

*Посвящаем эту книгу  
Джойс Нафтален, Лайонелю Нафталену,  
Адаму Уодлеру и Леоре Уодлер  
Морис Нафтален, Филип Уодлер*



# Оглавление

Об авторах.....	14
Об иллюстрации на обложке.....	15
<b>Предисловие от издательства .....</b>	<b>16</b>
<b>Предисловие.....</b>	<b>17</b>
Благодарности за второе издание .....	18
Для кого написана эта книга .....	18
Где взять примеры кода .....	19
Графические выделения .....	19
Утверждения .....	21
Как с нами связаться .....	21
Предисловие к первому изданию.....	21
Благодарности, включенные в первое издание .....	23
<b>ЧАСТЬ I. ДЖЕНЕРИКИ.....</b>	<b>25</b>
<b>Глава 1. Введение.....</b>	<b>26</b>
Параметризованные типы .....	26
Дженерики и шаблоны .....	29
Параметризованные методы и varargs-аргументы.....	29
Примитивные и ссылочные типы .....	32
Заключение .....	34
<b>Глава 2. Подтипизация и джокеры.....</b>	<b>35</b>
Подтипизация и принцип подстановки .....	35
Джокеры .....	37
Джокеры с extends .....	38
Джокеры с super .....	39
Принцип получения и вставки .....	41
Массивы.....	45
Ограниченные или неограниченные? .....	48
Использование неограниченного джокера .....	48
Использование ограниченного джокера .....	49
Запоминание джокера.....	49
Ограничения на джокеры .....	51
Создание экземпляра .....	51
Вызовы обобщенных методов .....	52
Супертипы.....	53
Заключение .....	53

<b>Глава 3. Сравнение и границы .....</b>	<b>54</b>
Интерфейс Comparable .....	54
Контракт интерфейса Comparable .....	55
Согласованность с equals .....	56
Сравнение целочисленных значений .....	57
Максимальный элемент коллекции .....	57
Фруктовый пример .....	60
Интерфейс Comparator .....	63
Методы интерфейса Comparator .....	65
Типы перечислений .....	68
Множественные границы .....	70
Мостовые методы .....	72
Ковариантное переопределение .....	74
Заключение .....	75
<b>Глава 4. Объявления .....</b>	<b>76</b>
Конструкторы .....	76
Статические члены .....	77
Вложенные классы .....	79
Как работает стирание .....	81
Заключение .....	83
<b>Глава 5. Сберегаемые и несберегаемые типы .....</b>	<b>84</b>
Сберегаемые типы .....	85
Тесты экземпляров и приведения .....	85
Непроверенные приведения .....	90
Обработка исключений .....	90
Параметризованные типы и массивы .....	92
Создание параметризованного массива .....	94
Принцип правдивости рекламы .....	96
Как создавать массивы .....	97
Массив рождает массив .....	98
Классная альтернатива .....	99
Как определить ArrayList .....	100
Принцип непристойного обнажения .....	103
Создание массива и vararg-аргументы .....	105
Где требуются сберегаемые типы .....	108
О дизайне дженериков в Java .....	108
Стирание .....	108
Неограниченные джокеры .....	109
Массивы .....	109
Arrays::copyOf .....	111
Заключение .....	112
<b>Глава 6. Рефлексия .....</b>	<b>113</b>
Параметризованные типы для рефлексии .....	114
Рефлектированные типы являются сберегаемыми .....	116

