

Окружење Visual Studio 2022

VS 2022 је окружење за програмирање коришћењем алатки које нуде функционалности неопходне за прављење C# пројеката који се извршавају под Windows оперативним системом. Такође је могуће конструисати пројекте у којима се комбинују модули написани у другим програмским језицима као што су C++, Visual Basic, F#.

Конзолна апликација је апликација која се извршава у прозору Command Prompt помоћу командних линија и нема графички кориснички интерфејс (GUI – graphical user interface).

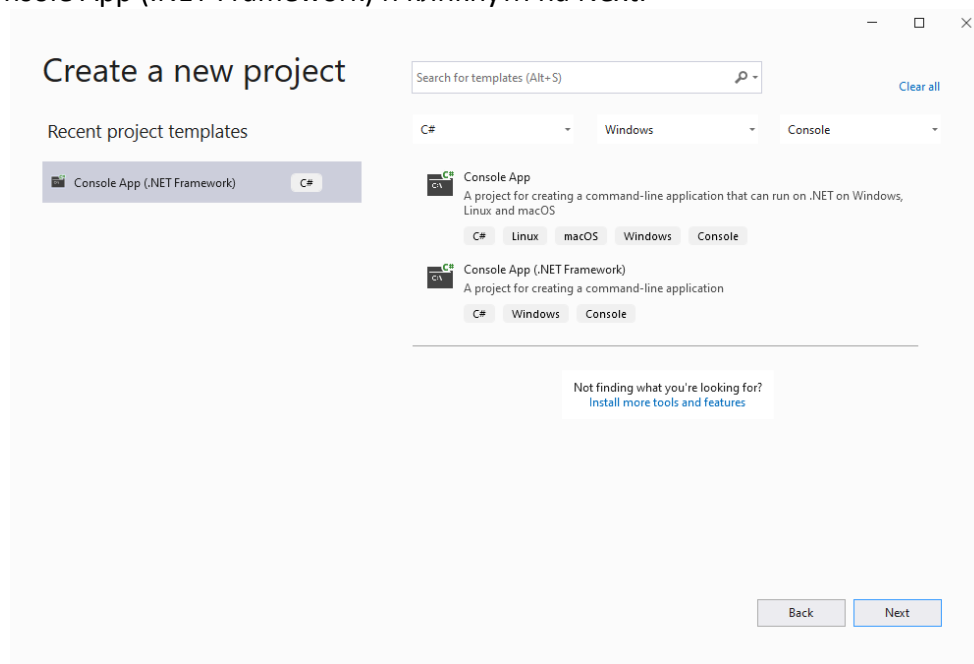
Прављење конзолне апликације у програмском окружењу Visual Studio 2022

Стартовати VS2022 и изабрати опцију Create a New Project.

На следећем екрану изабрати параметре којима се обезбеђује неопходно окружење за даљи рад у едитору C# програмског језика: C#, Windows, Console.

Тако се показује неколико могућих шаблона која се мођусобно разлику по броју и могућностима које пружају током рада.

Изабрати Console App (.NET Framework) и кликнути на Next.



Тако се отвара прозор Configure your new project.

Назив пројекта није толико битан, само да је на неки начин јединствен, али локације за тај пројекат мора бити везана за директоријум са именом ученика: C:\Users\ime_ucenika\Documents\C#

Configure your new project

Console App (.NET Framework) C# Windows Console

Project name

Location
 ...

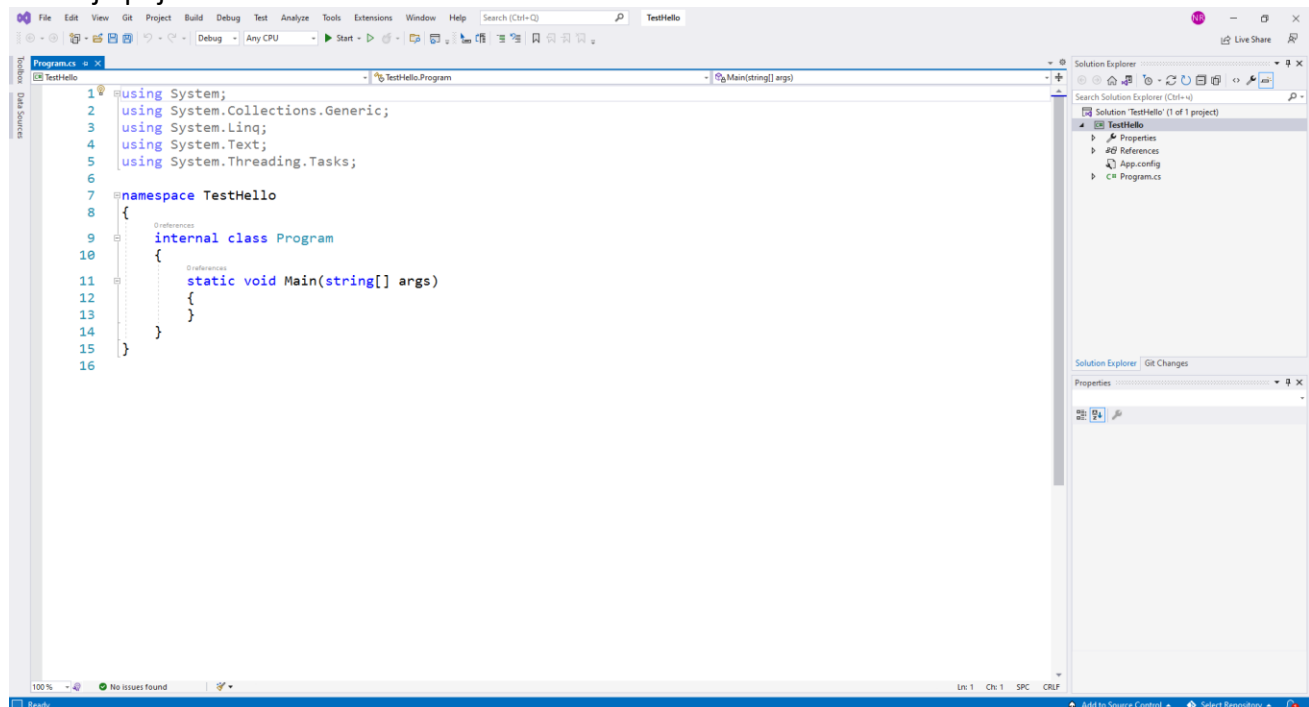
Solution name ⓘ

☐ Place solution and project in the same directory

Framework

Изабрати најновију инсталирану верзију .NET Framework која се нуди на списку инсталираних верзија и кликнути на Create.

VS2022 прави пројекат коришћењем шаблона Console Application а затим приказује почетни код за овај пројекат:



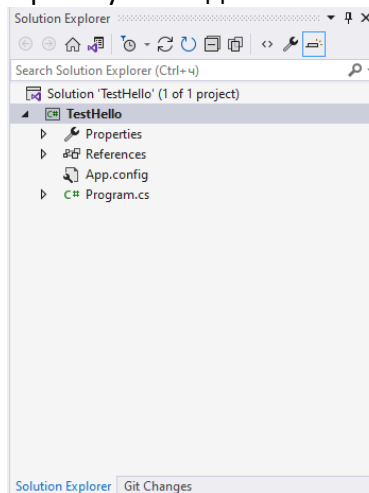
Главни мени на врху екрана нуди приступ опцијама које се користе у програмском окружењу. За приступање менију и командама могу се користити тастатура и миш као и у другим програмима под Windows оперативном систему. Палета алатки се налази испод главног менија и на њој су дугмад са пречицама до најчешће коришћених команди.

Прозор за унос и измене кода и текста, који заузима највећи део екрана, приказује садржај изворне датотеке (source file).

У пројекту са више датотека, када се обрађује више датотека, свака изворна датотека има своју картицу са називом одговарајуће изворне датотеке.

Притиском миша на одређену картицу изабрана изворна датотека се приказује испред осталих у прозору за унос кода.

Оквир за прегледање решења (Solution Explorer) се приказује на десној страни развојног окружења, непосредно поред прозора за унос кода:



У оквиру оквира за преглед решења се приказују називи датотека повезаних са одређеним пројектом.

Тренутно се у оквиру за преглед решења налазе:

- Solution 'TestHello' је датотека решења највишег нивоа. Свака апликација садржи једну датотеку решења. Једно решење може да обухвати више пројеката, а VS2022 прави датотеку решења како би олакшао организацију ових пројеката. Прави назив ове датотеке је TestHello.sin.
- TestHello је датотека пројекта у језику C#. Све датотеке пројекта указују на једну или више датотека које садрже изворни код и друге ствари везане за пројекат, као што су слике. Читав изворни код се мора написати у једном пројекту користећи исти програмски језик. Право име овог фајла је TestHello.csproj а чува се у посебном поддиректоријуму истог имена као назив фајла.
- Properties (својства) је фолдер пројекта TestHello и ако се прошири кликом на стрелицу поред имена, види се да садржи датотеку под именом Assemblyinfo.cs. Ова датотека се може користити за додавање атрибута програму, као што су име аутора, датум када је програм написан итд.
- References (референце) је фолдер који садржи референце на библиотеке компајлираног кода које апликација може да користи. Када се C# код компајлира, претвара се у библиотеку и добија јединствен назив. У окружењу .NET ове библиотеке се називају склопови (assemblies). Програмери користе склопове за паковање корисних функционалности које су написали тако да их могу делити са другим програмерима који желе те функционалности да користе у својим програмима. Проширењем фолдера References се види подразумевани скуп референци које је VS2022 додао у пројекат. Ови

склопови нуде приступ до бројних, најчешће коришћених могућности .NET Framework окружења.

- App.config је датотека за конфигурисање апликације и није обавезна. Могу се одредити параметри које апликација користи у тренутку извршавања како би се променило њено понашање, као што је верзија окружења .NET која се користи за покретање апликације.
- Program.cs је изворна датотека у језику C# и она се приказује у прозору за унос кода када се пројекат први пут направи. Програмер пише свој код за конзолну апликацију унутар ове датотеке. У њој се такође налази и код који VS2022 аутоматски нуди.

Изградња и покретање конзолне апликације

```
static void Main(string[] args)
{
    Console.WriteLine("Hello World!");
}
```

На менију Build кликнути на команду за грађење решења Build Solution.

Тиме се компајлира C# код и добија програм који може да се извршава.

Прозор Output (Резултат) се појављује испод прозора за уношење кода (или се пронађе на менију View).

Ако постоје грешке у коду, грешке ће бити пријављене у прозору Error List (листа грешака).

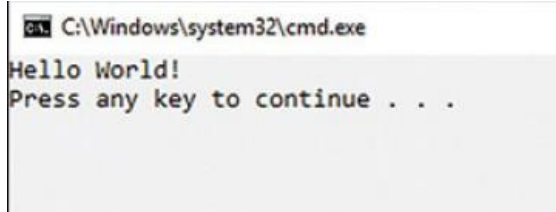
Најчешће, једна грешка пријављује више грешака при компајлирању због домино ефеката у коду. У VS ће се појавити црвена кривудава линија испод редова кода који нису исправно компајлирани.

Ако је програмер унео измене у коду, појавиће се звездица у имену датотеке: Program.cs*, и то значи да је датотека мењана после последњег меморисања садржаја датотеке.

Да би се измена сачувала може се или кликнути на икону Save all или кликнути на Build->Rebuild Solution.

Кликнути на Debug->Start without Debugging или са F5.

Отвара се командни прозор и покреће се програм.



```
C:\Windows\system32\cmd.exe
Hello World!
Press any key to continue . . .
```

Ако се кликне на Debug->Start Debugging извршава се апликација на идентичан начин али се командни прозор одмах затвара не чекајући да се кликне на било који тастер.

Улаз и излаз програма

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            //Izlaz u obliku teksta za korisnika
            Console.WriteLine("Ovo je pocetak programiranja u C#.");

            //Cekanje na pritisak tastera ENTER
            Console.ReadLine();
        }
    }
}

```

Линија кода `Console.WriteLine` исписује једну линију на конзоли за корисника.

Линија кода `Console.ReadLine` учитава линију текста који корисник уноси преко тастатуре.

Ипак, у овом случају, смисао овог исказа је да програм сачека корисника да притисне на тастатури дугме ЕНТЕР.

Ово се ради када желимо да корисник прочита садржај екрана конзоле пре гашења конзоле.

По дифолтним поставкама VS пројекти се аутоматски чувају пре сваког стартовања пројекта.

Да би се мануелно сачувао садржај пројекта: File->Save All или CTRL+S.

Ovo je pocetak programiranja u C#.

Press any key to continue . . .

Функционалност подвлачењем кода при креирању апликације је веома корисна опција унутар VS.

Ова функционалност се користи са службеним речима `#region` и `#endregion`, којима се дефинише почетак и крај дела кода који се може проширити или скупити.

Ако се модификује код:

```

#region using pretprocesorske direktive
using System;
#endregion

```

На овај начин се омогућава скупљање кода у једну линију и по потреби поновно проширење.

```

using pretprocesorske direktive

```

Задаци за самосталан рад

1. Написати код којим се добија следећи изглед екрана конзоле:

Milan.

Milanic.

Press any key to continue . . .

2. Написати код којим се добија следећи изглед екрана конзоле:

Milan Milanic.

Press any key to continue . . .

3. Написати код којим се добија следећи изглед екрана конзоле:

Bioskop	Film	Datum	Vreme

Dom sindikata	Ratovi zvezda	20.05.2018.	14:00
Sinapleks	Park iz doba jure	03.10.2018.	20:00

4. Написати код којим се добија следећи изглед екрана конзоле:

```

@@@@@@@@
@           @
@           @
@           @
@@@@@@@@

```

5. Постојећи код поделити у две групе: претпроцесорске директиве и сам код. Сваки од ових делова груписати преко #region и #endregion команди и проверити њихову функционалност у коду.