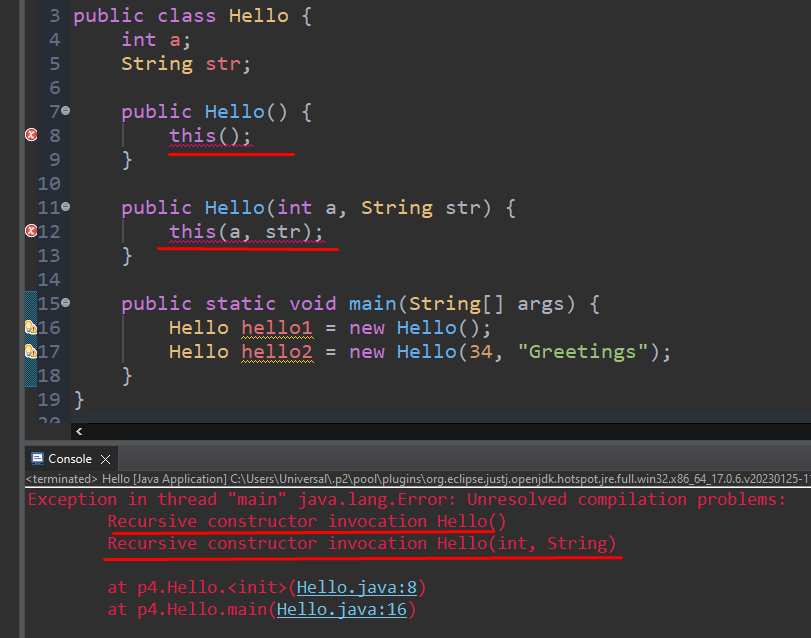
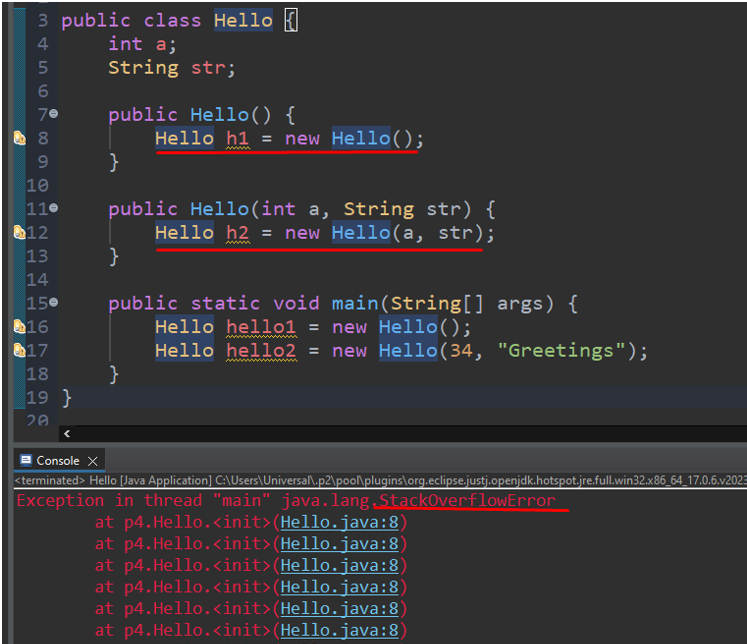
Javada ba’zida **constructor(c.)** lar bilan ishlashda c.ni o’zini ichida infinitive ko’p marta chaqirib yuborishimiz mumkin. Albatta bu narsa xatodir va Java bizga StackOverflowError exception otadi.

Masalan, pastdagi misolda Hello() nomli c.ni this() bilan chaqiryapmiz. Bu yerda this() bu Hello() c.ga teng bo’lib, xuddi recursive f-ya kabi o’zini o’zi chaqiryapti. Xuddi shu holat this(a, str); da ham bo’lyapti, bu esa o’z navbatida Hello(int a, String str); ni chaqiradi. 2 la holatda ham bizda c.ni recursive chaqirishdan to’xtatadigan base case yozilmagan. Biz this kalit so’zini ishlatish orqali, xatolik compile timeda sodir bo’lyapti:



Pastdagi misolda 8-va 12-qatorlarda objectlar yaratilyapti. C.ni ichida o’zga mos c.ni chaqirib object yaratsak, xatolikka olib keladi. Bu yerda compile time error chiqmaydi, lekin run qilib ko’rgacha, runtime error chiqadi. Errorni nomi **StackOverflowError** bo’lib, 8- va 12-qatorlarda constructorlarimiz o’zini o’zi chaqiryapti. Demak recursive constructor calling bo’lyapti:



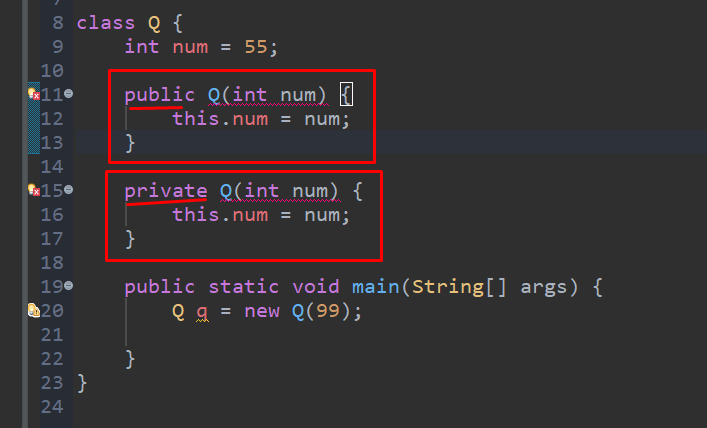
<https://www.geeksforgeeks.org/recursive-constructor-invocation-in-java/>

<https://www.javatpoint.com/recursive-constructor-invocation-in-java>

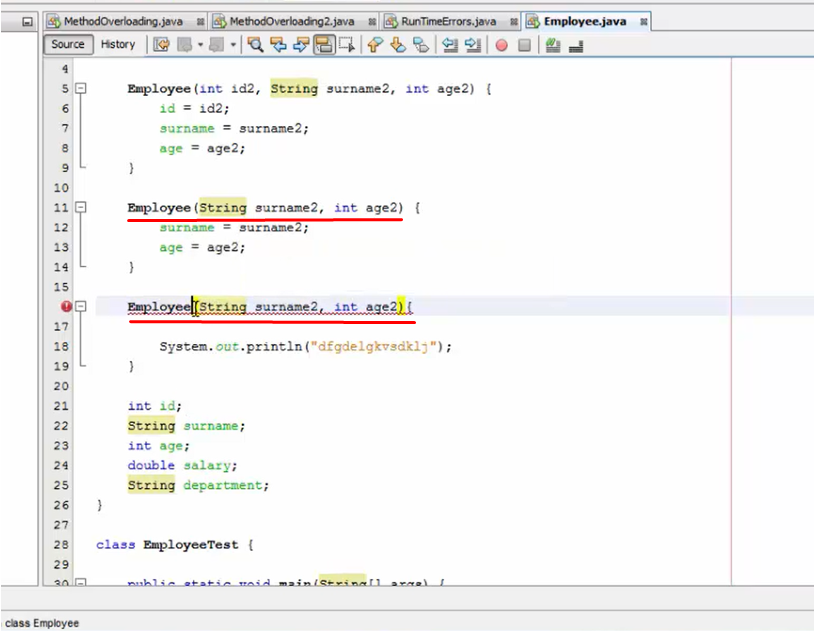
<https://copyprogramming.com/howto/recursive-constructor-invocation>