Рефлексия для примитивных типов.....	117
Обобщенная библиотека рефлексии.....	118
Рефлексия для параметризованных типов.....	120
Рефлексия параметризованных типов .....	122
Заключение .....	125
<b>Глава 7. Эффективные дженерики Java.....</b>	<b>126</b>
Искореняйте предупреждения о невозможности проверки .....	126
Обеспечивайте типобезопасность при вызове недоверенного кода .....	128
Специализируйте для создания сберегаемых типов .....	130
Избегайте бессмысленных переменных-типов .....	132
Используйте обобщенные вспомогательные методы для запоминания джокера .....	132
Приводите через простые типы, когда необходимо .....	133
Используйте параметризованные типы массивов с осторожностью .....	134
Используйте маркеры типов для передачи информации о типе времени выполнения.....	136
Заключение .....	137
<b>ЧАСТЬ II. КОЛЛЕКЦИИ.....</b>	<b>139</b>
<b>Глава 8. Основные интерфейсы каркаса коллекций Java .....</b>	<b>141</b>
Использование разных типов коллекций.....	142
Set .....	142
List .....	143
Map .....	143
SequencedMap .....	144
Queue .....	145
Упорядоченные коллекции.....	145
SequencedCollection .....	146
SequencedSet и NavigableSet .....	147
Deque .....	147
SequencedMap и NavigableMap.....	147
Заключение .....	147
<b>Глава 9. Предварительные сведения.....</b>	<b>148</b>
Итерируемые объекты и итераторы .....	148
Реализации.....	151
Представления .....	153
Производительность .....	154
Память .....	155
Счетчик команд и нотация O большое .....	157
Неизменяемость и немодифицируемость .....	159
Контракты .....	161
Организация по содержимому .....	163

Лямбда-выражения и потоки .....	164
Параллельные потоки .....	165
Коллекции и потокобезопасность .....	166
Синхронизация и унаследованные коллекции .....	168
Синхронизированные коллекции и итераторы с быстрым отказом .....	169
Конкурентные коллекции .....	170
Итераторы .....	172
Заключение .....	173

## **Глава 10. Интерфейс Collection..... 174**

Методы Collection .....	174
Добавление элементов .....	174
Удаление элементов .....	175
Опрос содержимого коллекции .....	175
Обеспечение доступности содержимого коллекции для дальнейшей обработки .....	176
Использование методов коллекций .....	178
Добавление элементов .....	180
Удаление элементов .....	181
Опрос содержимого коллекции .....	183
Обеспечение доступности содержимого коллекции для последующей обработки .....	183
Реализация Collection .....	186
Конструкторы коллекций .....	186
Заклучение .....	187

## **Глава 11. Интерфейс SequencedCollection..... 188**

Методы SequencedCollection .....	188
Добавление элементов .....	189
Инспектирование элементов .....	189
Удаление элементов .....	189
Генерирование обращенного представления .....	190
Заклучение .....	190

## **Глава 12. Множества..... 191**

Определение множества: отношения эквивалентности .....	192
Реализации интерфейса Set .....	193
HashSet .....	193
CopyOnWriteArraySet .....	196
EnumSet .....	197
UnmodifiableSet .....	199
Представления отображений в виде множеств .....	200
SequencedSet .....	201
LinkedHashSet .....	201
NavigableSet .....	203
Методы NavigableSet .....	204

TreeSet .....	209
ConcurrentSkipListSet .....	212
Сравнение реализаций множеств .....	214
Заключение .....	215
<b>Глава 13. Очереди .....</b>	<b>216</b>
Методы интерфейса Queue .....	217
Добавление элемента в очередь .....	217
Извлечение элемента из очереди .....	218
Использование методов очереди .....	218
Реализации Queue .....	219
PriorityQueue .....	220
ConcurrentLinkedQueue .....	223
BlockingQueue .....	223
Методы BlockingQueue .....	224
Использование методов BlockingQueue .....	225
Реализации BlockingQueue .....	228
Deque .....	233
Методы Deque .....	233
Реализации Deque .....	234
BlockingDeque .....	237
Сравнение реализаций очереди .....	238
Заключение .....	240
<b>Глава 14. Списки .....</b>	<b>241</b>
Методы интерфейса List .....	241
Методы позиционного доступа .....	241
Методы поиска .....	242
Методы генерирования представлений .....	242
Методы обхода списка .....	242
Методы, унаследованные от SequencedCollection .....	244
Фабричные методы .....	244
Использование методов List .....	244
Использование методов диапазонного представления и итератора .....	247
Реализации интерфейса List .....	248
ArrayList .....	248
LinkedList .....	250
CopyOnWriteArrayList .....	250
UnmodifiableList .....	251
Сравнение реализаций списка .....	252
Заключение .....	253
<b>Глава 15. Отображения .....</b>	<b>254</b>
Методы интерфейса Map .....	254
Операции, подобные Iterable .....	255
Операции, подобные Collection .....	255
Получение представлений ключей, значений и записей .....	256

