

**C++ DASTURLASH TILIDA VORISLIKDAN FOYDALANISH**

**ИСПОЛЬЗОВАНИЕ НАСЛЕДОВАНИЯ В ЯЗЫКЕ  
ПРОГРАММИРОВАНИЯ C++**

**USING INHERITANCE IN THE C++ PROGRAMMING LANGUAGE**

**Abdullayev Shaxboz Solijon o‘g‘li**

*Farg‘ona davlat universiteti Axborot texnologiyalari kafedrasida katta o‘qituvchisi*

[shaxbozfardu2023@gmail.com](mailto:shaxbozfardu2023@gmail.com)

**Komilov Islombek Nosirjon o‘g‘li**

*Farg‘ona davlat universiteti Axborot tizimlari va texnologiyalar yo‘nalishi*

*1-kurs talabasi*

[komilovislombek07.06@gmail.com](mailto:komilovislombek07.06@gmail.com)

**Annotatsiya:** *Ushbu maqola C++ dasturlash tilida vorislik konsepsiyasiga oid asosiy tushunchalar va uning dasturlashda qanday qo‘llanilishini ko‘rib chiqadi. Maqolada vorislikning afzalliklari, u qanday ishlashini va C++ tilida vorislikni qanday amalga oshirish mumkinligi batafsil yoritiladi. Shuningdek, vorislikdan foydalanish dastur tuzish jarayonini qanday soddalashtirishi, kodning qayta ishlanishini qanday ta‘minlashi va dasturlashning modularligini oshirishi ko‘rsatilgan.*

**Аннотация:** *В данной статье рассматриваются основные понятия концепции наследования в языке программирования C++ и способы ее использования в программировании. В статье подробно описаны преимущества наследования, как оно работает и как реализовать наследование в C++. Также показано, как использование наследования может упростить процесс программирования, обеспечить возможность повторного использования кода и повысить модульность программирования.*

**Abstract:** *This article explores the fundamental concepts of inheritance in the C++ programming language and its application in software development. It provides a detailed explanation of the benefits of inheritance, how it works, and how it can be implemented in C++. Additionally, the article highlights how inheritance simplifies*

*the software development process, enhances code reusability, and improves modularity in programming.*

**Kalit so‘zlar:** *C++ dasturlash, OOP (Obyektga yo‘naltirilgan dasturlash), vorizlik, inson-ota sinflari, polimorfizm, abstraksiya, obyektga yo‘naltirilgan dasturlash prinsiplari.*

**Ключевые слова:** *C++ программирование, ООП (Объектно-ориентированное программирование), наследование, родительские и дочерние классы, полиморфизм, абстракция, принципы объектно-ориентированного программирования.*

**Keywords:** *C++ programming, OOP (Object-Oriented Programming), inheritance, parent-child classes, polymorphism, abstraction, object-oriented programming principles.*

## **Kirish**

C++ dasturlash tili obyektga yo‘naltirilgan dasturlash (OOP) paradigmalaridan foydalanadi. OOP dasturlashda asosiy tushunchalardan biri — vorislikdir. Vorizlik, bir sinfning boshqa sinfdan xususiyat va metodlarni meros qilib olish imkonini beradi. Bu dasturiy ta‘minot yaratish jarayonida kodning qayta ishlanishini, oson tahlilini va rivojlanishini ta‘minlaydi. C++ tilida vorislik sinflar o‘rtasida bog‘lanishni o‘rnatish va kodni qayta ishlashda samarali usul hisoblanadi.

## **Vorislikning tushunchasi**

Vorislik — bu bir sinfning boshqa sinfdan xususiyatlarni (atributlarni) va metodlarni (funksiyalarni) meros qilib olish jarayonidir. C++ dasturlash tilida bu konsepsiya sinflar o‘rtasidagi bog‘lanishni o‘rnatishga yordam beradi. Vorislik ikkita asosiy turga bo‘linadi: oddiy vorislik va ko‘p martalik vorislik.

*Oddiy vorislik:* Bir sinf faqat bitta sinfdan meros oladi.

*Ko‘p martalik vorislik:* Bir sinf bir nechta sinfdan meros olishi mumkin.

## **Vorizlikni amalga oshirish**

C++ tilida vorislikni amalga oshirish uchun *public*, *protected*, yoki *private* akses modifikatorlarini ishlatish mumkin. Bular yordamida sinfning atributlari va metodlari boshqalar tomonidan qanday foydalanishiga qarab o‘rnatiladi.

Misol:

```
#include <iostream>
```

```

using namespace std;
// Ota sinfi
class Animal {
public:
void eat() {
cout << "Animal is eating." << endl; };
// Voris sinfi
class Dog : public Animal {
public:
void bark() {
cout << "Dog is barking." << endl; };
int main() {Dog dog;
dog.eat(); // Animal sinfidan meros olindi
dog.bark(); // Dog sinfining o‘z metodini chaqirish
return 0;}

```

Yuqoridagi misolda, Dog sinfi Animal sinfidagi *eat* metodini meros qilib olgan va o‘z metodini *bark* deb qo‘shgan.

### **Vorizlikning afzalliklari**

*Vorizlikning asosiy afzalliklari:*

Kodning qayta ishlanishi: Bir marta yozilgan kodni boshqa sinflarda qayta ishlatish imkoniyati mavjud.

Kengaytirilgan dasturlar: Yangi sinflarni avvalgi kodni qayta yozmasdan yaratish imkoniyatini beradi.

Polimorfizm: Vorizlik orqali polimorfizmni amalga oshirish mumkin, bu esa metodlarning turli variantlarini ishlatishga imkon beradi.

### **Polimorfizm va vorislik**

Polimorfizm — bu bir metodni bir nechta shaklda ishlatish imkoniyatini ta'minlaydi. Vorizlik va polimorfizmni birga ishlatish orqali dastur murakkabligini kamaytirish va tizimni yanada moslashuvchan qilish mumkin.

Misol:

```

#include <iostream>
using namespace std;
class Animal {public:

```

```
virtual void sound() {  
    cout << "Animal sound" << endl;};  
class Dog : public Animal {  
public:  
    void sound() override {  
        cout << "Bark" << endl;};  
int main() {  
    Animal *animal = new Dog();  
    animal->sound(); // Dog sinfidagi sound() chaqiriladi  
    delete animal;  
    return 0;}
```

Bu misolda, *sound()* metodining turli shakllari mavjud bo‘lib, objektingizning haqiqiy tipiga qarab chaqiriladi.

### **Xulosa**

Vorislik C++ dasturlash tilida obyektga yo‘naltirilgan dasturlash paradigmalarning muhim qismi bo‘lib, u dasturchilarga kodni qayta ishlash va dasturlarni samarali tashkil qilish imkonini beradi. Bu tamoyil yordamida dasturlash tizimi yanada modulli va kengaytirilgan bo‘ladi. C++ dasturlash tilidagi vorislik imkoniyatlarini o‘rganish, dastur tuzishda samaradorlikni oshiradi va yangi funksiyalarni qo‘shishda asqotadi.

### **Foydalanilgan adabiyotlar:**

1. Bjarne Stroustrup – The C++ Programming Language (4th Edition). Addison-Wesley, 2013.
2. Herbert Schildt – C++: The Complete Reference (4th Edition). McGraw-Hill Education, 2003.
3. Stanley B. Lippman, Jose L. Lajoie, Barbara E. Moo – C++ Primer (5th Edition). Addison-Wesley, 2012.
4. Andrew Koenig, Barbara E. Moo – Accelerated C++: Practical Programming by Example. Addison-Wesley, 2000.
5. Mukul K. Tiwari – Object-Oriented Programming with C++. Dreamtech Press, 2005.