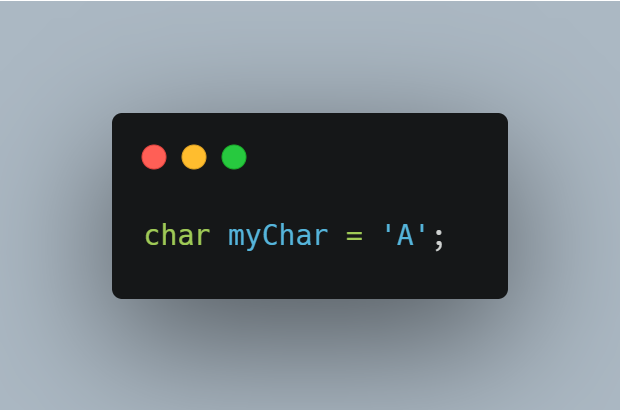
**Biežāk lietotās funkcijas darbam ar simboliem un simbolu virknēm programmēšanas valodā Java**

Simboli un simbolu virknes ir būtiskas daļas programmēšanā. Programmēšanas valoda Java nodrošina daudzas noderīgas funkcijas darbam ar simboliem un simbolu virknēm. Šajā mācību materiālā apskatīsim dažas no tām.

**1. Simbolu datu tips - char**

Java nodrošina simbolu datu tipu **char**, kas ļauj glabāt atsevišķu simbolu, piemēram, burtu vai ciparu. **char** vērtība tiek deklarēta ar vienādības zīmi un atbilstošu simbolu iekavās.

Piemēram:

**2. Simbolu virknes - String**

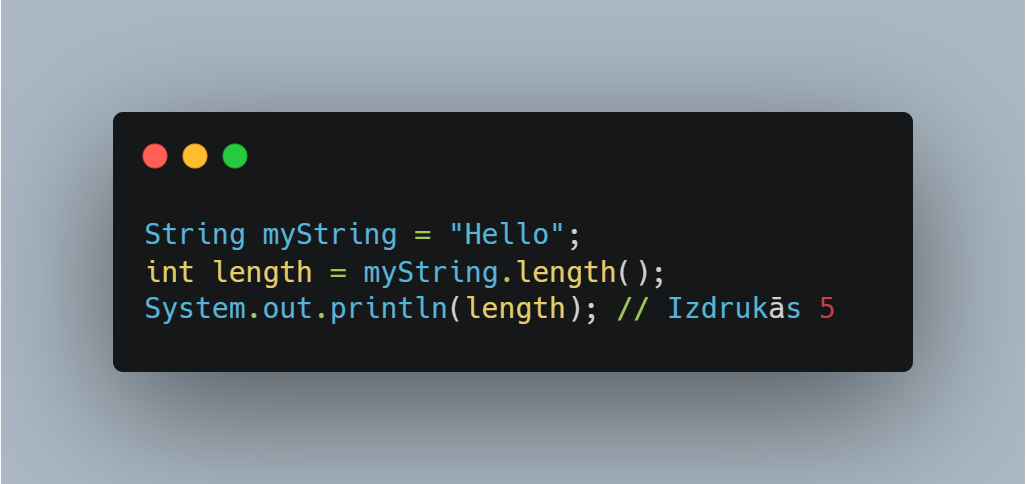
Simbolu virkne ir vairāku simbolu secība un to Java pārstāv ar **String** datu tipu. **String** ir īpašs datu tips, kas ļauj manipulēt ar simbolu virknēm. Simbolu virkni var deklarēt, izmantojot pēdiņas.

Piemēram:



**3. length() funkcija - simbolu virknes garums**

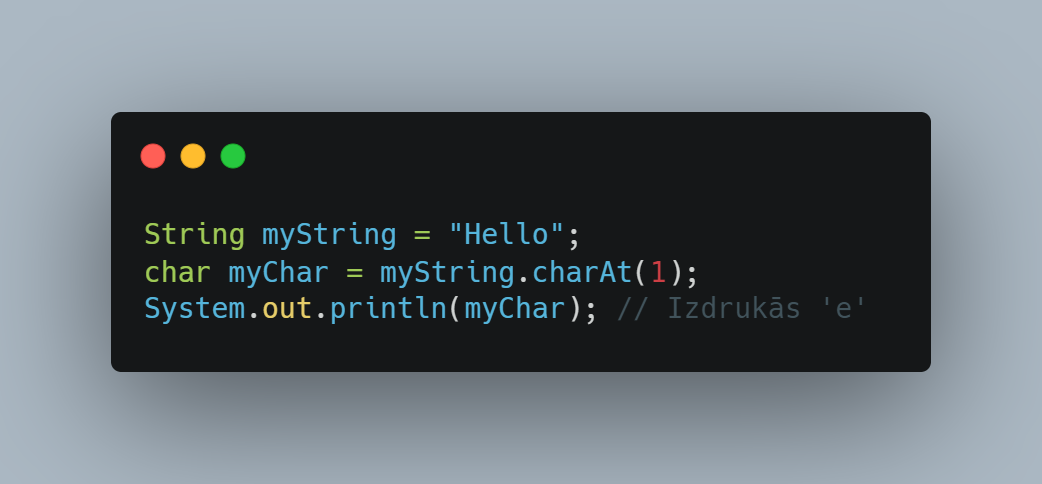
Lai iegūtu simbolu virknes garumu (simbolu skaitu), izmantojam **length()** funkciju. Tā ir metode, kas jāizsauc uz **String** objekta, un tā neprasa nekādus argumentus. Funkcija atgriež veselo skaitli, kas norāda simbolu virknes garumu.

Piemērs:

**4. charAt() funkcija - simbola iegūšana pēc pozīcijas**

Lai iegūtu simbolu no simbolu virknes, kas atrodas noteiktā pozīcijā, izmantojam **charAt()** funkciju. Funkcijai ir jānorāda pozīcija (indekss), no kuras jāatgriež simbols. Pozīcijas skaitīšana sākas no nulles.

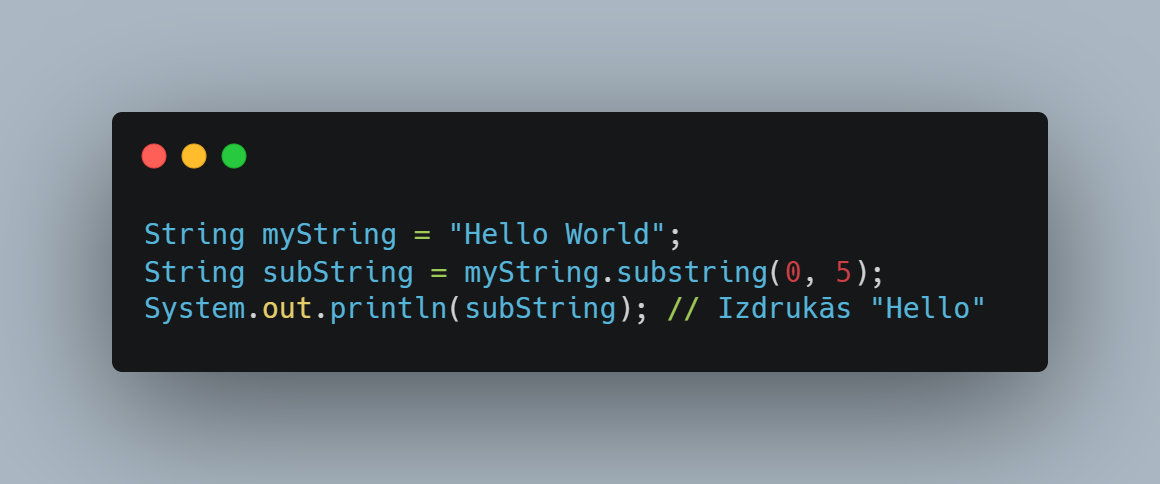
Piemērs:



**5. substring() funkcija - apakšvirknes izgūšana**

Lai izgūtu apakšvirkni no simbolu virknes, izmantojam **substring()** funkciju, norādot sākuma un beigu pozīcijas. Sākuma pozīcija tiek iekļauta, bet beigu pozīcija - neiekļauta rezultātā.