Составные операции .....	256
Фабричные методы .....	259
Интерфейс Map.Entry .....	260
Использование методов Map .....	260
Реализации интерфейса Map .....	262
IdentityHashMap .....	265
UnmodifiableMap .....	268
SequencedMap .....	269
Методы SequencedMap .....	269
LinkedHashMap .....	270
NavigableMap .....	273
Получение компаратора .....	273
Получение диапазонных представлений .....	273
Получение ближайших соседей .....	274
Другие представления .....	275
TreeMap .....	275
ConcurrentMap .....	276
ConcurrentHashMap .....	276
ConcurrentNavigableMap .....	277
ConcurrentSkipListMap .....	277
Сравнение реализаций отображения .....	278
Заключение .....	278

## **Глава 16. Класс Collections..... 280**

Обобщенные алгоритмы .....	280
Изменение порядка элементов списка .....	280
Изменение содержимого списка .....	281
Нахождение экстремальных значений в коллекции .....	282
Нахождение конкретных значений в списке .....	283
Фабрики коллекций .....	283
Обертки .....	285
Синхронизированные коллекции .....	285
Немодифицируемые коллекции .....	286
Проверяемые коллекции .....	286
Прочие методы .....	287
addAll .....	287
asLifoQueue .....	287
disjoint .....	287
enumeration .....	288
frequency .....	288
list .....	288
newSequencedSetFromMap и newSetFromMap .....	288
reverseOrder .....	289
Заключение .....	290

<b>Глава 17. Наставление по использованию</b>	
<b>каркаса коллекций Java .....</b>	<b>291</b>
Избегайте анемичных моделей предметной области .....	291
Не забывайте о «владельцах» коллекций.....	293
Отдавайте предпочтение неизменяемым объектам	
в качестве элементов множества или ключей отображения.....	297
Соблюдайте баланс интересов клиента и библиотеки	
при проектировании API .....	299
Пользуйтесь возможностями записей .....	300
Отдавайте предпочтение записям	
перед параллельными списками.....	301
Используйте записи в качестве составных ключей .....	303
Управляйте изменяемостью записей.....	304
Избегайте унаследованных реализаций.....	305
Избегайте синхронизированных оберток коллекций .....	306
Избегайте LinkedList.....	308
Настраивайте коллекции с помощью абстрактных классов .....	309
Заключение .....	312
<b>Глава 18. Ретроспективный взгляд на дизайн .....</b>	<b>313</b>
Фундаментальные проблемы дизайна каркаса коллекций.....	313
Желательные характеристики каркаса коллекций .....	314
Обеспечение немодифицируемости	
путем создания подтипов.....	315
Ограничения подтипов.....	316
Не только статическая типизация.....	318
Объединение и разделение .....	320
Резюме.....	321
Значения null .....	322
Несогласованность с equals.....	324
Сравнение Object и E.....	326
Конкурентная модификация .....	327
Послесловие .....	331
Литература .....	333
Предметный указатель.....	335

# Об авторах

**Морис Нафтален** имеет за плечами более 50 лет работы в ИТ-индустрии в качестве разработчика, проектировщика, архитектора, руководителя, преподавателя и автора. Он использует и преподает Java со времен JDK 1.0. Он обладатель звания Java Champion, автор книги «Mastering Lambdas» (2015), часто выступает на конференциях по всему миру. Он дезорганизует ежегодную неконференцию на тему Java – JAlba.

**Филип Уодлер** – профессор теоретической информатики в Эдинбургском университете, Шотландия, занимается исследованиями в области функционального и логического программирования и проектирования языков программирования. Являлся главным проектировщиком языка программирования Haskell и одним из проектировщиков GJ, работы, положенной в основу дженериков Java. Профессор Уодлер защитил докторскую диссертацию по информатике в университете Карнеги–Меллона.

**Стюарт Маркс** – руководитель проекта JDK Core Libraries в группе платформы Java в компании Oracle. В настоящее время занимается сопровождением каркаса коллекций Java. Поскольку его второе я – «доктор Депрекатор», он также работает над механизмом вывода из эксплуатации в Java. У него за плечами свыше 30 лет работы над разработками платформенных программных продуктов в области оконных систем, интерактивной графики, а также мобильных и встраиваемых систем. Имеет степень магистра информатики, полученную в Стэнфордском университете.

## Об иллюстрации на обложке

На обложке книги изображен аллигатор. Аллигаторы встречаются только в южных частях США и в Китае. В Китае они редки и водятся только в бассейне реки Янцзы. Как правило, аллигаторы не переносят соленую воду, поэтому обитают в пресноводных прудах, болотах и других водоемах.

Новорожденные аллигаторы совсем небольшие, около 15 см длиной. Но в первые годы жизни они растут очень быстро – по 30 см в год. Длина взрослой самки примерно 270 см, а вес – от 68 до 91 кг. Длина взрослого самца примерно 330 см, а вес – от 159 до 181 кг. Самый крупный аллигатор, найденный в Луизиане в начале 1900-х годов, достигал в длину 584 см. Основной внешний признак аллигатора – короткая широкая морда. Шкура взрослого аллигатора серо-черного цвета, при намокании становится черной. Брюхо белое. На спине у молодых аллигаторов видны желтые и белые полосы. Форма морды и цвет шкуры – физические признаки, отличающие аллигаторов от крокодилов, у которых морда тонкая, а цвет рыжевато-коричневый.

Аллигаторы ведут преимущественно ночной образ жизни, они охотятся и питаются после захода солнца. Они хищники, в рационе которых разнообразная пища: черепахи, рыбы, лягушки, птицы, змеи, небольшие млекопитающие и даже менее крупные аллигаторы. У взрослого аллигатора практически нет врагов – кроме человека.

# Предисловие от издательства

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпустить книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг – возможно, ошибку в основном тексте или программном коде, – мы будем очень благодарны, если вы сообщите нам о ней. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Если вы найдете какие-либо ошибки в коде, пожалуйста, сообщите о них главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com), и мы исправим это в следующих тиражах.

## Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

## О переводе терминов

В данной книге говорится как о парадигме параметризованных типов в целом, так и о дженериках как о механизме реализации типобезопасности в Java. Но авторы предпочитают использовать общий термин `generic`, что создает некоторую путаницу. В переводе применяются оба варианта, чтобы обеспечить плавный переход от общего понятия типобезопасности к дженерикам и коллекциям Java.

# Предисловие

Второе издание этой книги значительно переработано для версии Java 21. Главным побудительным мотивом для него было появление последовательных коллекций и первые последствия потоков и лямбда-выражений, поэтому еще до начала работы я ожидал, что в основном изменениям придется подвергнуть часть II – из-за последовательных коллекций. Но, как оказалось, я сильно недооценил масштаб изменений, произошедших с Java за последние 20 лет, и в итоге понадобилось вносить многочисленные поправки и в часть I тоже. Язык изменился; теперь он поддерживает записи, вывод типов локальных переменных, выражения сопоставления с образцом и параметризованные типы в выражениях сравнения типов. Кроме того, большую часть примеров в части I пришлось переработать с учетом изменений в платформенных библиотеках, в том числе объявления нерекомендуемыми конструкторов `Integer`, а также появления статических методов в интерфейсе `Comparator`, немодифицируемых коллекций и потоков.

Важной причиной для написания второго издания стало документирование некоторых выработанных сообществом Java с 2005 года приемов работы с параметризованными типами и коллекциями. По большей части они собраны в ретроспективном разделе «О дизайне параметризованных типов в Java» и в двух новых главах о коллекциях: в главе 17, в которой приведены наставления по использованию каркаса коллекций, и в главе 18, в которой дается обзор самых важных – и вместе с тем самых спорных – решений, положенных в основу дизайна каркаса. Кроме того, главы о сбережении (глава 5) и об эффективном использовании параметризованных типов (глава 7) подверглись существенной переработке, глава о паттернах проектирования исключена, а глава о миграции кода, написанного до появления параметризованных типов, доступна только в сети в виде приложения ([https://mauricenaftalin.github.io/JGC\\_2e\\_Book\\_Code/appendix.html](https://mauricenaftalin.github.io/JGC_2e_Book_Code/appendix.html)) как представляющая лишь технический и исторический интерес.

