C++ DASTURLASH TILIDA INTERFEYSLARDAN FOYDALANISH

ИСПОЛЬЗОВАНИЕ ИНТЕРФЕЙСОВ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++

USING INTERFACES IN C++ PROGRAMMING LANGUAGE

Shermatova Xilola Mirzayevna
Farg‘ona davlat universiteti Axborot texnologiyalari kafedrasi dotsenti
shermatovahilola1978@gmail.com
Ismoilova Gulhayot Rustamjon qizi
Farg‘ona davlat universiteti Axborot tizimlari va texnologiyalari yo‘nalishi 1-kurs talabasi
gulhayotismoilova@gmail.com

Annotatsiya: *Ushbu maqola C++ dasturlash tilida interfeyslardan foydalanishning ahamiyatini va uning dasturlashdagi o'rnini ko'rib chiqadi. Interfeyslardan foydalanish orqali dasturchilar kodning qayta ishlanishini, kengaytirilishini va oson saqlanishini ta'minlashlari mumkin. Maqolada interfeyslardan qanday qilib abstrakt sinflar yaratish, polimorfizm va modulni saqlash kabi obyektga yo'naltirilgan dasturlash tamoyillariga qanday erishish mumkinligi haqida misollar keltiriladi. Shuningdek, interfeyslardan foydalanishning afzalliklari, jumladan kodning umumiyligi va kengaytirilganligini qanday oshirishi haqida ham batafsil ma'lumot beriladi. Ushbu maqola interfeyslardan foydalanishni o'rganuvchilar va C++ni chuqurroq o'rganmoqchi bo'lgan dasturchilar uchun foydali manba hisoblanadi.*

Kalit so‘zlar: *C++, interfeyslardan foydalanish, oyektga yo'naltirilgan dasturlash (OOP), abstrakt sinflar, polimorfizm, dasturlashda modularlik, kengaytirilgan dasturlash, pure virtual functions (To'liq virtual funksiyalar), sinflar va interfeyslardan foydalanish*

Аннотация: *В этой статье рассматривается важность использования интерфейсов в языке программирования C++ и их роль в программировании. Используя интерфейсы, разработчики могут обеспечить повторное использование, расширяемость и легкость в обслуживании своего кода. В статье приведены примеры создания абстрактных классов с использованием интерфейсов, реализации полиморфизма и сохранения модульности в соответствии с принципами объектно-ориентированного программирования.*

Также рассматриваются преимущества использования интерфейсов, такие как повышение общей гибкости и расширяемости кода. Эта статья является полезным ресурсом для тех, кто учится работать с интерфейсами, а также для разработчиков, стремящихся углубить свои знания C++.

Ключевые слова: *C++, использование интерфейсов, объектно-ориентированное программирование (ООП), абстрактные классы, полиморфизм, модульность в программировании, расширяемое программирование, чистые виртуальные функции, использование классов и интерфейсов.*

Annotation: *This article discusses the importance of using interfaces in the C++ programming language and its role in programming. By utilizing interfaces, developers can ensure that their code is reusable, extensible, and easy to maintain. The article provides examples of how to create abstract classes using interfaces, implement polymorphism, and maintain modularity, all in line with object-oriented programming principles. Furthermore, it explains the advantages of using interfaces, such as enhancing code generality and extensibility. This article serves as a useful resource for those learning to use interfaces and for developers wishing to deepen their understanding of C++.*

Keywords: *C++, interface usage, object-oriented programming (OOP), abstract classes, polymorphism, modularity in programming, extensible programming, pure virtual functions, classes and interfaces usage.*

Kirish

C++ dasturlash tili o‘zining kuchli imkoniyatlari bilan tanilgan. Uning asosiy xususiyatlaridan biri bu ob'ektga yo‘naltirilgan dasturlash (OOP) paradigmalarini qo‘llab-quvvatlashidir. OOP paradigmasining ba'zi muhim tushunchalari — inkapsulyatsiya, meros olish, polimorfizm va abstraktsiya — dasturlashni yanada samarali va tartibli qilishga yordam beradi. C++ tilida interfeyslardan foydalanish aynan shu imkoniyatlarni kengaytirishda muhim rol o‘ynaydi.

Interfeys Nima?

