

## C# - ОСНОВЕ

---

Уз овај документ ћеш проћи кроз основе C# програмског језика као и његову употребу унутар *Visual Studio Code*-а. У учионицама у којима се одржавају вјежбе имате двије опције за израду данашњих задатака: једна је да користите виртуелну машину и радите унутар *Visual Studio*-а окружења, а друга је да користите *Visual Studio Code* који је доступан изван виртуелне машине. Топла је препорука да користите онај доступан унутар виртуелне машине, као и да на својим персоналним рачунарима не користите *VS Code*, већ неко од интегрисаних окружења за развој (IDE-а): [Visual Studio](#) и [JetBrains Rider](#) су најпознатији (за оба алата можеш добити професионалне лиценце на основу факултетског имејла).

Уколико ћеш радити у *Code*-у ће ти требати одговарајући *plugin* за C#. Препорука је да у фолдеру *Documents* направиш фолдер са својим именом и индексом и све задатке смјестиш у њега. Исте налоге користе сви студенти, па је ово препоручен начин за рад да не изгубиш задатке од претходног пута. За сваки од задатака у наставку направи нови фолдер унутар твог фолдера и у том фолдеру направи нову апликацију у којој ћеш ријешити задатак.

Уколико ћеш радити у *Visual Studio*-у, немаш никаквих предуслова.

За сваки задатак направи нови пројекат/нову апликацију. За оне који желе да науче више: [погледајте](#) како да унутар једне апликације имате више пројеката.

## Развој конзолних апликација

За израду задатака који слиједе може ти значити [овај](#) и [овај](#) (мало старија верзија) документ, као и званичну [документацију](#).

1. Пратећи упутство са [VS/ VS Code](#), направи своју прву конзолну апликацију (енгл. *Console Application*). Уколико радите у *Code*-у, обратите пажњу на тачку 7 из упутства и користите ту синтаксу.
2. Направи још једну конзолну апликацију која учитава природан број  $n$  и као резултат испише  $n$ -ти члан Фибоначијевог низа.
3. Прочитај документ о [препорученим конвенцијама](#). Анализирај колико си испоштовао конвенције у претходна два задатка. Рефакториши претходне задатке тако да испоштујеш конвенције и труди се да их се придржавате убудуће.
4. Направи конзолну апликацију која учитава стринг и провјерава да ли је стринг палиндром.
5. Направи конзолну апликацију која учитава стринг произвољне дужине и серијализује га у текстуални фајл.

## Класе

6. Направи нову конзолну апликацију у којој ћеш вјежбати рад са класама. Направи нову класу аутомобил која има поља: модел, марка, боја, година производње, врсту горива и мотор. Врста горива је енумерација (дизел и бензин). Мотор је такође класа са идућим пољима: назив, кубикажа, снага.
7. Направи пар објеката класа мотор и аутомобил. За обје класе направи *ToString* методу и испиши објекте на конзолу.

## Инсталација и коришћење пакета

8. Пратећи упутство са [VS/VS Code](#), инсталирај *Newtonsoft.Json* пакет и тестирајте га на примјеру.
9. Вратите се на пројекат који си користио за вјежбање рада са класама. У новој функцији учитај *cars.json* фајл (налази се на канвасу у фолдеру вјежбе/термин 1) користећи *Newtonsoft.Json* пакет (користи ово подешавање приликом учитавања `new JsonSerializerSettings { PreserveReferencesHandling = PreserveReferencesHandling.Objects }`, а у документацији пакета пронађи како да га искористиш). На конзолу испиши:
  - a. Старост сваког аутомобила
  - b. Називе аутомобила који су највише 5 година стари
  - c. Просјечну старост
10. Из листе аутомобила издвоји све јединствене моторе и серијализуј их у фајл под називом *engines.json*.
11. У новој конзолној апликацији направи класу производ и учитај *items.json*. Испиши све производе којима рок трајања није истекао, па испиши оне којима истиче у року од 7 дана.
12. Производе из претходног задатка пребаци у ријечник тако да су кључеви шифре производа, а вриједности су сами производи. Користећи ријечник израчунај колико ће корисника коштати куповина два производа са шифром 1, једног производа са шифром 3 и четири производа са шифром 4.