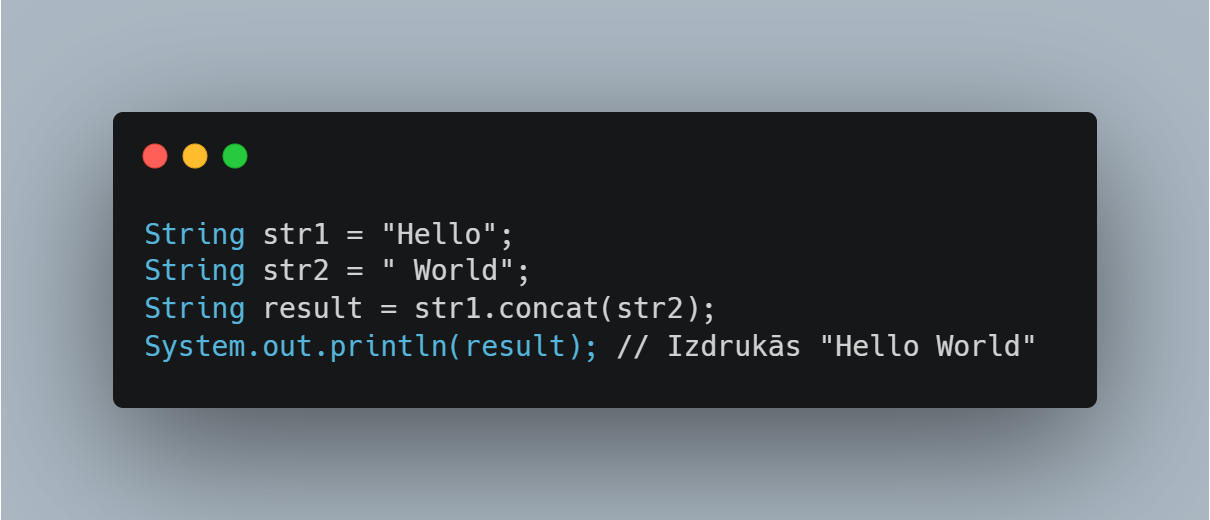
Piemērs:



1. **concat() funkcija - virkņu apvienošana**

Lai apvienotu divas virknes, izmantojam **concat()** funkciju. Tā pievieno norādīto virkni pašreizējai virknei.

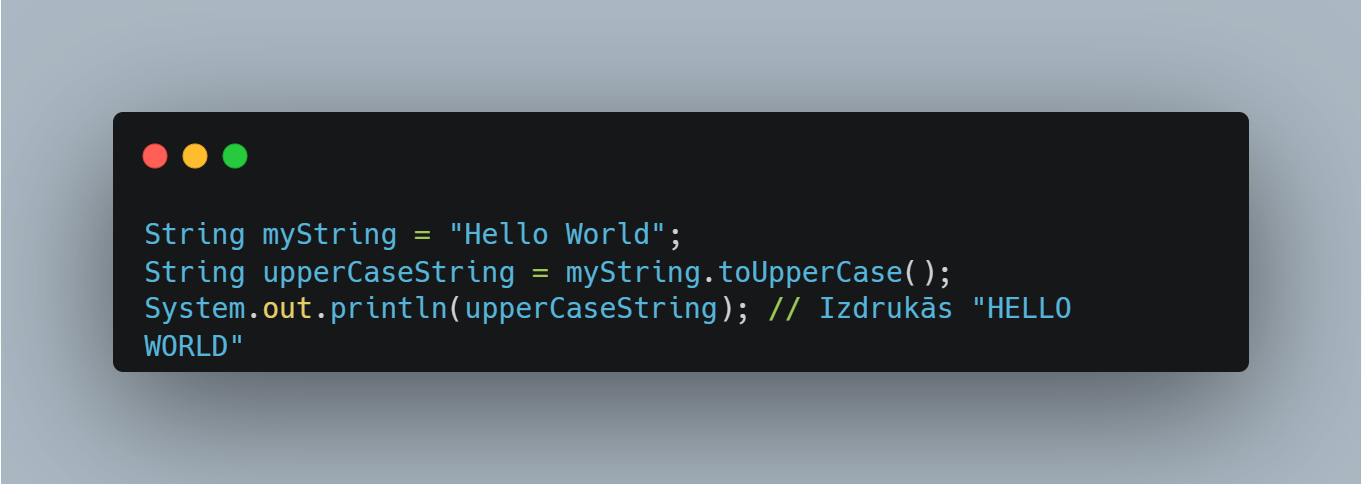
Piemērs:



1. **toUpperCase() funkcija - pārveido virkni uz lielajiem burtiem**

Lai pārveidotu visus simbolus virknē uz lielajiem burtiem, izmantojam **toUpperCase()** funkciju. Tā atgriež jaunu virkni, kurā visi burti ir pārveidoti uz lielajiem burtiem.

