

Generics: ? wildcard

Java Collections
5 уровень, 9 лекция

ОТКРЫТА

— Ну, и напоследок еще одна маленькая лекция по Generic.

Сейчас я тебе расскажу, как обходить «стирание типов» (Type erasure).

— Ага. Мне тоже хочется это знать.

— Как ты уже наверное знаешь, в Java есть тип Class, который используется, чтобы хранить ссылку на объект класса. Примеры:

Примеры	
1	Class clazz = Integer.class;
1	Class clazz = String.class;
1	Class clazz = "abc".getClass();

— Ага.

Но вот чего ты, наверное, не знаешь, так это того, что есть еще один класс Class, который является Generic’ом. И переменные этого Generic Class’a могут хранить только ссылки на тип, который был типом-параметром. Примеры:

Примеры	
1	Class<Integer> clazz1 = Integer.class; //все отлично работает
2	Class<String> clazz2 = Integer.class; //ошибка компиляции.
1	Class<String> clazz1 = String.class; //все отлично работает
2	Class<String> clazz2 = int.class; //ошибка компиляции.
1	Class<? extends String> clazz1 = "abc".getClass(); //все отлично работает
2	Class<Object> clazz2 = "abc".getClass(); //ошибка компиляции.

— А почему оно так работает?

— Дело в том, что значение поля class у типа Integer (т.е. у Integer.class) – это на самом деле объект типа Class<Integer>.

Но давай пойдем дальше.

Так вот, пользуясь тем фактом, что Class<T> — это Generic, и тем, что переменная его типа может хранить значение только типа T, можно сделать вот такую хитрую комбинацию:

Пример	
1	class Zoo<T>

```

4     ArrayList<T> animals = new ArrayList<T>();
5
6     Zoo(Class<T> clazz)
7     {
8         this.clazz = clazz;
9     }
10
11    public T createNewAnimal()
12    {
13        T animal = clazz.newInstance();
14        animals.add(animal);
15        return animal
16    }
17 }

```

Как это использовать

```
Zoo<Tiger> zoo = new Zoo<Tiger>(Tiger.class); // вот тут передается тип!
Tiger tiger = zoo.createNewAnimal();
```

Это не мегахитрая комбинация – мы просто передаем ссылку на нужный тип. Но если бы мы просто пользовались `Class` вместо `Class<T>`, то кто-то мог по ошибке передать туда два разных типа – один в качестве параметра `T`, другой – в конструктор.

— Ага. Понимаю. Ничего сверхъестественного не произошло и страшного тоже. Ссылка на тип есть, пользоваться ей можно, работает и ладно.

— Вот, слышу слова «не мальчика, но мужа!» Работает и ладно – это часто самый оптимальный вариант.

Очень много всего можно было бы теперь переделать в Java, но нужно сохранять совместимость со старым кодом.

Именно десятки тысяч популярных отлаженных библиотек – это сегодня самый весомый аргумент в пользу Java. Так что, сохраняя обратную совместимость, Java остается самым популярным языком, и поэтому не может внедрять радикальные новшества.

— А я сделаю свою Java с блекджеком и ...

— Ладно, я уже подустал за день. Давай до свидания.

— До свидания, Риша, и спасибо за такой интересный урок.

[← Предыдущая лекция](#)

Следующая лекция >

 +73

Комментарии (53) популярные новые старые

популярные новые старые

новые старые

старые

JavaCoder

Введите текст комментария

11/11/2019 10:00 AM

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

[НАЧАТЬ ОБУЧЕНИЕ](#)

Ответить

+8

Max Zap

Уровень 41

1 ноября 2021, 12:06

Этот метод deprecated с 9 версии. Вместо него используется нынче `clazz.getDeclaredConstructor().newInstance()`.
docs.oracle.com

Ответить

+7

Сергей Жирков

Backend Developer в MvideoMobile

25 сентября 2021, 04:38

Это вообще хоть где-то пригодится?))

Ответить

+1

Лиза Воренувкина

Уровень 43, Кривой Рог, Ukraine

25 сентября 2021, 11:20

Дают значит надо .Я просто читаю интересно хоть и сложно без практики .

Ответить

+1

Михаил

Full Stack Developer в финтех

17 февраля, 22:07

Да, пригодится. Например, когда нужно будет работать с различными наследниками одного предка, но при этом избежать преобразования типов, т.к. это ограничивает полиморфизм.

Ответить

0

Сергей

Java Developer в Адвантум

27 июля 2021, 20:23

Пц! Это вообще что?! Такое ощущение, что блок Collections делали по принципу "и так сойдет".

— Ага. Понимаю. Ничего сверхъестественного не произошло и страшного тоже. Ссылка на тип есть, пользоваться ей можно, работает и ладно.

— Вот, слышу слова «не мальчика, но мужа!» Работает и ладно – это часто самый оптимальный вариант.

Ответить

+5

LuneFox

инженер по сопровождению в BIFIT

EXPERT

1 февраля, 19:42

Ответить

+2

Igor Petrashevsky

Уровень 44

25 августа, 03:12

это было понятно с появлением глубокопродуманной переменной с именем clazz

Ответить

0

Valua Sinicyn

Уровень 41, Харьков, Украина

15 февраля 2021, 10:44

Если бы еще "мегахитрая комбинация" компилировалась, было бы вообще супер. А так, очередной сферический конь.

Ответить

+1

Vova Enin

Уровень 36

10 сентября 2021, 10:01

Все отлично компилируется

Ответить

0

Хорс

Уровень 41, Харьков

28 августа 2020, 16:43

Мне оч помогло это разъяснение:
<https://stackoverflow.com/questions/6231973/difference-between-list-list-listt-liste-and-listobject>

Ответить

0

Сергей

Уровень 41, Екатеринбург

11 апреля 2020, 09:16

Кажется, перепутали заголовки 7 и 9 лекции

Ответить

+17

Роман Тарнакин

Уровень 35, Москва, Россия

25 мая 2020, 13:48

100%

Ответить

0

Artem

Уровень 40, Москва

28 октября 2020, 12:56

Лупа И Пупа

Ответить

+1

Pig Man

Главная свинья в Свинарнике

10 февраля 2021. 19:51

Flexo

Bending Unit #3370318

16 апреля 2021, 16:42

...

судя по блокам идущих вслед за лекциями задач, перепутали не названия, а содержания лекций

Ответить

Anna Avilova

architect

25 апреля, 17:22

...

слоган всего блока лекций по коллекшн)))

Ответить

Den_lx

Уровень 41, Гродно, Беларусь

7 марта 2020, 20:47

...

после игр в казино легко понять смысл wildcard (там wild заменяет все символы)

Ответить

Boris

Уровень 41, Молдова

EXPERT

5 марта 2020, 00:36

...

нормально сначала задание потом обесняют:))

Ответить

Vorlock

Уровень 31, Днепр, Украина

27 января 2020, 22:38

...

спасибо в шляпу

Ответить

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



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

▼

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

Доступно в Google Play

Загрузите в App Store

НАЧАТЬ ОБУЧЕНИЕ

