

**C++ DASTURLASH TILI BOSHQARUV
OPERATORLARI: TAKRORLANISH OPERATORI**

**ОПЕРАТОРЫ УПРАВЛЕНИЯ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++:
ОПЕРАТОР ЦИКЛА**

**CONTROL OPERATORS IN C++ PROGRAMMING LANGUAGE: LOOP
OPERATOR**

Shermatova Xilola Mirzayevna

Farg'ona davlat universiteti Axborot texnologiyalari kafedrasida dotsenti

shermatovahilola1978@gmail.com

Ma'rufjonov Muqimjon Alohiddin o'g'li

*Farg'ona davlat universiteti Axborot tizimlari va texnologiyalar yo'nalishi
1-kurs talabasi*

Annotatsiya: Ushbu maqola C++ dasturlash tilidagi takrorlanish operatorlariga bag'ishlangan bo'lib, takrorlanish operatorlarining turlari, sintaksisi, ishlash mexanizmi va ularni real dasturlash masalalarida qo'llanilishi haqida batafsil ma'lumot beradi. C++ dasturlash tilida takrorlanish operatorlari dasturchiga bir necha marta bir xil kodni takrorlamasdan, dastur oqimini samarali boshqarish imkonini beradi. Maqola, asosan, boshlang'ich va o'rta darajadagi dasturchilar uchun mo'ljallangan bo'lib, unda for, while, do-while sikllarining sintaksisi, misollar orqali ishlatilishi, ular orasidagi farqlar va qo'llanilishi to'g'risida batafsil ma'lumotlar keltirilgan. Bundan tashqari, maqolada takrorlanish operatorlarining dastur samaradorligiga qanday ta'sir qilishi, turli holatlarda qaysi operatorni tanlash muhimligi haqida so'z boradi. Shu bilan birga, takrorlanish operatorlarining ishlashida yuzaga keladigan muammolar va ularni yechish usullari ham ko'rsatilgan.

Аннотация: Данная статья посвящена операторам цикла в языке программирования C++, включая их виды, синтаксис, механизм работы и применение в реальных задачах программирования. Операторы цикла в C++ позволяют программисту эффективно управлять потоком выполнения программы, избегая многократного повторения одного и того же кода.

Статья ориентирована в основном на начинающих и программистов среднего уровня. В ней подробно рассматриваются синтаксис и примеры использования циклов for, while, do-while, их различия и области применения.





Кроме того, в статье обсуждается влияние операторов цикла на производительность программ, а также важность выбора подходящего оператора в различных ситуациях. Помимо этого, рассматриваются возможные проблемы, возникающие при работе с циклами, и предлагаются методы их решения.

Abstract: This article is dedicated to loop operators in the C++ programming language, covering their types, syntax, working mechanism, and applications in real programming tasks. Loop operators in C++ allow programmers to efficiently control the program flow without repeating the same code multiple times.

The article is mainly intended for beginner and intermediate-level programmers. It provides a detailed explanation of the syntax and usage of for, while, and do-while loops, highlighting their differences and applications through examples.

Additionally, the article discusses how loop operators impact program performance and the importance of selecting the appropriate loop in different scenarios. Furthermore, it addresses common issues that may arise when working with loops and presents methods to resolve them.

Kalit so‘zlar: C++, boshqaruv operatorlari, takrorlanish operatori, sikl, for sikli, while sikli, do-while sikli, dasturlash, algoritm, takrorlanish, kodni qisqartirish.

Ключевые слова: C++, операторы управления, оператор цикла, цикл, цикл for, цикл while, цикл do-while, программирование, алгоритм, повторение, сокращение кода.

Keywords: C++, control operators, loop operator, loop, for loop, while loop, do-while loop, programming, algorithm, repetition, code optimization.

Kirish

C++ dasturlash tili, o‘zining keng imkoniyatlari va samarali boshqaruv strukturasi bilan dasturchilarga murakkab masalalarni yechishda katta yordam beradi. Dastur tuzishda eng ko‘p ishlatiladigan operatorlar — boshqaruv operatorlaridir. Bu operatorlar yordamida dastur oqimini boshqarish mumkin. Takrorlanish operatorlari (yoki sikllar) dasturdagi biror amalni ma’lum shartlar asosida bir necha bor bajarishga yordam beradi. C++ dasturlash tilidagi for, while, va do-while sikllari yordamida takrorlanadigan kod qismlari soddalashtiriladi, dastur samaradorligi oshadi va xatoliklar ehtimoli kamayadi.

Maqolaning maqsadi takrorlanish operatorlarini o‘rganishni boshlayotgan dasturchilarga ularni samarali ishlatish uchun zarur bo‘lgan barcha tushunchalar va misollarni taqdim etishdir. Shuningdek, har bir operatorning afzalliklari va





kamchiliklarini muhokama qilish, eng yaxshi amaliyotlarni ko'rsatish ham maqolaning asosiy vazifalaridan biridir.

Takrorlanish Operatorlari Tushunchasi

Takrorlanish operatorlari — bu ma'lum bir amalni bir necha marta bajarish uchun ishlatiladigan boshqaruv operatorlaridir. Ular asosan quyidagi holatlarda qo'llaniladi:

Dasturda bir xil amalni bir nechta ma'lumotlar ustida bajarish (masalan, massivdagi barcha elementlarni qayta ishlash).

Ma'lum bir shart bajarilguncha yoki ma'lum marta amalni takrorlash.

Takrorlanish operatorlari asosan uch xil turda bo'ladi:

for sikli

while sikli

do-while sikli

Har bir sikl turi o'ziga xos ishlash prinsipi va sintaksisiga ega.

For sikli

```
for (int i = 0; i < 5; i++) {  
    cout << "Salom, dunyo!" << endl;}
```

For sikli — takrorlanish soni ma'lum bo'lganda ishlatiladi. Sintaksisda 3 ta parametr mavjud:

Boshlang'ich qiymat ($i = 0$)

Shart ($i < 5$)

Yangi qiymat ($i++$)

Bu sikl har bir iteratsiyada i ni oshirib boradi va shart bajarilgunga qadar kodni takrorlaydi.

Afzalliklari: Takrorlanish soni aniq bo'lganda ishlatiladi va kodekni o'qish juda oson.

Kamchiliklari: Takrorlanish soni noma'lum bo'lsa, foydalanuvchi i qiymatini yangilashni unutishi mumkin.

While Sikli

```
int i = 0;
```

```
while (i < 5) {
```

```
    cout << "Salom, C++!" << endl; i++; }
```

While sikli shartni tekshirganidan keyin kodni bajaradi. Agar shart to'g'ri bo'lsa, sikl takrorlanadi.

Afzalliklari: Shartni aniqlash juda oddiy va shart bajarilgunga qadar kodni qayta ishlaydi.



Kamchiliklari: Shartni noto‘g‘ri belgilash yoki yangilamagan holda siklni cheksiz takrorlash mumkin.

Do-while Sikli

```
int i = 0;  
do {cout << "Takrorlash do-while bilan!" << endl;  
i++;} while (i < 5);
```

Do-while sikli avval kodni bajaradi, keyin shartni tekshiradi. Bu sikl kamida bitta marta bajariladi.

Afzalliklari: Kod kamida bir marta bajariladi.

Kamchiliklari: Shart tekshirilmaydigan holatlarda noto‘g‘ri ishlashi mumkin.

Break va Continue Operatorlari

Takrorlanish sikllarida ba‘zan siklni to‘xtatish yoki keyingi iteratsiyaga o‘tish zarur bo‘ladi. Buning uchun break va continue operatorlari ishlatiladi.

break — siklni to‘xtatadi.

continue — siklning joriy iteratsiyasini o‘tkazib yuboradi va keyingisiga o‘tadi.

```
for (int i = 0; i < 10; i++) {  
    if (i == 5) break; // Siklni to'xtatadi  
    if (i % 2 == 0) continue; // Toq sonlarni o'tkazib yuboradi  
    cout << i << " toq son." << endl;}
```

Xulosa



C++ dasturlash tilidagi takrorlanish operatorlari dasturlashning asosiy vositalaridan biridir, chunki ular bir xil amallarni ko‘p marta bajarishga imkon beradi. Takrorlanish — bu ko‘plab dasturlarda tez-tez uchraydigan vazifa, masalan, ma’lumotlarni qayta ishlash, massivlarni aylanib chiqish, foydalanuvchi kiritmalarini qayta ishlash, va boshqa ko‘plab operatsiyalar. Takrorlanish operatorlari yordamida dasturchilar kodni soddalashtiradi, uning samaradorligini oshiradi va yozilish vaqtini qisqartiradi.

C++ tilidagi for, while, va do-while sikllari har biri o‘ziga xos ishlash tamoyiliga ega. For sikli, odatda, takrorlanish soni oldindan ma’lum bo‘lganda eng samarali bo‘ladi. Bu sikl o‘zining aniq va qisqa sintaksisi bilan dasturchilarga takrorlanishning aniq sonini belgilash imkonini beradi. Shuningdek, while sikli shartni tekshirganidan keyin kodni bajaradi, shuning uchun u shartni bajarilishiga qarab takrorlanadi, bu esa uning ishlashini dinamik qiladi. Do-while sikli esa kamida bitta takrorlashni ta’minlaydi, chunki u avval kodni bajaradi va keyin shartni tekshiradi. Bu xususiyat, agar dastur kamida bir marta bajarilishi kerak bo‘lgan amallarni bajarishi kerak bo‘lsa, juda foydalidir.





Bundan tashqari, break va continue operatorlari yordamida siklning ishlashini yanada boshqarish mumkin. Break operatori siklni to'xtatadi, bu esa ayniqsa, sikl ichida shartni tekshirish kerak bo'lganda yoki muayyan bir qiymatga yetganda foydalidir. Continue operatori esa joriy iteratsiyani o'tkazib yuboradi va siklning keyingi iteratsiyasiga o'tishga imkon beradi. Bu operatorlar dasturchiga siklni yanada moslashuvchan va samarali qilish imkonini beradi.

Foydalanilgan adabiyotlar

1. Stroustrup B. – *The C++ Programming Language*, Addison-Wesley, 2013
2. Deitel & Deitel – *C++ How to Program*, Pearson Education, 2017
3. Malik D. S. – *C++ Programming: From Problem Analysis to Program Design*, Course Technology, 2018
4. Cplusplus.com – <https://cplusplus.com/doc/tutorial/control/>
5. GeeksforGeeks – <https://www.geeksforgeeks.org/loops-in-cpp/>