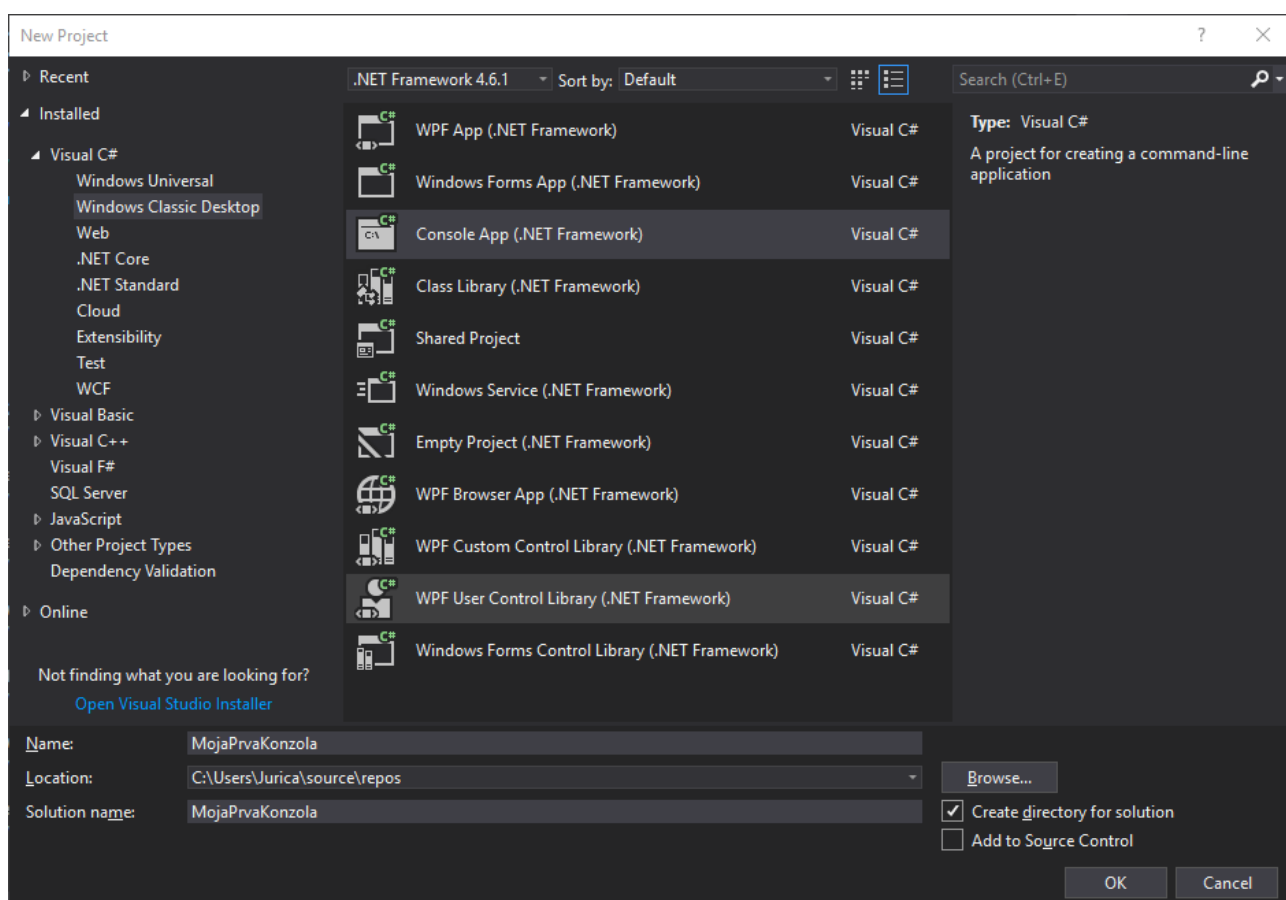


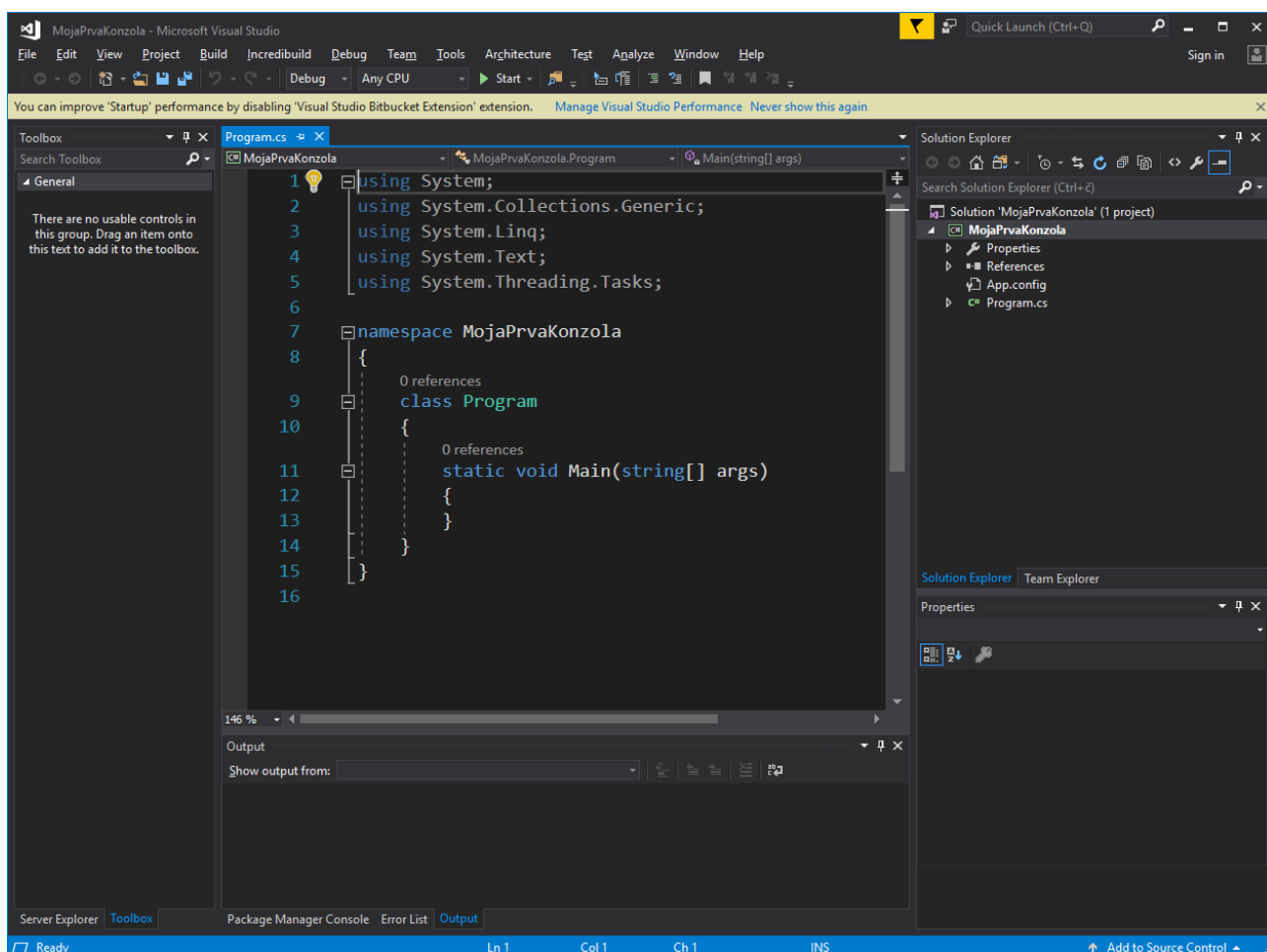
## Zadatak 1: Upoznavanje s *Visual Studio* razvojnom okolinom

1. Pokrenuti *Visual Studio*.
2. Otvoriti dijalog za kreiranje novog projekta: izbornik *File - New - Project*. Pogledati koji se sve tipovi projekata mogu kreirati.
3. U tipu projekta izabrati *Visual C#*, a u predlošcima odabrati *Console Application* te promijeniti naziv u „MojaPrvaKonzola“:



4. Nakon pritiska na tipku OK kreirat će se jednostavni projekt. Uočiti prozore koji su otvoreni u razvojnoj okolini. Raspored prozora i njihova vidljivost ovisi o postavkama u *Visual Studiju*. Većina prozora se aktivira iz izbornika *View*, *Debug* ili *Help*.
5. Posebnu pažnju obratiti na sljedeće prozore:
  - a. prozor s tekstom izvornog koda,
  - b. *Solution Explorer* (aktivira se iz *View* izbornika) sa stablom datoteka uključenih u projekt,
  - c. *Properties* prozor (*View* izbornik) koji prikazuje svojstva trenutno odabrane komponente,

- d. *Class View* (*View* izbornik) koji prikazuje hijerarhiju tipova u projektu,
- e. *Output* (*View* izbornik) koji sadrži ispis poruka tijekom izvođenja programa,
- f. *Index* (*Help* izbornik) sadrži abecedno kazalo pojmova u dokumentaciji.



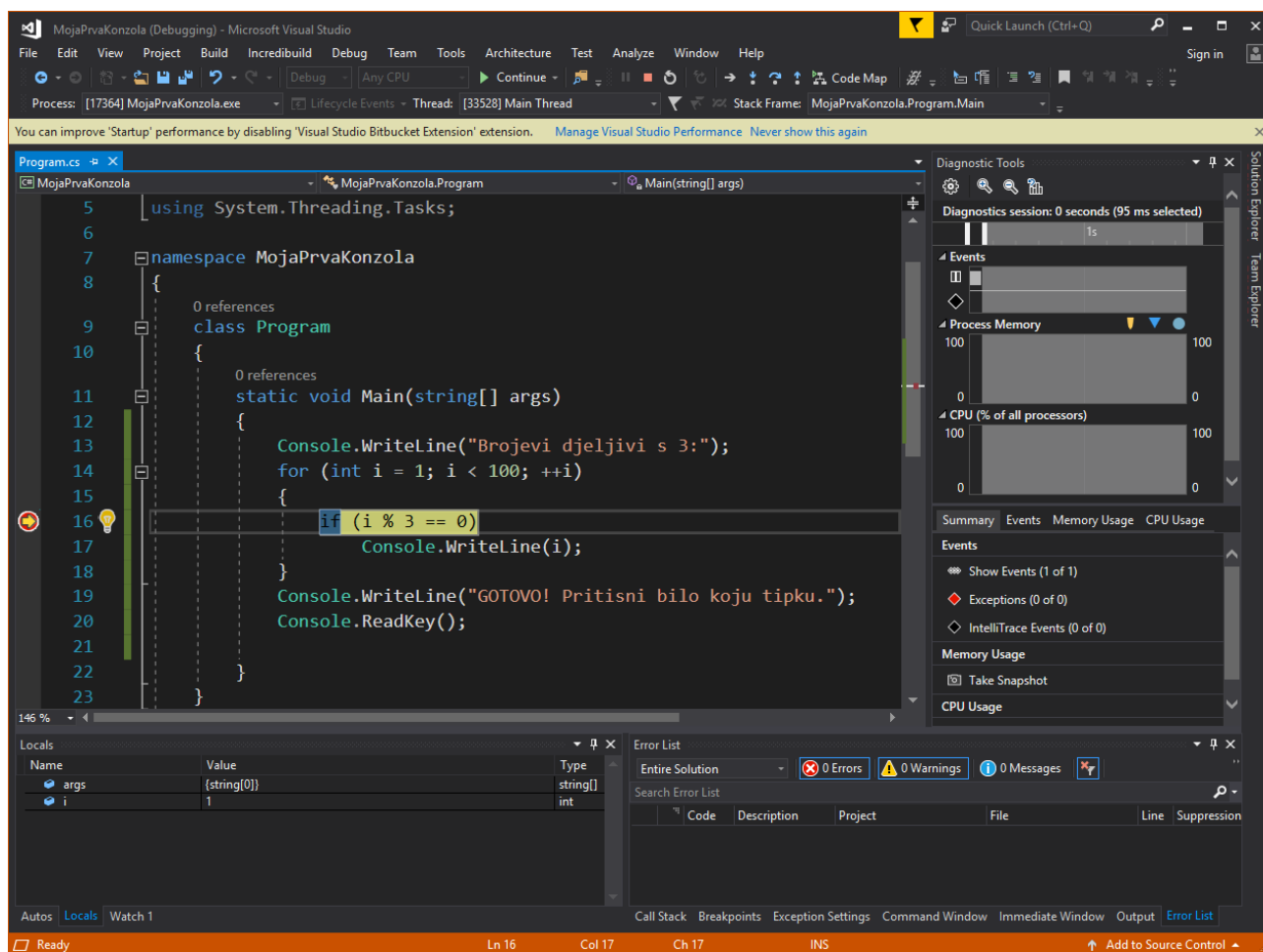
6. Pokrenuti prevođenje aplikacije naredbom *Build Solution* iz izbornika *Build* ili odgovarajućom kraticom (obično kombinacija tipki Ctrl+Shift+B). Uočiti ispis u *Output* prozoru.
7. Aplikaciju pokrenuti naredbom *Start Debugging* (izbornik *Debug*) ili odgovarajućom kraticom (F5). Uočiti kako se otvara i zatvara konzola te ispis u *Output* prozoru.
8. Obrisati jednu vitičastu zagradu iz koda te pokrenuti prevođenje aplikacije. Nakon neuspjelog prevođenja otvara se *Error* prozor s popisom pogrešaka u kodu. Dvostrukim klikom miša na pogrešku, fokus se automatski prebacuje u liniju koda izvornog koda gdje je pogreška.
9. Poništiti promjene (tj. unesenu pogrešku) u tekstu izvornog koda pomoću *Undo* naredbe iz *Edit* izbornika (Ctrl+Z ili Alt+BackSpace).

## Zadatak 2: Analiza izvođenja programa

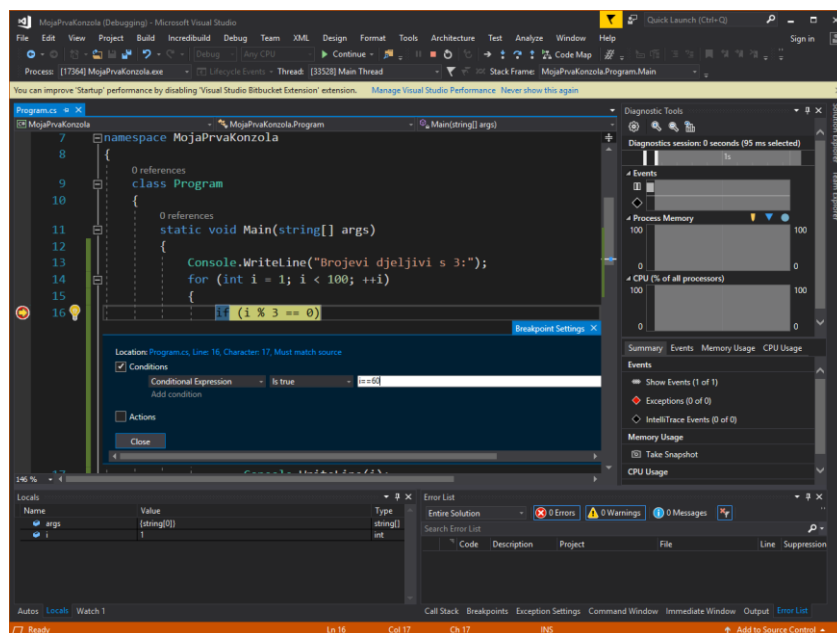
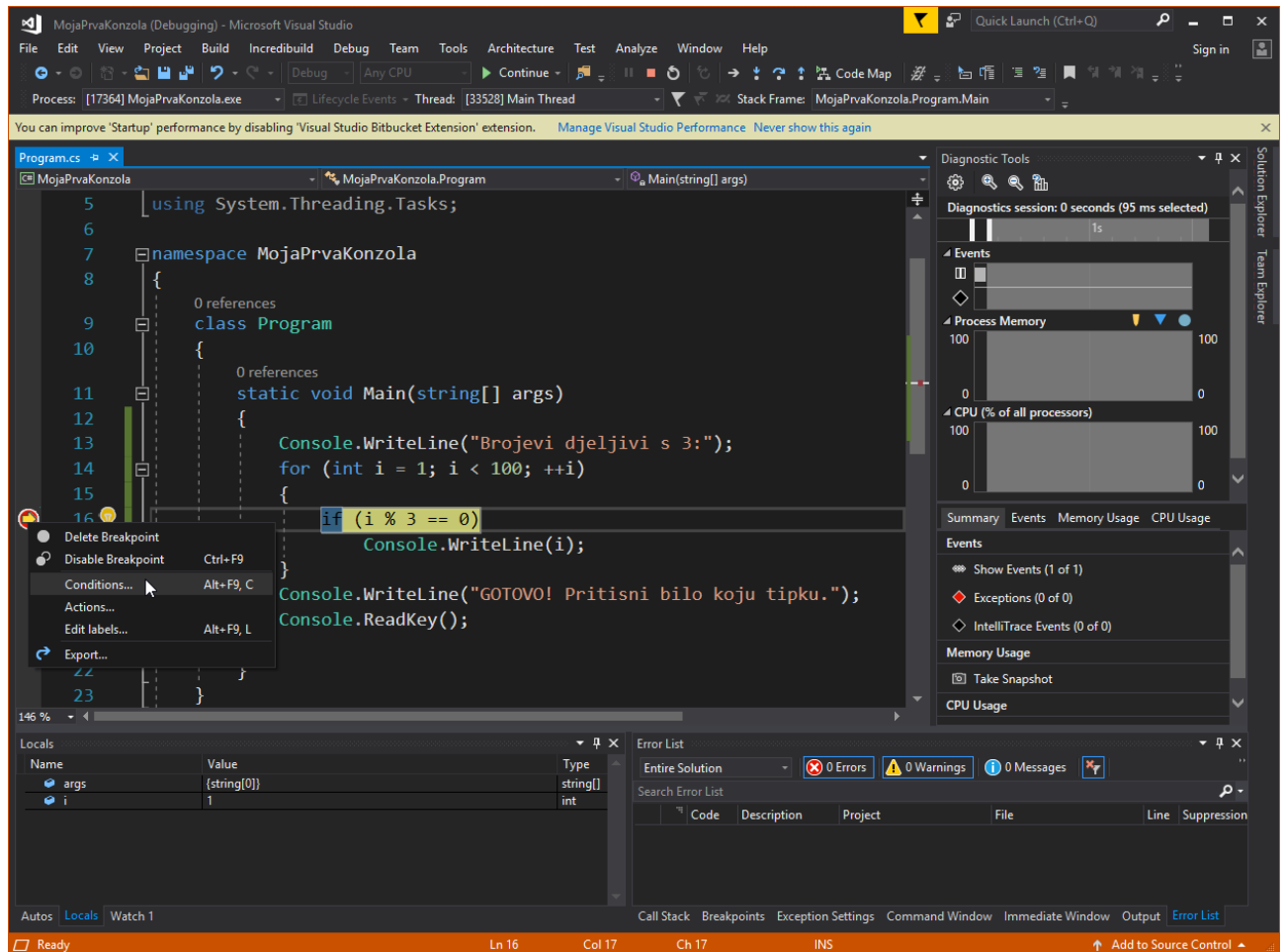
1. Unutar vitičastih zagrada metode Main dodati sljedeće naredbe:

```
Console.WriteLine("Brojevi djeljivi s 3:");  
for (int i = 1; i < 100; ++i)  
{  
    if (i % 3 == 0)  
        Console.WriteLine(i);  
}  
Console.WriteLine("GOTOVO! Pritisni bilo koju tipku.");  
Console.ReadKey();
```

2. Pokrenuti program (npr. tipkom F5 ili klikom na zelenu strelicu u alatnoj traci). Ako je program napisan bez pogreške, nakon prevođenja će se automatski pokrenuti i ispisati sve brojeve manje od 100 koji su djeljivi s 3. U slučaju pogreške, program se neće pokrenuti, već će se pojaviti dijalog s upozorenjem.
3. Po uspješnom izvođenju programa zatvoriti konzolu s ispisanim brojevima pritiskom na bilo koju tipku.
4. Unutar petlje, na naredbu if postaviti *točku prekida (breakpoint)*: postaviti treptajući kurzor u taj redak i iz izbornika *Debug* odabrati naredbu *Toggle Breakpoint* (F9). Lijevo od linije teksta pojavit će se crveni krug koji simbolizira točku prekida. Isto se može postići klikom miša na polje gdje se crveni krug pojavljuje.
5. Pokrenuti program tipkom F5. Izvođenje programa će se zaustaviti u točki prekida.



6. Otvoriti *Watch 1* prozor (iz izbornika *Debug - Windows - Watch*) koji omogućava pregledavanje vrijednosti objekata, te u njega upisati varijablu *i*. U *Watch* prozoru će se ispisati trenutna vrijednost varijable.
7. Nastaviti izvođenje programa korak-po-korak, naredbom *Step Over* (bilo iz izbornika *Debug*, tipkom F10 ili klikom miša na tipku na alatnoj traci). Nakon svakog koraka provjeriti *Watch* prozor. Uočiti kako se pri ulasku u početak petlje, nakon izvođenja naredbe *++i*, promijenjena vrijednost varijable *i* prikazuje crvenom bojom.
8. Pritisnuti tipku F5: program će nastaviti s izvođenjem do sljedeće točke prekida.
9. Dodati uvjet za prekid: desnom tipkom miša kliknuti na crvenu točku prekida te iz padajućeg izbornika izabrati *Condition*. U dijalog unijeti uvjet „*i == 60*“:



10. Nastaviti izvođenje programa tipkom F5; petlja će se sljedeći puta prekinuti u prolazu kada će biti zadovoljen uvjet prekida.
11. Kada se petlja sljedeći puta prekine, promijeniti vrijednost varijable `i` u *Watch* prozoru na 45 te nastaviti izvođenje. Uočiti u konzoli ponavljanje ispisanih vrijednosti.
12. Nakon ponovnog prekida petlje, izvesti sljedeće dvije naredbe korak-po-korak, tako da se u konzoli ispiše broj 60. Potom „uhvatiti“ mišem žutu strelicu uz lijevi rub prozora izvornog koda te ju povući gore na naredbu `if`. Time „ručno“ vraćamo izvođenje programa – ako sada nastavimo izvođenje programa, u konzoli će se ponovno ispisati vrijednost 60!

## Zadatak 2: Argumenti metode Main

### 1. Zadatak

Napisati konzolnu aplikaciju `Izracunaj` koja će izračunati jednostavni aritmetički izraz naveden kao parametar poziva. Na primjer:

```
Izracunaj 2 + 5
```

Pretpostaviti da su prvi i treći parametar brojevi tipa `double`, a drugi parametar je operator (+, -, \* ili /). Između brojeva i operatora moraju prilikom poziva programa biti praznine da bi ih metoda `Main` prihvatila kao zasebne argumente.

*Uputa:* Budući da su argumenti `Main` metode tipa `string`, treba ih pretvoriti u `double` koristeći statičku metodu

```
double.Parse()
```

koja kao argument prima `string`, a kao rezultat vraća broj tipa `double`.

Provjerite ispravnost programa za različite parametre.

### 2. Zadatak

U slučaju neispravnih argumenata `Parse` metoda može baciti iznimke. Provjerite u dokumentaciji kojeg tipa mogu biti te iznimke te dodajte `try-catch` blok koji će hvatati iznimke i ispisati korisniku razumljivu poruku.