

# Programinio kodo derinimas (debugging) naudojant Visual Studio

Informatika (IV klasė)

# Ką išmoksime šiandien

- Kaip naudoti Visual Studio derinimo įrankius (breakpoints, Watch Window, Step Over/Into),
- Kaip identifikuoti ir ištaisyti klaidas Python arba C++ kode,
- Praktiškai išbandysime derinimo technikas su Python arba C++ kodu.

# Kas yra derinimas ir kodėl jis svarbus?

## Kas yra derinimas?

Derinimas – tai klaidų paieška ir taisymas programiniame kode. Jis padeda užtikrinti, kad programa veiktų teisingai ir atitiktų vartotojų lūkesčius.

## Kodėl derinimas svarbus?

- Net ir maža klaida gali sukelti dideles problemas (pvz., neteisingus skaičiavimus, programos sutrikimus).
- Derinimas padeda užtikrinti programos kokybę ir patikimumą.
- Tai pagrindinis įgūdis, kurio reikia kiekvienam programuotojui.

# Klaidos poveikis

## Pavyzdys

Įsivaizduokite, kad turite programą, kuri skaičiuoja dviejų skaičių sumą.

Jei kviesite šią funkciją su skaičiais **5** ir **3**, ji grąžins **2**, nors teisingas rezultatas turėtų būti **8**.

Tokia klaida gali sukelti neteisingus skaičiavimus, pvz., finansinėse programose ar mokslinėse skaičiavimo sistemose.

***Net maža klaida gali turėti didelį poveikį, todėl derinimas yra būtinas programuotojų darbo dalis.***

## Kodas su klaida

```
def calculate_sum(a, b):  
    # Klaida: naudojamas  
    # minusas vietoj pluso  
    return a - b
```

```
int calculate_sum(int a, int b) {  
    //Klaida: naudojamas minusas  
    //          vietoj pluso  
    int result = a - b;  
    return result;  
}
```

# Pagrindinės derinimo technikos

- **Breakpoints:** Sustabdyti programos vykdymą tam tikrame kodo taške.
- **Stebėjimo langas (Watch Window):** Stebėti kintamųjų reikšmes vykdymo metu.
- **Žingsnis po žingsnio vykdymas (Step Over, Step Into):** Analizuoti kodo vykdymą po vieną eilutę.
- **Klaidų žinutės:** Analizuoti klaidas, kurias išmeta kompiliatorius arba vykdymo metu.

# Praktinė užduotis

[https://github.com/agorbatniov/inf12/tree/main/Program%C5%B3\\_testavimas\\_ir\\_taisyimas\\_\(VS\\_Debugging\)](https://github.com/agorbatniov/inf12/tree/main/Program%C5%B3_testavimas_ir_taisyimas_(VS_Debugging))

- **Python:** 01\_kodas\_su\_klaida.py
- **C++:** 01\_kodas\_su\_klaida.cpp

# Žingsniai derinimui

- Atidaryti projektą Visual Studio.
- Nustatyti breakpoint kodo eilutėje, kurioje norima pradėti analizę.
- Paleisti programą derinimo režime (Debug -> Start Debugging **F5**).
- Stebėti kintamųjų reikšmes naudojant Watch Window.
- Naudoti "Step Over **F10**" ir "Step Into **F11**" mygtukus, kad analizuotumėte kodo vykdymą.
- Surasti klaidą ir ją ištaisyti.

# Praktinė užduotis: Debugging Python kode (failas: 02\_kodas\_su\_klaida.py)

Python kodas, kuris turėtų apskaičiuoti studentų pažymių vidurkį ir nustatyti, ar **studentas išlaikė egzaminą (vidurkis  $\geq 5$ )**. Tačiau kode yra klaidų, kurias reikia surasti ir ištaisyti.

```
1 def calculate_average(grades):
2     total = 0
3     for grade in grades:
4         total += grade
5     average = total / len(grades)
6     return average
7
8 def is_passed(average):
9     if average > 5:
10         return True
11     else:
12         return False
13
14 # Pagrindinė programa
15 grades = [8, 7, 6, 9, 4] # Pavyzdiniai pažymiai
16 average = calculate_average(grades)
17 print("Vidurkis:", average)
18 print("Ar išlaikė egzaminą?", is_passed(average))
```



# Praktinė užduotis: Debugging C++ kode (failas: 02\_kodas\_su\_klaida.cpp)

C++ kodas, kuris turėtų apskaičiuoti stačiakampio plotą ir perimetrą. Tačiau kode yra klaidų, kurias reikia surasti ir ištaisyti.

```
1  #include <iostream>
2  using namespace std;
3
4  // Funkcija, kuri apskaičiuoja stačiakampio plotą
5  double calculate_area(double length, double width) {
6      return length + width;
7  }
8
9  // Funkcija, kuri apskaičiuoja stačiakampio perimetrą
10 double calculate_perimeter(double length, double width) {
11     return 2 * (length + width);
12 }
13
14 int main() {
15     double length = 5.0;
16     double width = 3.0;
17
18     double area = calculate_area(length, width);
19     double perimeter = calculate_perimeter(length, width);
20
21     cout << "Plotas: " << area << endl;
22     cout << "Perimetras: " << perimeter << endl;
23
24     return 0;
25 }
```

# Instrukcijos

## **Python užduotis:**

- Atidarykite Visual Studio ir sukurkite naują Python projektą
- Įklijuokite aukščiau pateiktą Python kodą
- Sukurkite Git Repository savo GitHub'e
- Naudokite breakpoints, kad sustabdytumėte programos vykdymą ir stebėtumėte kintamųjų reikšmes
- Suraskite klaidas ir ištaisykite jas
- Išsaugokite pakeitimus GitHub'e

## **C++ užduotis:**

- Atidarykite Visual Studio ir sukurkite naują C++ projektą.
- Įklijuokite aukščiau pateiktą C++ kodą.
- Sukurkite Git Repository savo GitHub'e
- Naudokite breakpoints, kad sustabdytumėte programos vykdymą ir stebėtumėte kintamųjų reikšmes.
- Suraskite klaidas ir ištaisykite jas.
- Išsaugokite pakeitimus GitHub'e

# Praktinė užduotis 2: Debugging

## Python užduotis

Python kodas, kuris turėtų rasti didžiausią ir mažiausią skaičių sąraše, taip pat apskaičiuoti jų sumą. Tačiau kode yra klaidų, kurias reikia surasti ir ištaisyti.

(failas: 03\_kodas\_su\_klaida.py)

## C++ užduotis

C++ kodas, kuris turėtų apskaičiuoti matricos eilučių sumas ir rasti didžiausią sumą. Tačiau kode yra klaidų, kurias reikia surasti ir ištaisyti.

(failas: 03\_kodas\_su\_klaida.cpp)

# Apibendrinimas

## **Ką išmokome?**

- Kaip naudoti Visual Studio derinimo įrankius.
- Kaip identifikuoti ir ištaisyti klaidas programiniame kode.

## **Kodėl tai svarbu?**

- Derinimas padeda užtikrinti programų kokybę ir veikimą.

# Papildoma informacija

## Literatūra:

- "Debugging Techniques in Visual Studio" by Dan Clark
- "Effective Debugging: 66 Specific Ways to Debug Software" by Diomidis Spinellis

## Internetiniai šaltiniai:

- [Visual Studio Debugging Documentation](#)