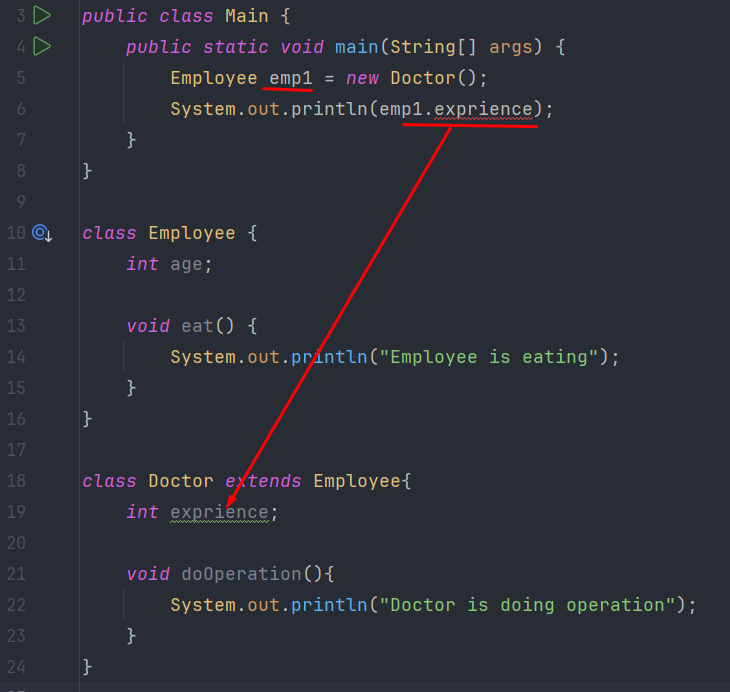
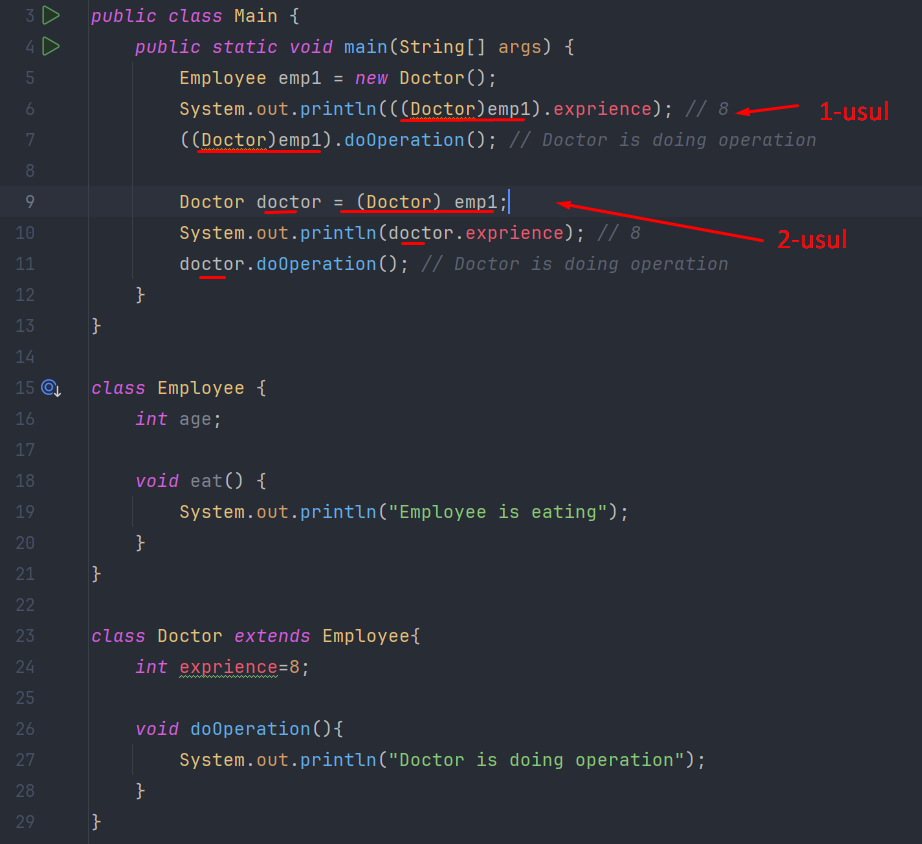