<https://www.tutorialspoint.com/recursive-constructor-invocation-in-java>

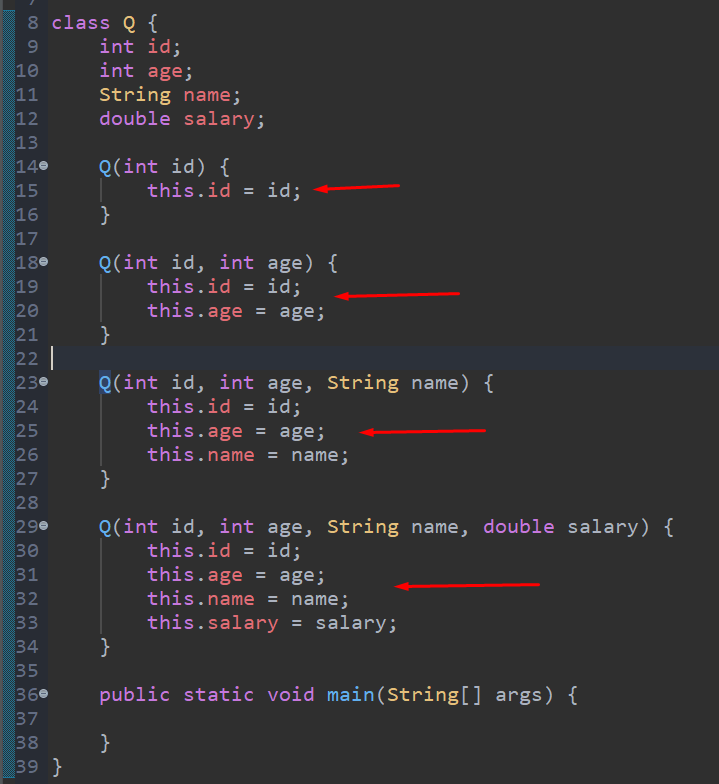
Faqat **access modifierlari** bilan farq qiladigan constructorlar **overlaod** bo’lmaydi. **Compile time error** beradi:



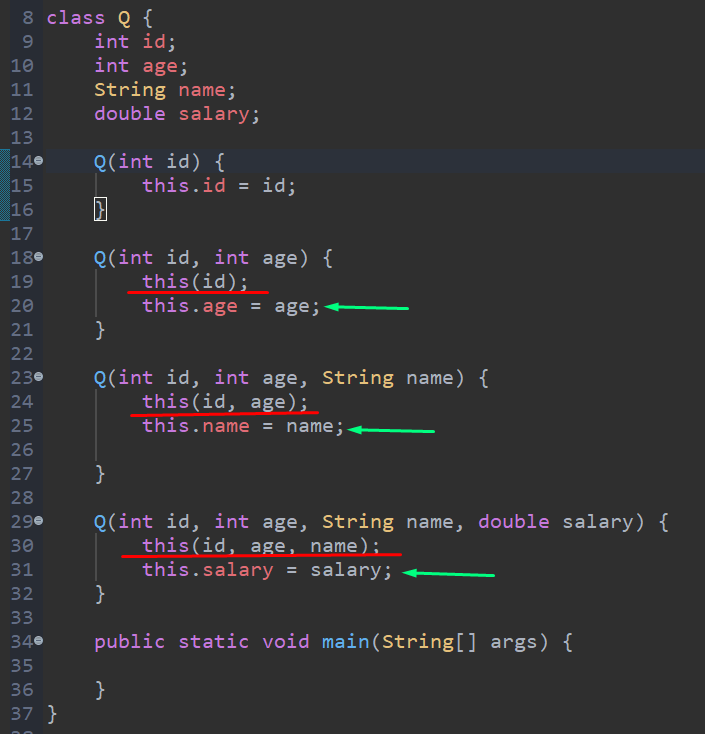
E’tibor bergan bo’lsangiz, faqat **11**- va **16**-qatorlardagi constructorlar overload bo’lmaydi. Sababi ularni parametrlari bir xildir, ya’ni **String** va **int** dir. Bu esa constructorlarni overload qoidasiga mos tushmaydi:



E’tibor bergan bo’lsangiz, 4 ta constructorimizda **id, age, name, salary** kabi fieldlar takrorlanyapti:



Xo’sh bundan qutulishni iloji bormi? Albatta buning uchun **this** dan foydalanish kerak. Bunda har bir constructorga mos **this** chaqiriladi:



E’tibor bergan bo’lsangiz, ba’zi bir fieldlar yana takrorlanyapti. Bu fieldalar yashil strelka bilan ajratib ko’rsatilgan. Bundan ham qutulishni iloji bor. Eng ko’p field o’zida saqlaydigan constructorni o’zimizga belgilab olib, shu bo’yicha boshqa constructorlarda shuni ishlatamiz. Agar bitta parameter oladigan constructorda, 4 ta parameter oladigan constructorni **this** blan chaqirsak, 3 ta parameter qolib ketadi. Shu qolib ketgan 3 ta parameterga biz o’ziga mos typeni default qiymatini berishimiz mumkin. Masalan **int => 0, String => null, double => 0.0** yoki **0** ham tushadi:

