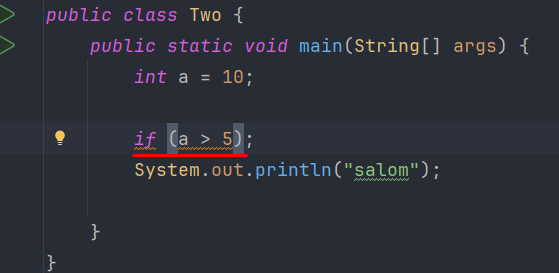
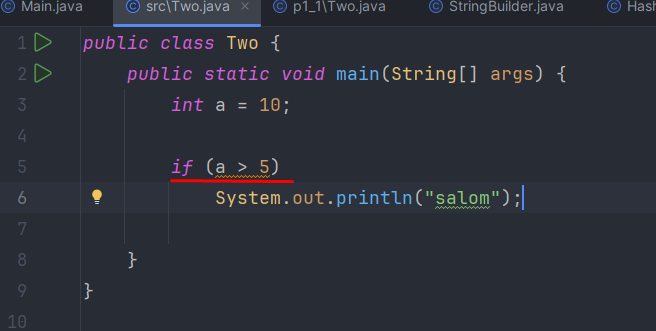
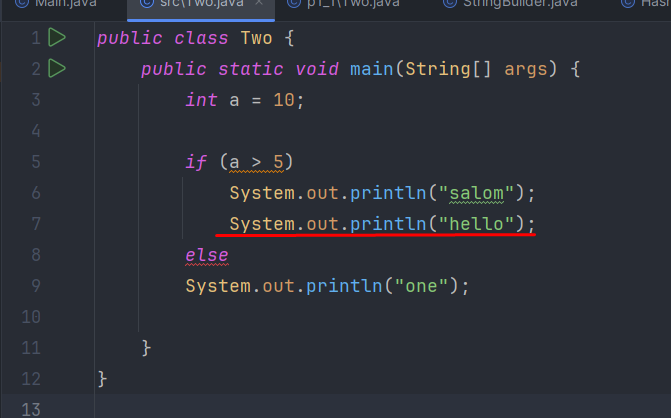
Javada **if** ni pastdagi kabi tanasini yozmasdan bo’sh qoldirish mumkin. Bunda undan keyingi ifoda **if** ni ichidan **true** yoki **false** qaytishidan qat’iy nazar ishlayveradi. Chunki **if** dan keyin **;** belgisini qo’yganmiz. Bu esa **if** ni tugaganligini bildiradi:

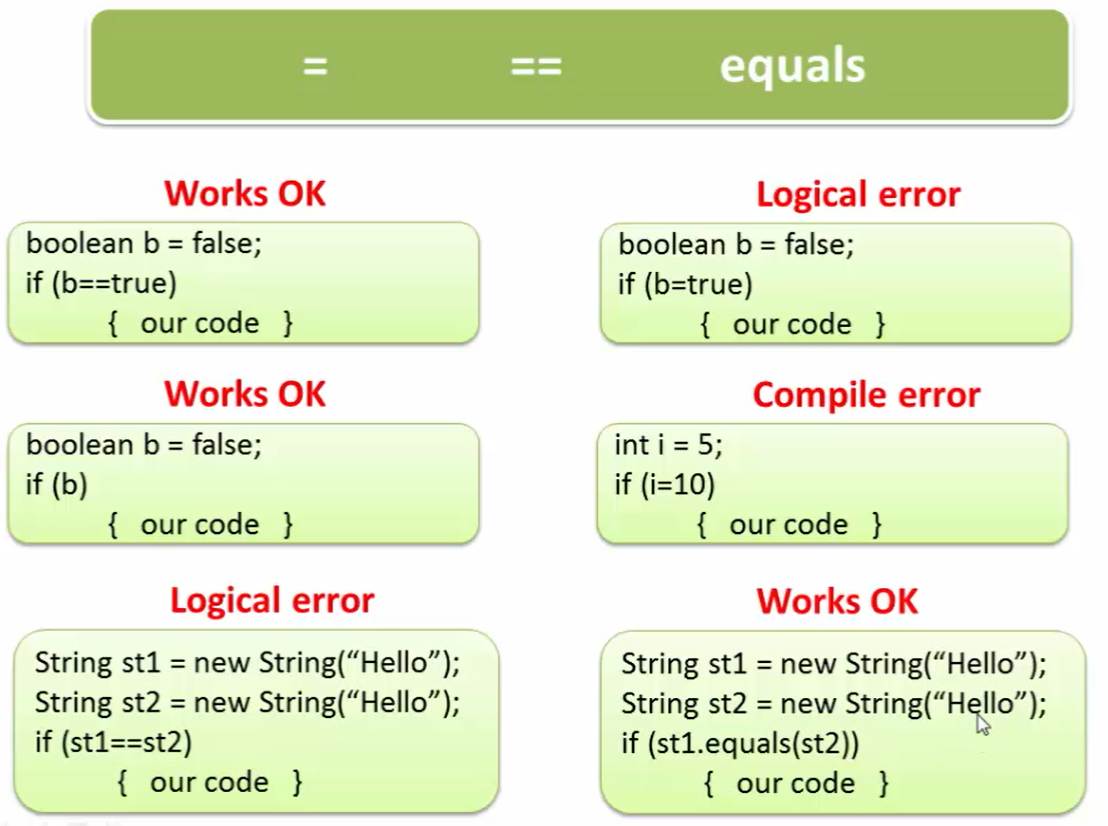


Lekin pastdagi ifodada **sout** agar **if** dan true qiymat qaytsa ishlaydi:



Mana bunday ham yozish ham noto’g’ri. Chunki 7-qatordagi sout kerak emas. If dan keyin agar {} belgisi bo’lmasa, u holda if dan keying birinchi qatorni oladi, qolgan qatorni olmaydi, xatolik hisoblanadi:



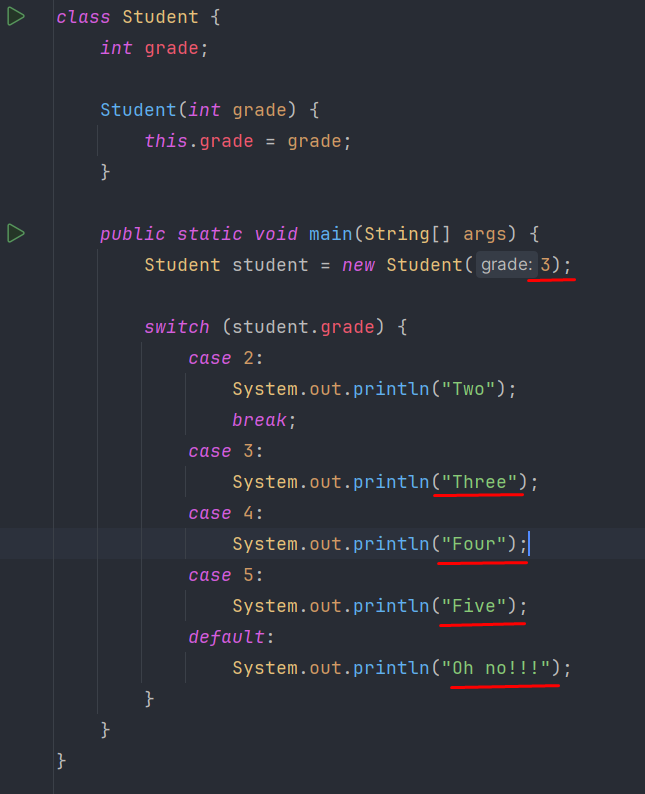


Pastda **switch** da **break** bo’lmasa, u holda shart bajariladigan **case** dan boshlab birinchi **break** uchragancha keyingi **case** larni bajaraveradi:

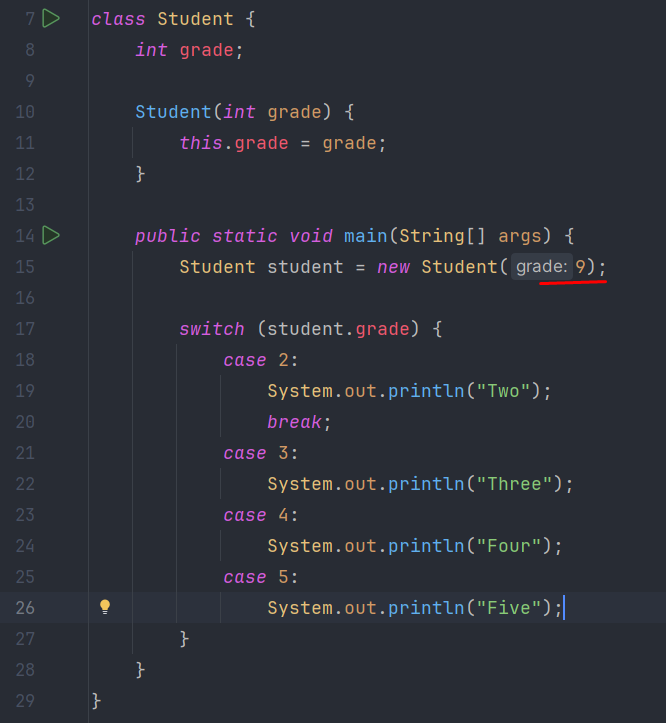


Natija: **“Three” “Four”**

Agar birorta **break** uchramasa shart bajariladigan **case** dan keyin, u holda **default** dagi qism ham ishlaydi:



Agar **switch-case** da **default** qismi bo’lmasa va birorta **case** ga tushmaydigan son bersak, u holda ekranga hech nima chiqmaydi:

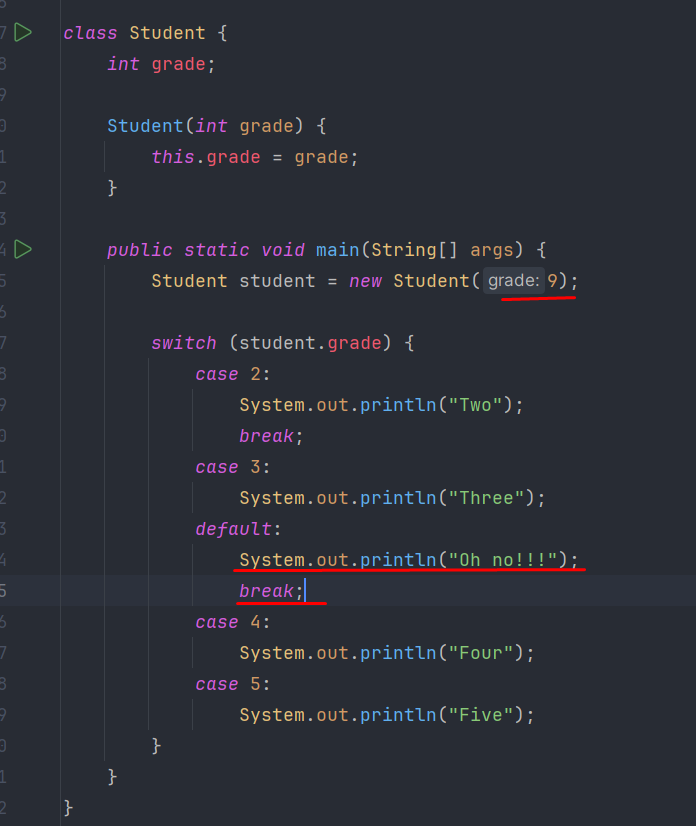


Istasak biz default statementni xohlagan joyga ko’chirishimiz mumkin. Bunda faqat bir narsani yaxshi bilish kerak. Agar hech bir case ga tushadigan qiymat bo’lmasa, tabiiyki default qismi ishlaydi, lekin default qismidan keyin hech qanday break yozilmagan bo’lsa, u holda defaultdan keying case lar ishlaydi to break uchragancha. Pastda bajariladigan qatorlar uzun qizil rang bilan chizilgan:



Natja: **“Oh no!!!” “Four” “Five”**

Agar **default** dan keyin break bo’lsa, u holda bu **break** ishlab, keying qatordagi kodlarni bajarilganini qo’ymaydi:



Natija: **“Oh no!!!”**

Faqat bitta **default** statement yozish kerak, aks holda xatolik beradi:



Xuddi shunday 2 ta bir xil shartli **case** yozish mumkin emas, xatolik beradi:



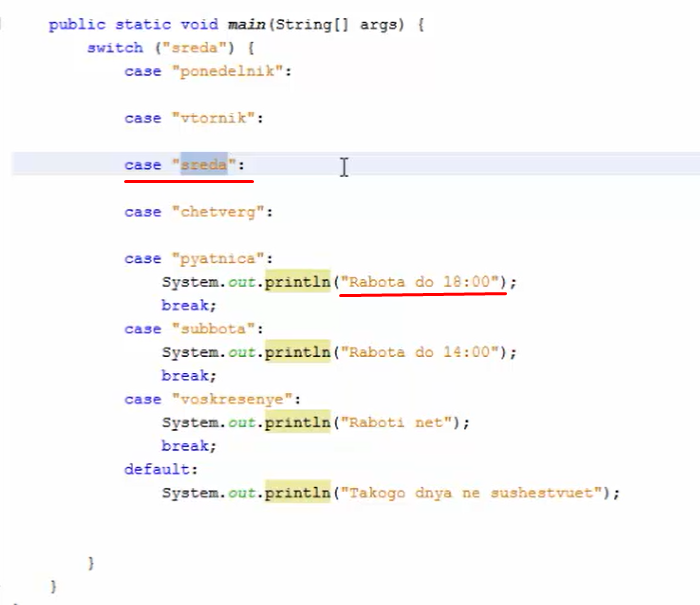
Agar istasak **case** da matematik ifoda yozish mumkin. Masalan **21**-qatorda **3+5** deb yozdik. Xato emas, faqat chiqqan natija boshqa **case** da shu natija bo’lmasligi kerak:



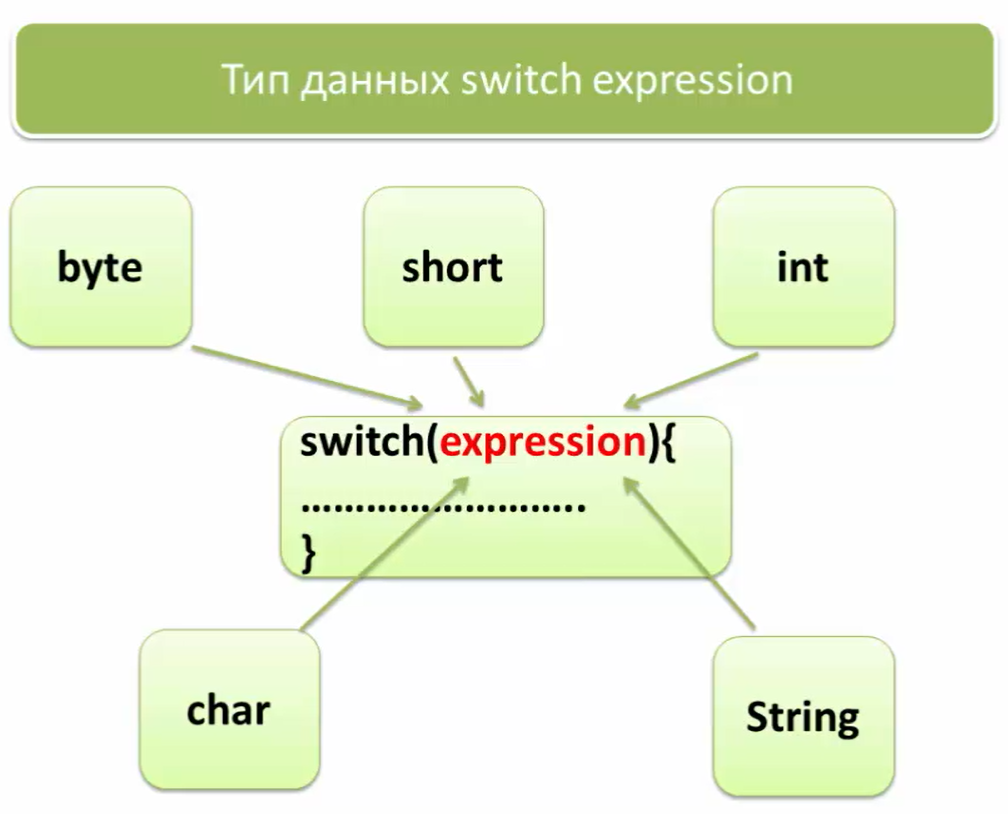
Agar matematik ifodadan chiqqan natija boshqa **case** dagi natija bilan bir xil bo’lsa, u holda xatolik beradi. Chunki **2** ta **case** da bir xil qiymat bo’lib qoladi: Pastda xatolik berilgan holat berilgan:



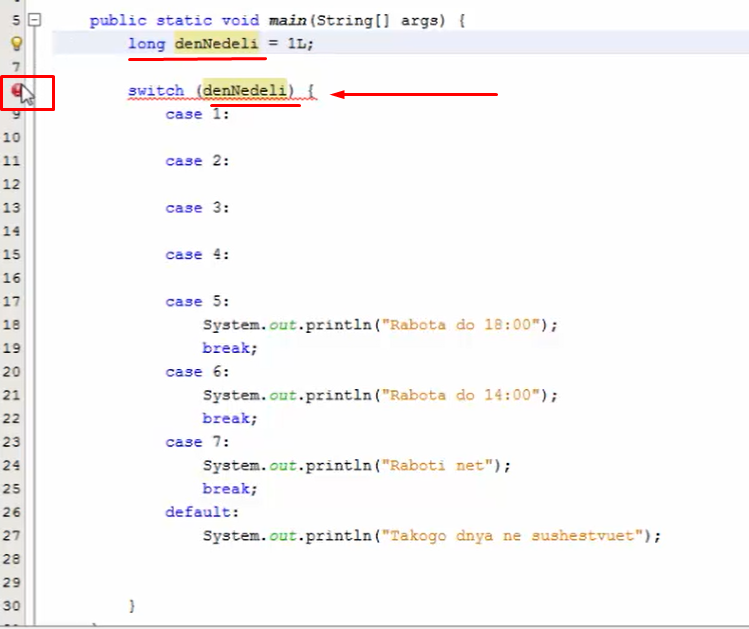
Agar istasak bir nechta **case** uchun bitta **break** va **sout** yozish mumkin:



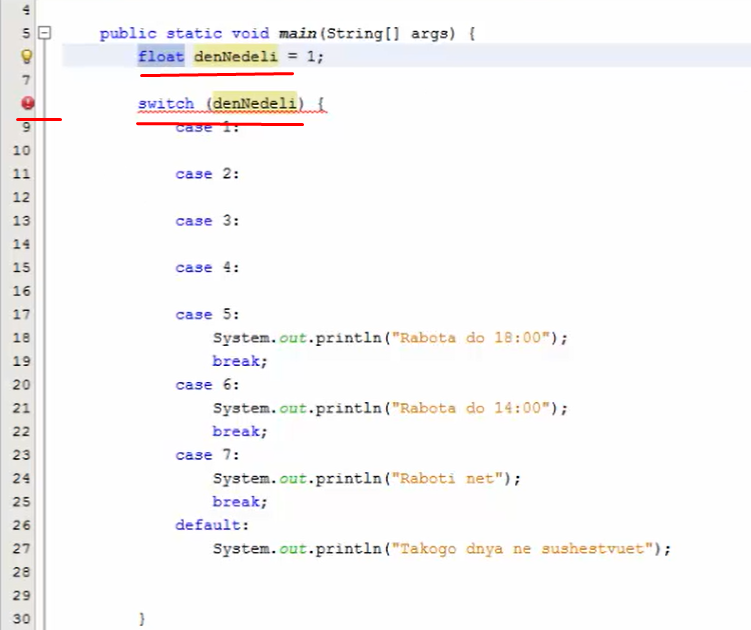
**Switch-case** bilan ishlayotganda **switch** ni ichidagi **expression** ga istalgan typedagi qiymat bera olmas ekanmiz. Faqat **byte**, **short**, **int**,**char**, **String** typedagi qiymatlar bera olamiz. Bu typelardan tashqari boshqa **long**, **boolean**, **double**, **float** va istalgan **reference** type(**String** dan tashqari xolos) larni bra olmaymiz:



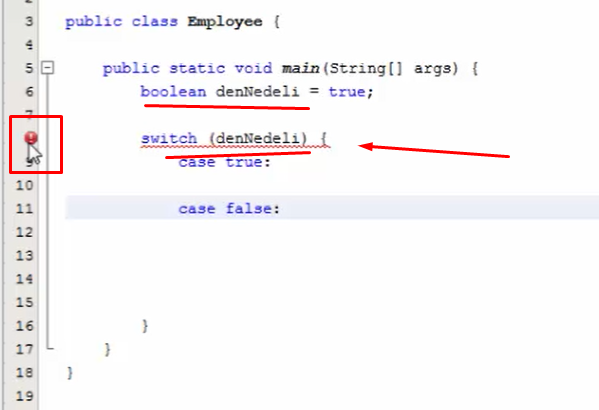
Masalan, **switch** ga **long** type bera olmas ekanmiz, xatolik beradi:



Masalan, **switch** ga **float** type bera olmas ekanmiz, xatolik beradi:



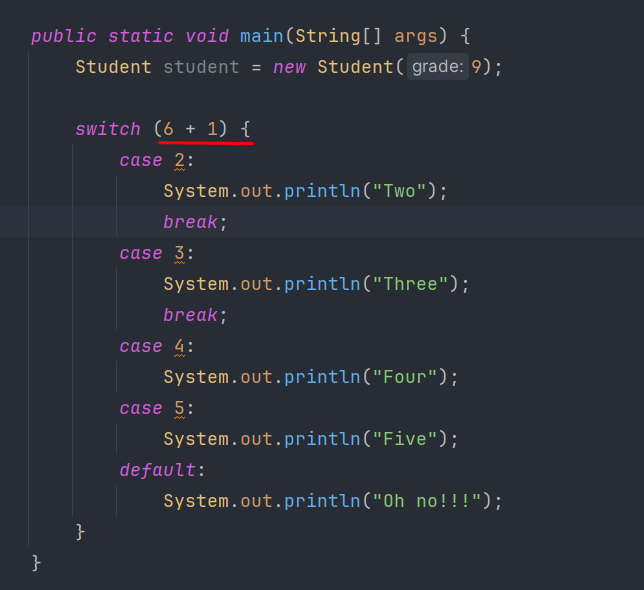
Masalan, **switch** ga **boolean** type bera olmas ekanmiz, xatolik beradi:



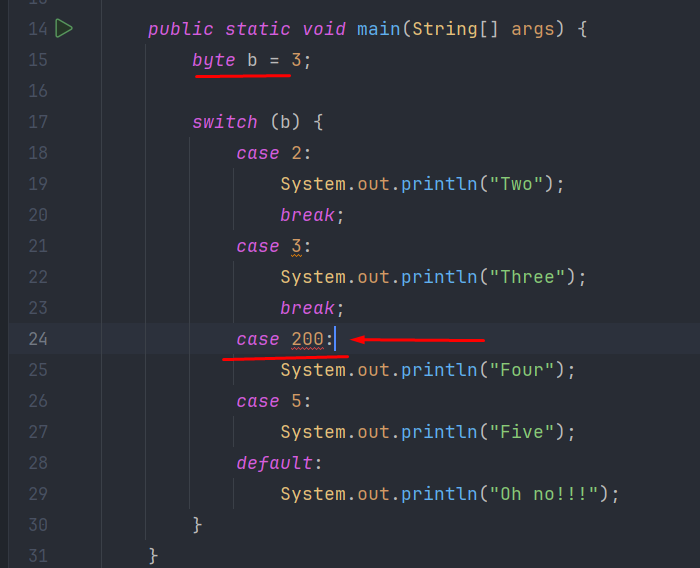
Istasak **switch** dagi expression ga matematik ifoda yozish mumkin:



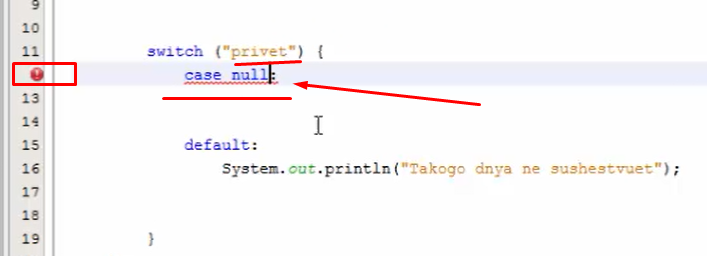
Yoki bunday ham yozish mumkin:



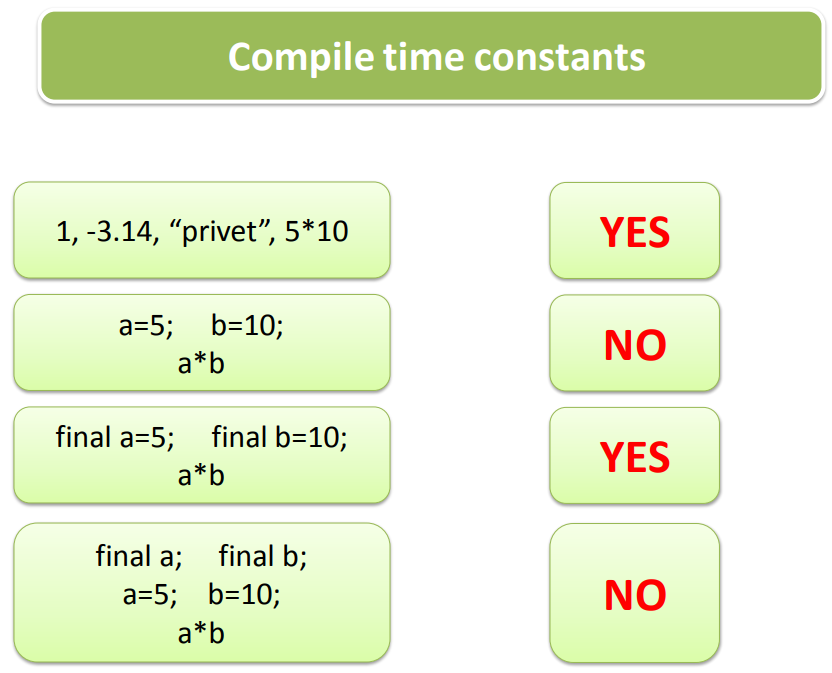
Pastda misolda b ni type byte bo’lgani uchun uni chegarasi [-128, 127] dadir. Shuning uchun 24-qatorda xatolik beradi:



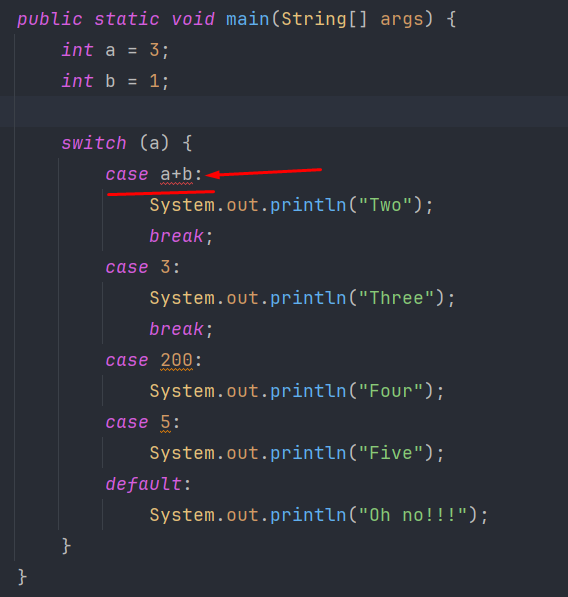
Agar **switch** ifoda **string** oladigan bo’lsa, case da **NULL** nib era olmaymiz. To’g’ri **String** uchun default qiymat null, lekin compilator buni xato sifatida ko’radi. Haqiqiy string qiymat kelishi kerak. Pastdagi ifoda xatodir:



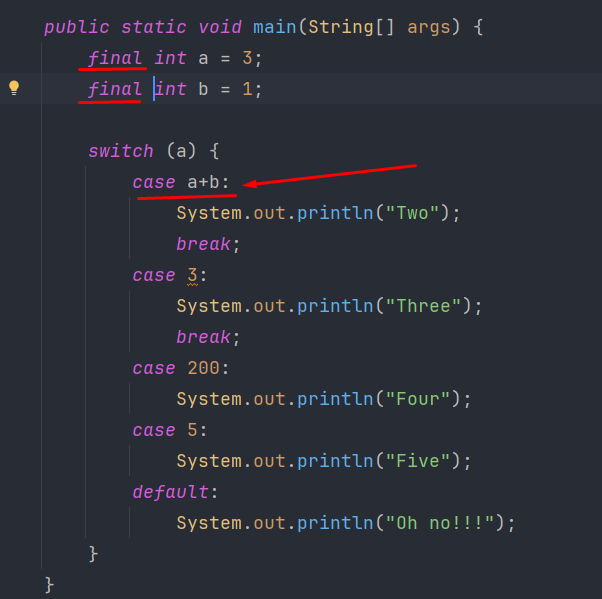
Switch qiziq bir holat bor. Case qismida doim compile time constant lar bo’lishi kerak, ya’ni 34, 90, “hello”, 4\*6 kabi aniq qiymatlar olishi kerak. Aks holda xatolik beradi. Runtime da aniq bo’lmasligi kerak ularni qiymati.



Pastdagi ifoda xato hisoblanadi:



Lekin yuqoridagi ifodani final bilan yozsak xato bo’lmaydi. Chunki final ni qiymati o’zgarmasdir. Pastdagi ifoda to’g’ri ishlaydi:



Lekin bunday yozsak xato hisoblanadi:

