Algoritmų parinkimas.

„For“ ciklai

Ciklų sąvoka ir vaidmuo algoritmuose

- **Ciklų sąvoka**
- **Ciklas** – tai programavimo struktūra, leidžianti pakartotinai vykdyti tam tikrą kodo bloką, kol yra tenkinama tam tikra sąlyga.

- **Ciklų tipai:**
- **For ciklas:** Naudojamas, kai žinome, kiek kartų reikia vykdyti kodo bloką.
- **While ciklas:** Naudojamas, kai nežinome, kiek kartų reikės vykdyti kodo bloką, ir jis tęsis tol, kol sąlyga bus teisinga.

- **Vaidmuo algoritmuose**
- **Efektyvumas:** Ciklai padeda spręsti pasikartojančias užduotis efektyviau, sumažinant kodo apimtį ir padidinant jo aiškumą.
- **Automatizavimas:** Leidžia automatizuoti užduotis, kurios reikalauja daugybės pasikartojimų, pvz., duomenų apdorojimo, skaičiavimo ir kitų algoritmų.

Ciklų nauda

- **Ciklų vaidmuo programavime**
- **Pakartotinis vykdymas:** Ciklai leidžia pakartotinai vykdyti tam tikrą kodo bloką be poreikio rašyti tą patį kodą daugybę kartų.

- **Pasikartojančių užduočių sprendimas:**
- Ciklai yra idealūs, kai reikia atlikti užduotis, kurios kartojasi, pavyzdžiui:
  - Skaičių suma
  - Elementų iš sąrašo peržiūra
  - Duomenų apdorojimas (pvz., skaičiuojant vidurkį ar maksimalų skaičių)

Pavyzdys

Jei reikia išspausdinti skaičius nuo 1 iki 10, naudojant ciklą, tai galima padaryti labai efektyviai

Kodo Pavyzdys

```python
# Python

for i in range(5):
    print(i)

```

```cpp
// C++

for (int i = 0; i < 5; i++) {
    cout << i << endl;
}
```

Ciklo struktūros dalys

- **Inicializacija**
  - Tai pradinis ciklo kintamųjų nustatymas.
  - Kintamasis dažniausiai naudojamas sekimui, kiek kartų ciklas jau buvo vykdytas.
  - *Pavyzdys:* `i = 0` (Python), `int i = 0` (C++).

- **Sąlyga**
  - Sąlyga nurodo, kada ciklas turėtų baigtis.
  - ***Kol sąlyga teisinga*** (True), ciklas vykdys savo kodo bloką.
  - *Pavyzdys:* `i < 10`

- **Inkrementas**
  - Po kiekvieno ciklo įvykdymo ciklo kintamasis yra atnaujinamas (dažniausiai padidinamas).
  - Tai užtikrina, kad ciklas artėja link pabaigos.
  - *Pavyzdys:* `i += 1` (Python), `i++` (C++).

Python ir C++ sintaksės skirtumai

- **Python**:
- Naudoja **įtraukas** (indentation), kad parodytų kodo blokų ribas.
- Kiekvienas sąlyginio sakinio blokas turi būti tinkamai įtrauktas.

- **C++**:
- Naudoja **riestinius skliaustus {}** kodo blokams nurodyti.
- Kodo blokai gali būti parašyti be papildomų įtraukų (nors rekomenduojama naudoti įtraukas siekiant aiškumo).

- **Esminis skirtumas:**
- **Python** pasikliauja įtraukomis kaip būtina sintaksės dalimi.
- **C++** naudojasi riestiniais skliaustais **{}**, kurie leidžia lankstumą, bet padidina klaidų tikimybę, jei nėra tinkamai naudojamos įtraukos

Algoritmas naudojant „For“ ciklus

Pavyzdinė užduotis:

- Spausdinti skaičių kvadratus nuo 1 iki n

```python
# Python

n = int(input("Įveskite skaičių n: "))
suma = 0
for i in range(1, n + 1):
    suma += i
print("Suma:", suma)
```

```cpp
// C++
int n, suma = 0;
cout << "Įveskite skaičių n: ";
cin >> n;
for (int i = 1; i <= n; i++) {
    suma += i;
}
cout << "Suma: " << suma << endl;
```

Santrauka

- Ciklų sąvoka ir vaidmuo algoritmuose
- Ciklų nauda
- Ciklo struktūros dalys
- Algoritmai naudojant „For“ ciklus