinkedList. Но если я его не использовал достаточное время чтобы он отложился у меня в голове, то когда он мне понадобится, я про просто не вспомню. Учитывая, что разных классов только в JDK, наверное, сотни и есть ещё десятки фреймворков со своими классами, то какими здравомысленными эти классы не были просто понять их чтобы запомнить - не выйдет.

Ответить+1

LuneFoxинженер по сопровождению в BIFITEXPERT19 декабря 2021, 17:58...

Тут уж не поспоришь. Помнить подробности применения сотен классов с тысячами методов просто нереально.

Ответить0

E KУровень 41, Краснодар, Россия4 июня 2021, 17:18...

Собираю небольшую библиотеку по Java
[ссылка](#)

Если кому есть чего добавить -> welcome
Буду признателен за полезный и интересный контент по теме

Ответить+48

РустемУровень 24, Уфа, Россия25 июня 2021, 23:31...

возможно вам пригодится HeadFirst по SQL в библиотечку ;) https://disk.yandex.ru/i/HegOIIOzlx_11g

Ответить0

E KУровень 41, Краснодар, Россия28 июня 2021, 00:30...

Большое спасибо!

Положил на полочку - такое лишним точно не будет)

Ответить0

DenisУровень 40, Москва, Russia21 июля 2021, 07:37...

Спасибо, человек

Ответить0

КристиУровень 31, Минск, Беларусь23 июля 2021, 12:41...

Спасибо!!! Много полезного в одном месте)))

Ответить0

АлександрУровень 41, Екатеринбург29 июля 2021, 12:20...

Спасибо добрый человек. Дай бог тебе здоровья!

Ответить0

KesЧайник в Банк1 августа 2021, 09:37...

E K
Кросаучег. Я быренько всё скачал.

Ответить

НАЧАТЬ ОБУЧЕНИЕ

Кому интересно - качаю книги по Java и программированию из канала [ссылка](#)

Ответить

+1

Анастасия Ольшанская

Уровень 31

7 ноября 2021, 08:27

[Пара книг по Git \(тоже пригодится в коллекцию\)](#)

Ответить

+3

Анастасия Ольшанская

Уровень 31

7 ноября 2021, 10:13

Еще в копилочку

[Марко_Фаэлла_Стильный_Java_Код,_который_работает_всегда_и_езде](#)

[Элегантные объекты. Java Edition](#)

Ответить

+2

Максим Дудин

Уровень 38, Калининград

13 декабря 2021, 19:19

Круть просто... дай бог тебе здоровья

Ответить

0

мистер т

Уровень 35, Москва

6 апреля 2021, 21:08

```
int maxFileSize = Integer.parseInt(properties.getProperty("max.size", "10000"));
если в проперти нет max.size возвращает 2е значение (10000)
```

Ответить

+13

Dmitry Vasilyev

Уровень 26, Саратов, Россия

4 апреля 2021, 07:52

В JavaRush принято делать лекции полезными (от польза) и интересными (от интерес).

Ответить

+2

Алексей

Уровень 37, Санкт-Петербург

10 февраля 2021, 18:51

мне вот тут понравилось, с примерами [тыц](#)

Ответить

+1

Алексей Мирный

Backend Developer

14 декабря 2020, 11:39

Я уже продвинутый ДжаваРашер и знаю, что если дают лекцию с хреновым объяснением, то на следующей странице будут супер-хард задачи которые непонятно как решать))

Ответить

+33

Максим Дудин

Уровень 38, Калининград

13 декабря 2021, 19:22

Да уже прошлый стек задач напряг... если задача решается за 20 сек, это не к добру, дальше жди, что-нибудь страшное

Ответить

0

Показать еще комментарии

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



НАЧАТЬ ОБУЧЕНИЕ

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

 Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

 **ДОСТУПНО В**
Google Play

 Загрузите в
App Store