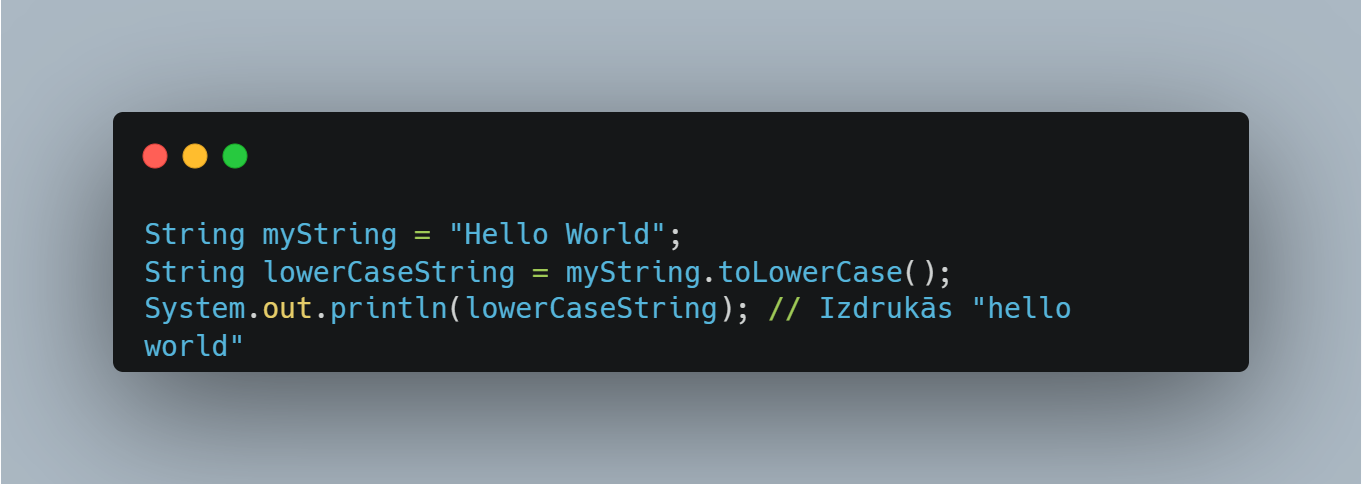
Piemērs:



1. **toLowerCase() funkcija - pārveido virkni uz mazajiem burtiem**

Lai pārveidotu visus simbolus virknē uz mazajiem burtiem, izmantojam **toLowerCase()** funkciju. Tā atgriež jaunu virkni, kurā visi burti ir pārveidoti uz mazajiem burtiem.

Piemērs:

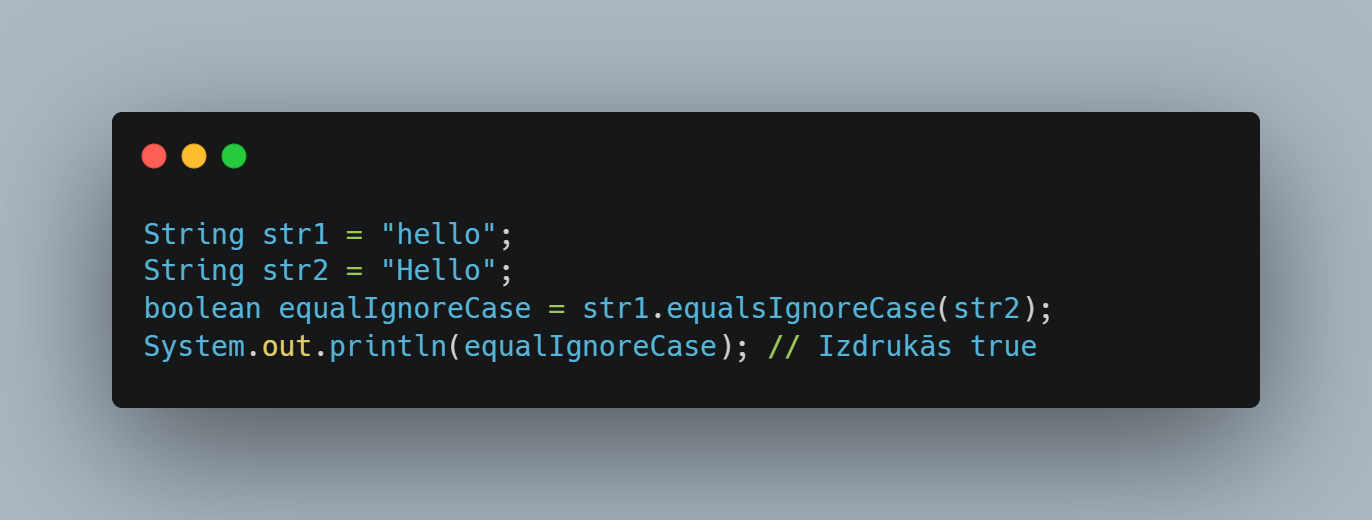


**toUpperCase()** un **toLowerCase()** funkcijas ir ļoti noderīgas, ja nepieciešams pārveidot simbolu virknes lielumu, piemēram, lai veiktu reģistru salīdzināšanu vai standartizētu virknes formatējumu. Šīs funkcijas ļauj vienkārši un efektīvi manipulēt ar simboliem un veikt operācijas ar simbolu virknēm, neatkarīgi no to lieluma.

1. **equalsIgnoreCase() funkcija - salīdzina virknes, ignorējot lielumu**

Lai salīdzinātu divas virknes, ignorējot lielumu, izmantojam **equalsIgnoreCase()** funkciju. Tā salīdzina virknes un atgriež boolean vērtību true, ja tās ir vienādas, ignorējot lielumu, un false pretējā gadījumā.

Piemērs:



1. **reverse() funkcija - apgriež simbolu virkni**

Lai apgrieztu simbolu virkni, izmantojam **reverse()** funkciju. Lai to izmantotu, jāimportē **java.lang.StringBuilder** un jāizveido jauns **StringBuilder** objekts ar sākotnējo virkni, kam pēc tam izsauc **reverse()** metodi.

Piemērs:



1. **startsWith() un endsWith() funkcijas - pārbauda, vai virkne sākas vai beidzas ar konkrētu simbolu vai virkni**

Lai pārbaudītu, vai virkne sākas vai beidzas ar konkrētu simbolu vai virkni, izmantojam **startsWith()** un **endsWith()** funkcijas. Abām funkcijām ir jānorāda simbols vai virkne, ar kuru salīdzināt sākumu vai beigas. Abas funkcijas atgriež boolean vērtību true, ja apskatāmā virkne sākas vai beidzas ar norādīto simbolu vai virkni, un false pretējā gadījumā.

Piemērs:



1. **split() funkcija - sadala simbolu virkni pa norādīto atdalītāju**

Lai sadalītu simbolu virkni pa norādīto atdalītāju un iegūtu apakšvirknes, izmantojam **split()** funkciju. Funkcijai ir jānorāda atdalītājs, un tā atgriež masīvu ar iegūtajām apakšvirknēm.

Piemērs:



Tālākās koda piemērus un attēlus var atrast programmēšanas grāmatās vai tiešsaistes resursos, piemēram(Izmantotie resursi):

* Oracle oficiālā Java dokumentācija: <https://docs.oracle.com/javase/10/docs/api/>
* Java for Beginners: <https://www.javaforbeginners.com/>
* W3Schools Java Tutorial: <https://www.w3schools.com/java/>

Šie resursi piedāvā plašu informāciju par Java programmēšanas valodu un ietver kodu piemērus, kas palīdzēs Jums uztvert un praktiski izmēģināt iepriekš minētās funkcijas darbam ar simboliem un simbolu virknēm.