В конце этого раздела я поместил, хотя и немного в измененной форме, предисловие к первому изданию. Мне казалось важным передать то чувство волнения, которое охватывало меня при виде тщательно продуманного набора функциональности, с которым версия Java 5 шагнула в XXI век.

— Морис Нафтален,  
Эдинбург, февраль 2025

## БЛАГОДАРНОСТИ ЗА ВТОРОЕ ИЗДАНИЕ

Перво-наперво я хочу поблагодарить всех тех, кто, прочитав первое издание, все же побуждал меня приняться за его пересмотр. На протяжении всего проекта я ощущал поддержку со стороны сообщества Java; я считаю, что мне повезло быть частью такого благосклонного и отзывчивого сообщества.

Сотрудник издательства O'Reilly Зэн Маккуэйд оказал мне неоценимую помощь в подготовке обоснования нового издания. Мой редактор, Сара Хантер, всегда была готова прийти на помощь, когда она была особенно необходима. Мой корректор, Рэйчел Хэд, обладает потрясающим шестым чувством, позволяющим ей находить технические проблемы. Мой выпускающий редактор, Алия Рахман, доблестно исправляла многочисленные мелкие, но существенные погрешности в тексте.

Преимуществом работы над следующим изданием является возможность исправить ошибки, оставшиеся в предыдущем (и попытаться не насажать при этом новых). Из читателей, указавших мне на опечатки в первом издании, хочу отметить Валентина Бака, Томаса Костика, Свейна Эгила Нильсена, Эмануэля Фруа, Андерса Грэнлунда, Равиндра Ранвала, Рэндольфа Ротфусса, Яна де Рюйтера и Кристофера Занвальдта.

Я благодарен многим пожертвовавшим своим временем, чтобы оставить отзыв о книге, предложить исправления, поделиться полезными идеями, крупными и не очень. В некоторых случаях они очень щедро делились со мной временем. Это Мохамед Аниис, Ларри Кейбл, Рэй Джаджадината, Брайан Гётц, Мэри Гоусети, Анджей Гржечик, Джородж Хейнеман, Кэй Хорстманн, Тимми Джоуз, Хайнц Кабутц, Марк Лой, Крис Ньюлэнд, Скотт Оукс, Саймон Парк, Жозе Помар, Саймон Риттер, Роберт Шольте, Дэниэл Шайя, Иван Шипка, Фред Смит, Тушар Сривастава и Йенс-Хаген Сирбе. Разумеется, все оставшиеся ошибки и упущения – целиком моя вина.

Счастлив повторить, что спустя 19 лет включенная в первое издание рекомендация бюллетеня Хайнца Кабутца «The Java Specialists' Newsletter» все еще остается в силе – и это после 325 выпусков!

С удовольствием признаю, что многие идеи, изложенные в этом издании, рождались в сотрудничестве со Стюартом Марксом. Они оттачивались в многочисленных долгих обсуждениях и совместных презентациях. Опыт, знания и поддержка делают его лучшим техническим редактором, о котором любой автор может только мечтать.

Фил Уодлер был и остается моим добрым приятелем. Как обычно, свидетельствую свою любовь Бену, Дэниэлю, Айзеку, Джо и Рут.

## Для кого написана эта книга

Эта книга предназначена всем, кто что-то знает о Java и хотел бы узнать больше. Это не учебное пособие по Java, поэтому мы предполагаем, что читатель знаком с такими базовыми понятиями Java, как классы, статические методы, методы экземпляра и т. д. Мы используем синтаксис Java 21, но большинство синтаксических конструкций, относящихся конкретно к параметризованным

классам, объясняется в тексте. Если вы будете испытывать затруднения с пониманием синтаксиса какого-то кода, то, возможно, ваша интегрированная среда разработки сможет предложить вариант рефакторинга, который прояснит его назначение.

Мы предполагаем наличие знаний самого начального уровня об API коллекций. Если вам доводилось писать программы с использованием `ArrayList` или `HashSet`, то никаких трудностей не возникнет. Обсуждая поведение тех или иных классов коллекций и их методов, мы часто ссылаемся на документацию — она всегда дает исчерпывающую спецификацию.

## ГДЕ ВЗЯТЬ ПРИМЕРЫ КОДА

Примеры кода для этого издания имеются в проекте Maven по адресу [https://github.com/MauriceNaftalin/JGC\\_2e\\_Book\\_Code](https://github.com/MauriceNaftalin/JGC_2e_Book_Code). Отметим, что некоторые примеры не компилируются по причинам, указанным в относящемся к ним тексте.

Вопросы технического характера, а также замечания по примерам кода следует отправлять по адресу [support@oreilly.com](mailto:support@oreilly.com).

Вы можете использовать приведенный в книге код в собственных программах и в документации. Спрашивать у O'Reilly Media разрешения необязательно, если только вы не собираетесь воспроизводить значительную часть кода. Например, никто не возбраняет включить в свою программу несколько фрагментов кода из книги. Однако для продажи или распространения примеров из книг издательства O'Reilly разрешение требуется. Цитировать книгу и примеры в ответах на вопросы можно без ограничений. Но для включения значительных объемов кода в документацию по собственному продукту нужно получить разрешение.

Мы высоко ценим, хотя и не требуем, ссылки на наши издания. В ссылке обычно указываются название книги, имя автора, издательство и ISBN, например: «Java Generics and Collections, 2nd Edition, by Maurice Naftalin and Philip Wadler (O'Reilly). Copyright 2025 Morningside Light, 978-1-098-13672-7».

Если вы полагаете, что планируемое использование кода выходит за рамки изложенной выше лицензии, пожалуйста, обратитесь к нам по адресу [permissions@oreilly.com](mailto:permissions@oreilly.com).

## ГРАФИЧЕСКИЕ ВЫДЕЛЕНИЯ

В книге применяются следующие графические выделения:

- код набран моноширинным шрифтом, а для дополнительного выделения используется полужирный шрифт:

```
class InnocentClient {
    public static void main(String[] args) {
        List<Integer>[] intLists = DeceptiveLibrary.createIntLists(1);
        List<? extends Number>[] numLists = intLists;
        numLists[0] = List.of(1.01);
        int i = intLists[0].get(0); // исключение приведения к типу класса!
    }
}
```

- мы часто включаем код, принадлежащий телу метода `main`:

```
List<Integer>[] intLists = Deceptivelibrary.createIntLists(1);
List<? extends Number>[] numLists = intLists;
numLists[0] = List.of(1.01);
int i = intLists[0].get(0); // исключение приведения к типу класса!
```

В репозитории кода эти фрагменты собраны в методы `main`. Зачастую несколько фрагментов собираются в один класс. Отметим, что не все классы в репозитории компилируются: объяснения ошибок компиляции приводятся в тексте;

- включенные в основной текст фрагменты кода, в в том числе такие элементы программы, как имена переменных и функций, баз данных, типов данных и переменных окружения, предложения и ключевые слова языка, печатаются *моноширинным* шрифтом (как в примере имени метода `main` в предыдущем абзаце);
- директивы импорта в тексте опущены. Но они имеются в репозитории;
- примеры интерактивных сеансов, содержащие командную строку и соответствующий вывод, набраны *моноширинным* шрифтом, а данным, вводимым пользователем, предшествует знак процента:

```
% javac g/Stack.java g/ArrayStack.java g/Stacks.java l/Client.java
Note: Client.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

Иногда мы приводим фрагмент сеанса JShell, в котором за приглашением и введенными пользователем данными следует вывод JShell:

```
jshell> List.of("Larry", "Curly", "Larry", "Moe")
$2 ==> [Larry, Curly, Larry, Moe]
```

- когда пользовательские данные не умецаются в одной строке, первая строка заканчивается обратной косой чертой:

```
% javac -Xlint:unchecked g/Stack.java g/ArrayStack.java \
%   g/Stacks.java l/Client.java
l/Client.java:4: warning: [unchecked] unchecked call
to push(E) as a member of the raw type Stack
    for (int i = 0; i<4; i++) stack.push(new Integer(i));
```

- новые термины, URL-адреса, адреса электронной почты, имена и расширения файлов набраны *курсивом*;
- текст, набранный *моноширинным курсивом*, следует заменить предоставленными пользователем данными или значениями, вытекающими из контекста.



Так обозначается замечание общего характера.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)