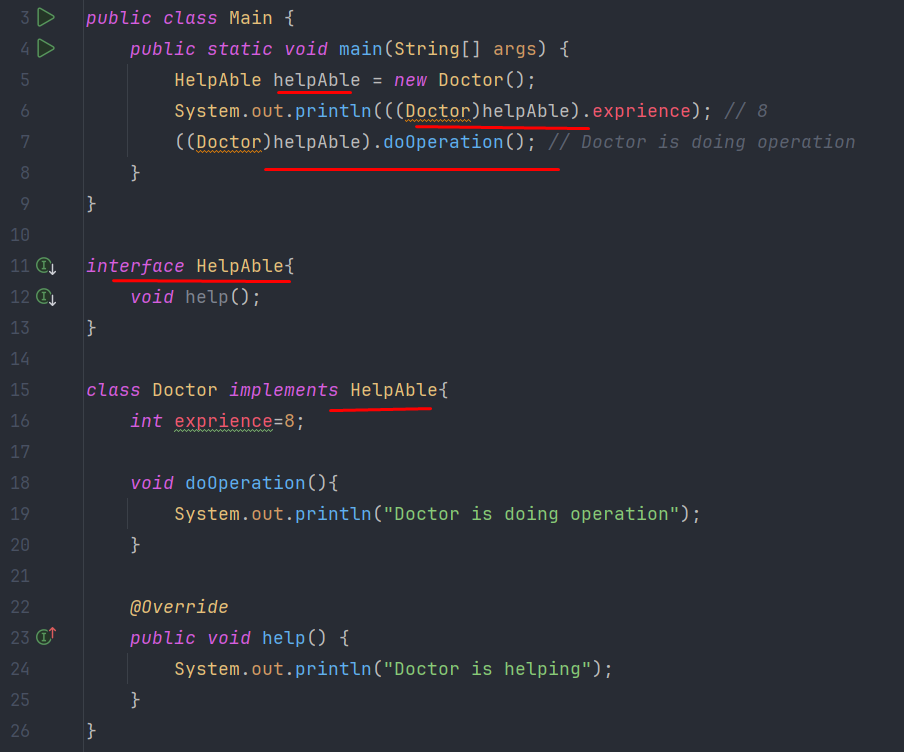
Pastdagi misolda reference type casting ni ko’ramiz. E’tibor bersak, emp1 o’zgaruvchisi faqat Employee classdagi o’zgaruvchilarni, methodlarni va agar bolasida overridden bo’lgan method bo’lsa, shu bolasini overrriden bo’lgan methodini chiqaradi. emp1.exprience o’zgaruvchisi esa faqat bola classda mavjud. Demak bola classni o’zgaruvchisini emp1 yordamida chaqira olmaymiz. Xatolik beradi. Sababi hamma employee doctor emas:



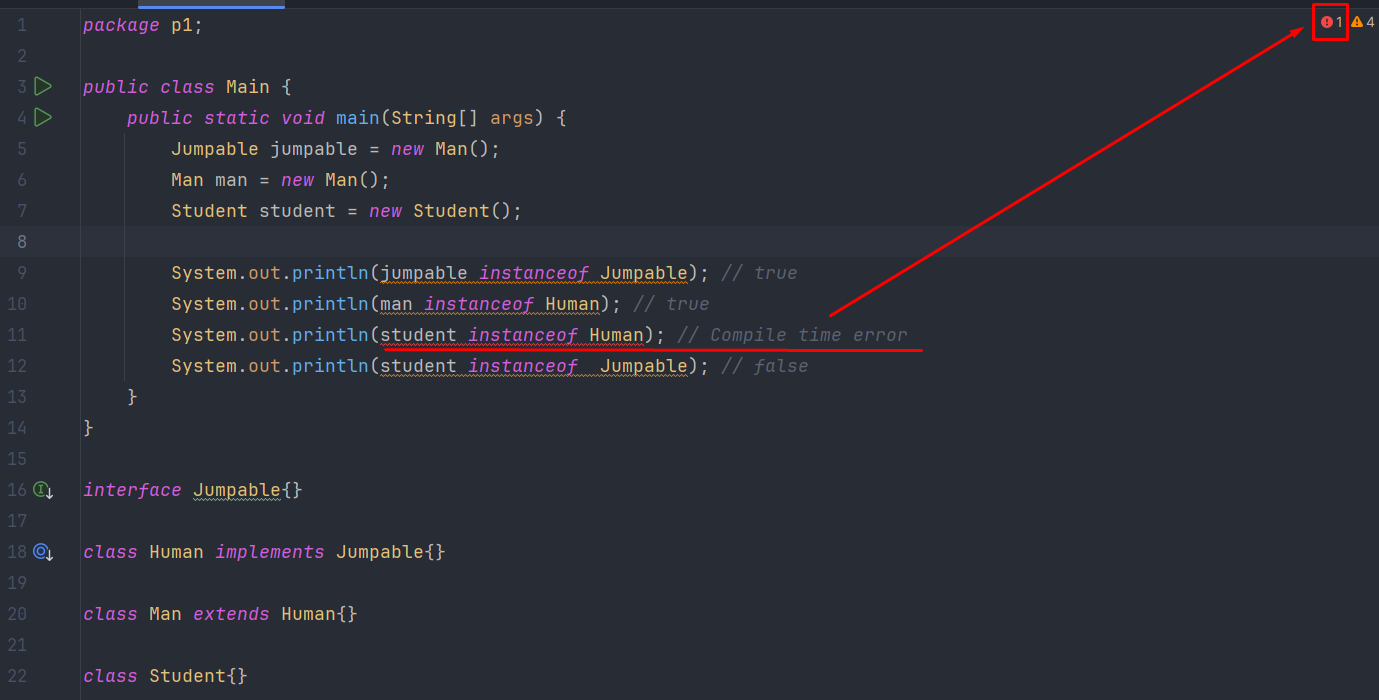
Xo’sh bunday paytda bizga casting degan tushuncha yordam beradi. Pastda ko’rish mumkinki, casting ni 2 xil usul bilan qilish mumkin ekan. Pastda berilgan 2 la usul ham:



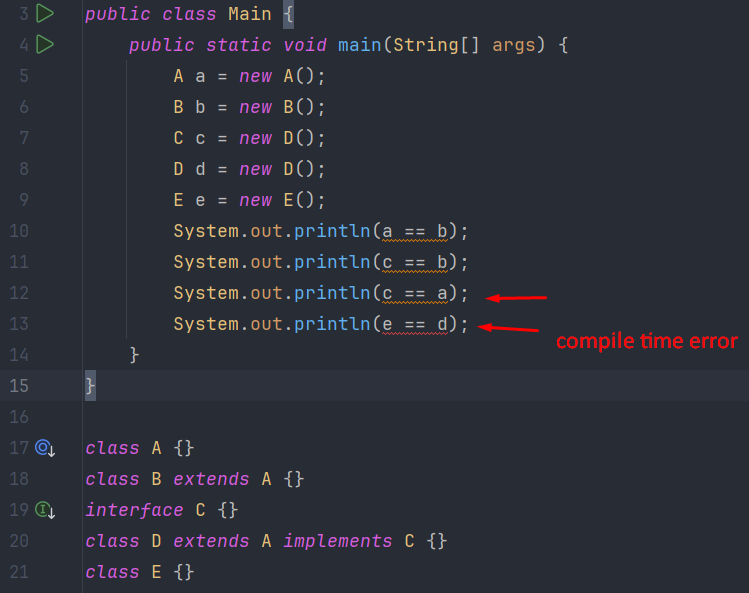
Casting interface lar bilan ham bo’lishi mumkin:



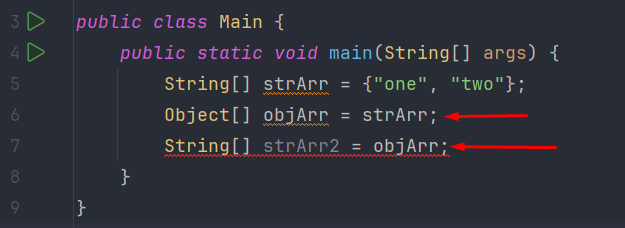
Shuni yozishim kerak



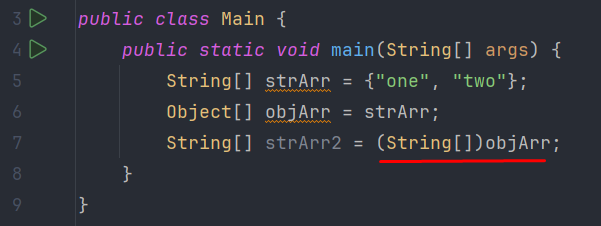
Shuni yozishim kerak



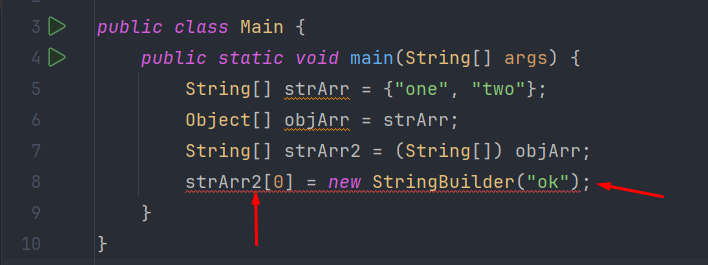
**Object** type li arrayga **istalgan** typeli arrayni tenglashtirish mumkin. Chunki istalgan type li **array** bu **object** type li arraydir. Masalan pastda **6**-qatorda **string** type li **strArr** ni **objArr** ga tenglashtirishimiz mumkin. Hech qanday xatolik bo’lmaydi. Lekin teskarisi esa mumkin emas. **7**-qatorda type string bo’lgan **strArr2** arrayga type **Object** bo’lgan **objArr** arrayni o’zlashtira olmaymiz. Chunki istalgan type li array string array emasdir. Shuning uchun **7**-qatorda **compile time** xatolik olyapmiz:



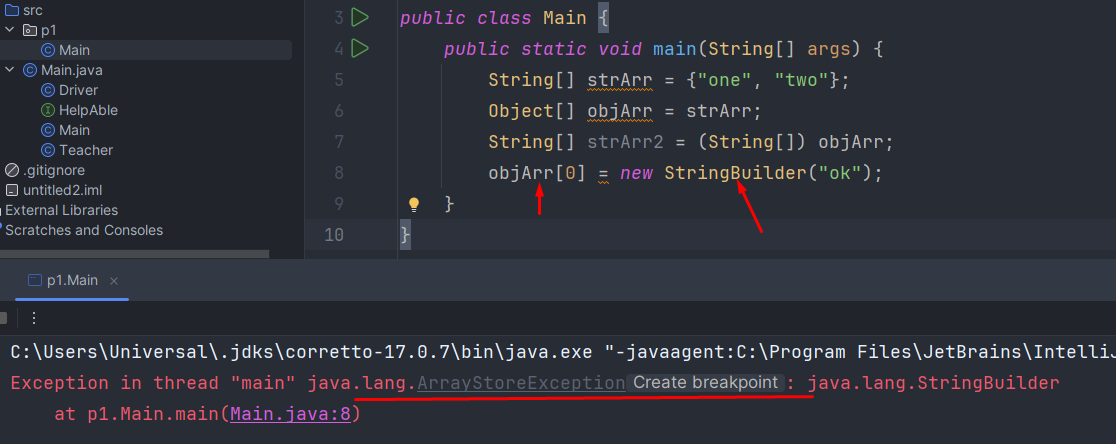
Xo’sh bu xatolikdan qanday qutulamiz. Buning uchun bizga casting yordam beradi. Casting qilsak, xatolik yo’qoladi:



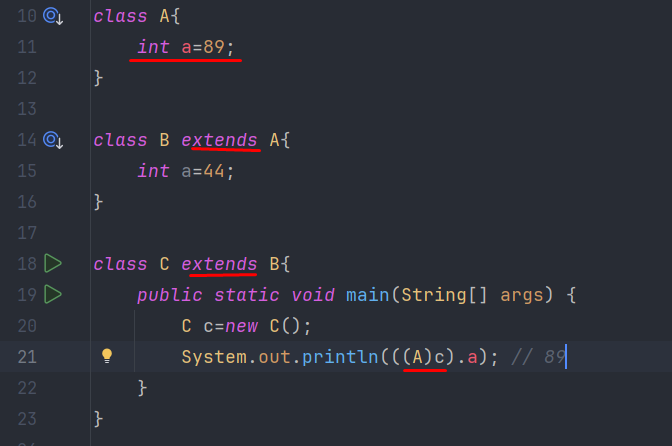
Pastda e’tibor bersangiz **strArr2** ni type string bo’lgani uchun, **StringBuilder()** qabul qila olmaydi. Shuning uchun compile time error beradi:



Lekin pastdagi misolga e’tibor bersangiz, u holda bu misolda **objArr** arrayni type **Object** bo’lgani uchun bemalol istalgan object qabul qila oladi, shu qatori **StringBuilder** ni ham. Bunda compile time error bermaydi. Lekin run qilsak, u holda runtime da exception beradi. Sababi **objArr** ni type **Object** bo’lsa ham, **objArr** **strArr** ga pointer qilyapti. **objArr[0]** esa **“one”** nomli **stringga** pointer qilyapti. **StringBuilder** bunday holatda berish runtime xatolikka sabab bo’ladi. **StringBuilder** va **String** boshqa-boshqa objectlardir:



Agar bizga ota classning otasini o’zgaruvchisi kerak bo’lsa, u holda biz **super** ni ishlata olmaymiz. Buning uchun **casting** kerak bo’ladi. Pastda bizda **3** ta class bor. **A => B => C** kabi munosabatda bo’lib, **A** class Bobo , **B** class Ota, **C** class bola classdir. **C** classdan turib, **A** classni o’zgaruvchisini olmoqchi bo’lsak, u holda **21**-qatordagi kabi casting qilish kerak:



Yuqoridagi misolda har biriga **foo()** nomli method qo’shamiz. Endi ularni istalgan object bilan casting qilib chaqirishimizdan qat’iy nazar, baribir bola **C** classni **overridden** bo’lgan **foo()** methodini chiqaradi:

