

Osnove objektnog programiranja

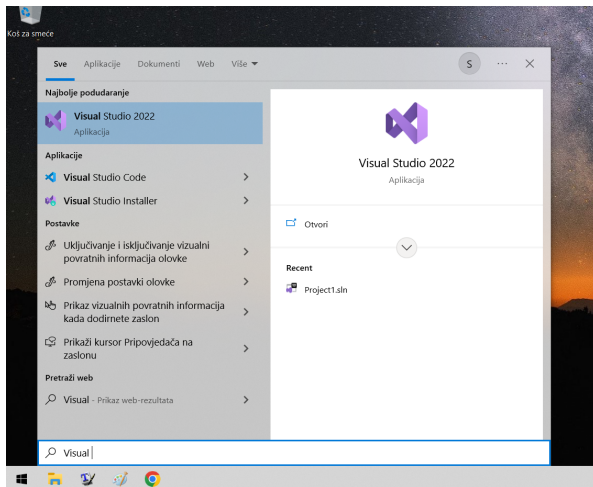
Objektno programiranje - 1. vježbe (2. dio)

Marko Živković

Prirodoslovno-matematički fakultet,
Sveučilište u Zagrebu

6. ožujka 2026.

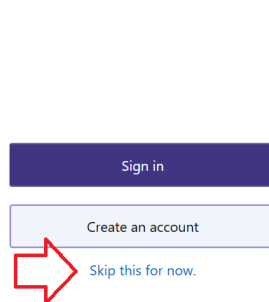
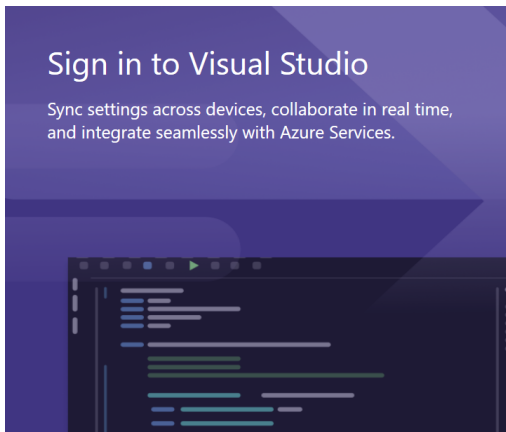
Pokretanje Visual Studia



Pokretanje Visual Studio



Pokretanje Visual Studio

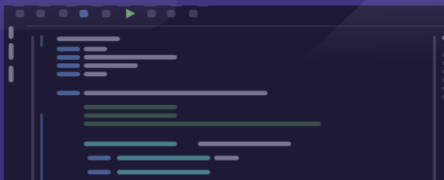


Odabrati temu po volji

Personalize your Visual Studio experience

Optimize the layout and keyboard shortcuts for your workflow. Select the color theme that fits your style.

You can always change these settings later.

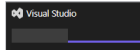


Development settings

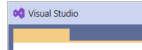
Visual C++

Choose your color theme

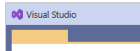
☐ Dark



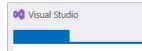
☒ Blue



☐ Blue (Extra Contrast)



☐ Light



Start Visual Studio

Stvaranje novog projekta

Visual Studio 2022

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

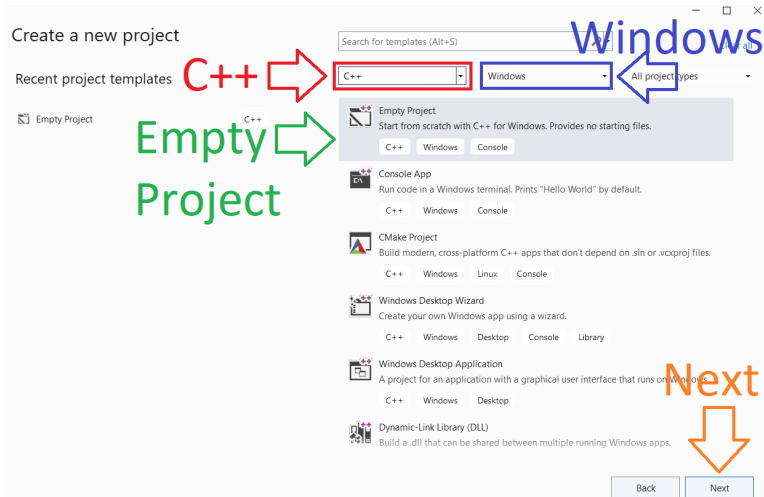
[Continue without code →](#)



Do you like this start window?



Stvaranje novog projekta (C++, Windows, Empty Project)



Naziv i mjesto spremanja novog projekta

Configure your new project

Empty Project C++ Windows Console

Project name

PrviProjekt

Location

C:\Projekti\

Solution name ⓘ

PrviProjekt

☐ Place solution and project in the same directory

Project will be created in "C:\Projekti\PrviProjekt\PrviProjekt"

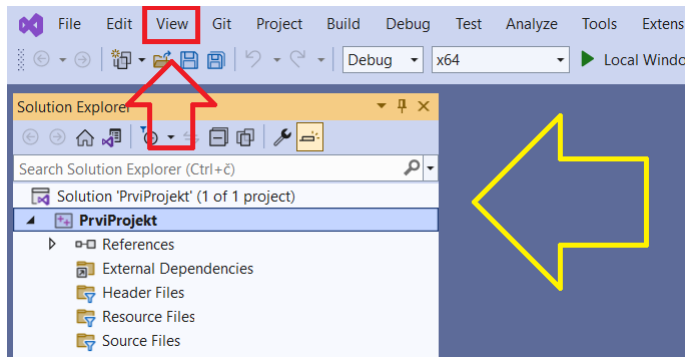
Back Create

Naziv projekta

Mjesto spremanja

Create

Napomena: *Solution Explorer*



- ako se ne prikazuje prozor *Solution Explorer* odabrati *View* → *Solution Explorer*
- slično ako nedostaje neki drugi prozor prikazan u prezentaciji

- Osigurajte da je Configuration postavljen na Debug i x64
- Dodajte novu .cpp datoteku u Source Files.

Prvi zadatak

Zadatak 1. Kompajlirajte i pokrenite sljedeći program (program učitava dva cijela broja i ispiše njihov zbroj).

```
1  #include <iostream>
2
3  int main() {
4      std::cout << "Unesite_dva_broja:_ " << std::endl;
5      int v1 = 0, v2 = 0;
6      std::cin >> v1 >> v2;
7      std::cout << "Zbroj_brojeva_" << v1 << "_i_" << v2
8              << "_je_" << v1 + v2 << std::endl;
9      return 0;
10 }
```

Zadatak 2. Napišite klasu `Student` koja će imati konstantnu varijablu `ime` i promjenjivu varijablu `godina`. Dodajte funkciju koja postavlja godinu (*setter*) i funkciju koja ispisuje sve podatke o studentu na standardni izlaz. Napišite `main` funkciju koji će koristiti sve te funkcije.

Prva klasa

```
1  #include <iostream>
2  using namespace std;
3
4  class Student {
5      const string ime;
6      int godina = 0;
7
8  public:
9      Student(string ime) : ime(ime) {} // konstruktor
10
11     void setGodina(int godina) {
12         this->godina = godina;
13     }
14
15     void ispisiPodatke() {
16         cout << "Student:_ " << ime;
17         if (godina > 0) cout << "\ngodina:_ " << godina;
18         cout << endl;
19     }
20 };
21
22 int main() {
23     Student prvi("Petar_Petrovic");
24     prvi.setGodina(3);
25     prvi.ispisiPodatke();
26 }
```

Zadatak 3. Klasu `Student` maknite iz datoteke u kojem je `main` funkcije te je deklarirajte u header datoteci (.h) i implementirajte u zasebnoj .cpp datoteci.

Main.cpp

```
1  #include <iostream>
2  #include "Student.h"
3
4  using namespace std;
5
6  int main() {
7      Student prvi("Petar_Petrovic");
8      prvi.setGodina(3);
9      prvi.ispisiPodatke();
10 }
```

Student.h - deklaracije

```
1 #pragma once
2 #include <string>
3
4 using namespace std;
5
6 class Student {
7     const string ime;
8     int godina = 0;
9
10    public:
11    Student(string ime);
12
13    void setGodina(int godina);
14
15    void ispisiPodatke();
16};
```


Student.cpp - implementacije

```
1  #include <iostream>
2  #include <string>
3  #include "Student.h"
4
5  using namespace std;
6
7  Student::Student(string ime) : ime(ime) {}
8
9  void Student::setGodina(int godina) {
10     this->godina = godina;
11 }
12
13 void Student::ispisiPodatke() {
14     cout << "Student:_ " << ime;
15     if (godina > 0) cout << "\ngodina:_ " << godina;
16     cout << endl;
17 }
```

Zadatak 4. Uvedite klasu `Predmet` i omogućite da student upiše i ispiše predmet. Neka funkcija `Student::ispisiPodatke` ispiše i listu predmeta koje je student upisao.

Predmet.h

```
1  #pragma once
2  #include <string>
3
4  using namespace std;
5
6  class Predmet {
7      const string naziv;
8
9  public:
10     Predmet(string naziv);
11
12     string getNaziv(); // getter
13 };
```

Predmet.cpp

```
1  #include <iostream>
2  #include <string>
3  #include "Predmet.h"
4
5  using namespace std;
6
7  Predmet::Predmet(string naziv) : naziv(naziv) {}
8
9  string Predmet::getNaziv() {
10     return naziv;
11 }
```

Student.h

```
1  #pragma once
2  #include <string>
3  #include <unordered_set>
4
5  using namespace std;
6
7  class Predmet; // prethodna deklaracija
8
9  class Student {
10     const string ime;
11     int godina = 0;
12     unordered_set<Predmet*> upisaniPredmeti; // moze i vector jer je skup malen
13
14 public:
15     Student(string ime);
16
17     void setGodina(int godina);
18
19     void upisiPredmet(Predmet*);
20
21     void ispisiPredmet(Predmet*);
22
23     void ispisiPodatke();
24 };
```

Student.cpp

```
1  #include <iostream>
2  #include <string>
3  #include "Predmet.h"
4  #include "Student.h"
5
6  ...
7
8  void Student::upisiPredmet(Predmet* predmet) {
9      upisaniPredmeti.insert(predmet);
10 }
11
12 void Student::ispisiPredmet(Predmet* predmet) {
13     upisaniPredmeti.erase(predmet);
14 }
15
16 void Student::ispisiPodatke() {
17     cout << "Student:_" << ime;
18     if (godina > 0) cout << "\ngodina:_" << godina;
19     cout << "\nupisani_predmeti:_";
20     for (Predmet* predmet : upisaniPredmeti) {
21         cout << predmet->getNaziv() << ",_";
22     }
23     cout << "\n" << endl;
24 }
```

Main.cpp

```
1  #include <iostream>
2  #include "student.h"
3  #include "predmet.h"
4
5  using namespace std;
6
7  int main() {
8      Student prvi("Petar_Petrovic");
9      prvi.setGodina(3);
10     Student drugi("Ana_Horvat");
11     drugi.setGodina(2);
12
13     Predmet p1("Elementarna_matematika");
14     Predmet p2("Matemacka_analiza");
15     Predmet p3("Objektno_programiranje");
16
17     prvi.upisiPredmet(&p1);
18     prvi.upisiPredmet(&p2);
19     prvi.upisiPredmet(&p3);
20     drugi.upisiPredmet(&p1);
21
22     prvi.ispisiPodatke();
23     drugi.ispisiPodatke();
24
25     prvi.ispisiPredmet(&p2);
26     prvi.ispisiPodatke();
27 }
```

Zadatak 5. Dodajte mogućnost da klasa `Predmet` ispiše listu studenata koji su upisali taj predmet.

Predmet.h

```
1  #pragma once
2  #include <string>
3  #include <unordered_set>
4
5  using namespace std;
6
7  class Student; // prethodna deklaracija
8
9  class Predmet {
10     string naziv;
11     unordered_set<Student*> studenti;
12
13 public:
14     Predmet(string naziv);
15
16     void upisiStudenta(Student* student);
17     void ispisiStudenta(Student* student);
18
19     string getNaziv();
20
21     void ispisiListuStudenata();
22 };
23
```

Predmet.cpp

```
1  #include <iostream>
2  #include <string>
3  #include "Predmet.h"
4  #include "Student.h"
5
6  using namespace std;
7
8  Predmet::Predmet(string naziv) : naziv(naziv) {}
9
10 void Predmet::upisiStudenta(Student* student) {
11     studenti.insert(student);
12 }
13
14 void Predmet::ispisiStudenta(Student* student) {
15     studenti.erase(student);
16 }
17
18 string Predmet::getNaziv() {
19     return naziv;
20 }
21
22 void Predmet::ispisiListuStudenata() {
23     cout << naziv << ":_";
24     for (Student* student : studenti) {
25         cout << student->getName() << ",_";
26     }
27     cout << "\n" << endl;
28 }
```

Student.h

```
1  #pragma once
2  #include <string>
3  #include <unordered_set>
4
5  using namespace std;
6
7  class Predmet;
8
9  class Student {
10     const string ime;
11     int godina = 0;
12     unordered_set<Predmet*> upisaniPredmeti;
13
14 public:
15     Student(string ime);
16
17     void setGodina(int godina);
18
19     void upisiPredmet(Predmet*);
20
21     void ispisiPredmet(Predmet*);
22
23     string getName(); // novo, getter za ispis liste studenata
24
25     void ispisiPodatke();
26 };
```

Student.cpp

```
1  ...
2
3  void Student::upisiPredmet(Predmet* predmet) {
4      upisaniPredmeti.insert(predmet);
5      predmet->upisiStudenta(this);
6  }
7
8  void Student::ispisiPredmet(Predmet* predmet) {
9      upisaniPredmeti.erase(predmet);
10     predmet->ispisiStudenta(this);
11 }
12
13 string Student::getName() {
14     return ime;
15 }
16
17 ...
```

main()

```
1  int main() {
2      Student prvi("Petar_Petrovic");
3      prvi.setGodina(3);
4      Student drugi("Ana_Horvat");
5      drugi.setGodina(2);
6
7      Predmet p1("Elementarna_matematika");
8      Predmet p2("Matematička_analiza");
9      Predmet p3("Objektno_programiranje");
10
11     prvi.upisiPredmet(&p1);
12     prvi.upisiPredmet(&p2);
13     prvi.upisiPredmet(&p3);
14     drugi.upisiPredmet(&p1);
15
16     prvi.ispisiPodatke();
17     drugi.ispisiPodatke();
18
19     prvi.ispisiPredmet(&p2);
20
21     prvi.ispisiPodatke();
22
23     p1.ispisiListuStudenata();
24     p2.ispisiListuStudenata();
25     p3.ispisiListuStudenata();
26 }
```

Dodajmo sljedeće retke u `main()`:

```
1 p2.upisiStudenta(&prvi);  
2 p2.ispisiListuStudenata();  
3 prvi.ispisiPodatke();
```

Dobit ćemo da predmet `p2` ima upisanog studenta `prvi`, ali da student `prvi` nema upisan predmet `p2`. To ne bi smjelo biti moguće.

Zadatak 6. Onemogućite da predmet upiše studenta, a da student istovremeno ne upiše predmet. Isto za ispis.

Ideja: Upis studenta u predmet ne smije biti javno dostupan, ali treba biti dostupan u funkciji `Student::upisiPredmet`.

Predmet.h

```
1  #pragma once
2  #include <string>
3  #include <unordered_set>
4
5  using namespace std;
6
7  class Student; // prethodna deklaracija
8
9  class Predmet {
10     string naziv;
11     unordered_set<Student*> studenti;
12
13 public:
14     Predmet(string naziv);
15
16     friend class Student;
17
18     string getNaziv();
19
20     void ispisiListuStudenata();
21 };
```

Predmet.cpp

```
1  #include <iostream>
2  #include <string>
3  #include "Predmet.h"
4  #include "Student.h"
5
6  using namespace std;
7
8  Predmet::Predmet(string naziv) : naziv(naziv) {}
9
10 string Predmet::getNaziv() {
11     return naziv;
12 }
13
14 void Predmet::ispisiListuStudenata() {
15     cout << naziv << ":_";
16     for (Student* student : studenti) {
17         cout << student->getName() << ",_";
18     }
19     cout << "\n" << endl;
20 }
```


Student.cpp

```
1  ...
2
3  void Student::upisiPredmet(Predmet* predmet) {
4      upisaniPredmeti.insert(predmet);
5      predmet->studenti.insert(this);
6  }
7
8  void Student::ispisiPredmet(Predmet* predmet) {
9      upisaniPredmeti.erase(predmet);
10     predmet->studenti.erase(this);
11 }
12
13 ...
```