Interfeys — bu faqat metodlar deklaratsiyalarini o‘z ichiga olgan, ammo hech qanday konkret implementatsiyani (yoki kodni) bermaydigan abstrakt sinfdir. Interfeys yordamida biz sinflarga bir xil xususiyatlar va metodlarni taqdim etamiz, lekin ularning implementatsiyasini har bir sinf o‘ziga xos tarzda belgilaydi. Interfeyslar orqali biz sinflar o‘rtasida umumiy shartnomani (contract) belgilaymiz.

C++ tilida interfeyslarni yaratish uchun **abstrakt sinflar** (abstract classes)dan foydalanamiz. Abstrakt sinfda kamida birta **pure virtual function** (to‘liq virtual metod) bo‘lishi kerak. Bu metodlar sinf tomonidan amalga oshirilishi kerak bo‘ladi.

Interfeyslarni Yaratish

C++da interfeysni yaratish uchun abstrakt sinfdan foydalaniladi. Abstrakt sinfda metodlar faqat deklaratsiyalari bilan bo‘ladi va ularni amalga oshirish majburiy bo‘ladi.

1. Interfeys yaratish

Interfeýsi yaratish uchun, faqat metodlarni e'lon qilamiz. Bu metodlar shakllarni chizish va ularning yuzasini hisoblash uchun bo'ladi.

```
#include <iostream>
class IShape {
public:
    virtual void draw() = 0;
    virtual double area() = 0;  };
```

2. Shakllar sinflari

Endi, IShape interfeýsi yordamida Circle (doira) va Square (kvadrat) sinflarini yaratamiz. Har bir sinf o'zining metodlarini amalga oshiradi.

```
class Circle : public IShape {
private:
double radius;
public:
Circle(double r) : radius(r)
void draw() override {
std::cout << "Doira chizilmoqda." << std::endl;}
double area() override {
return 3.14 * radius * radius; }};
class Square : public IShape {
private:
double side;
public:
Square(double s) : side(s) {}
void draw() override {
std::cout << "Kvadrat chizilmoqda." << std::endl;}
double area() override {
return side * side; }};
```

Asosiy dastur

Endi, main funksiyasida interfeýsni ishlatib, har bir shaklni yaratamiz va ularni ishlatib ko'ramiz. Bu yerda biz polimorfizmni ishlatamiz, ya'ni turli shakllar bir xil interfeýs orqali boshqariladi.

```
int main() {IShape* shape1 = new Circle(5.0);
IShape* shape2 = new Square(4.0);
shape1->draw();
std::cout << "Doira yuzi: " << shape1->area() << std::endl;
shape2->draw();
std::cout << "Kvadrat yuzi: " << shape2->area() << std::endl;
delete shape1;
delete shape2;
return 0;
```

Dastur qanday ishlaydi

IShape interfeýsi ikkita metodni e'lon qiladi: draw() (chizish) va area() (yuza hisoblash).

Circle va Square sinflari bu interfeýsi implementatsiya qiladi va har bir metodni o'ziga xos tarzda amalga oshiradi.

main funksiyasida, shape1 va shape2 ob'ektlari yaratiladi va ular IShape turidan bo'ladi. Har bir shakl o'zining metodlarini bajaradi, masalan, draw() va area().

Dastur oxirida, dinamik xotiradan ajratilgan ob'ektlar delete orqali tozalanadi.

Xulosa: C++ dasturlash tilida interfeyslardan foydalanish dasturlarni yanada moslashuvchan, kengaytiriladigan va saqlashda oson qilishga yordam beradi. Interfeyslar va polimorfizm OOPning muhim tarkibiy qismlaridan biridir. Dasturlashda interfeyslardan to'g'ri foydalanish, kodning takrorlanuvchanligini kamaytiradi va yangi sinflarni joriy qilishda qulaylik yaratadi.Shu tariqa, interfeyslar yordamida C++ dasturlarini samarali tashkil etish mumkin, bu esa dasturiy ta'minotning rivojlanishiga katta yordam beradi.

FOYDALANILGAN ADABIYOTLAR:

1. Bjarne Stroustrup. Programming: Principles and Practice Using C++ (2nd Edition). Person Education, Inc. 2014. second printing, January 2015.
2. Harry Hariom Choudhary, Bjarne M Stroustrup. C++ Programming Professional.: Sixth Best Selling Edition for Beginner's & Expert's 2014.
3. Bjarne Stroustrup.The C++ Programming Language, 4th Edition. Person Education, Inc. 2013. Third printing, April 2014.
4. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. C va C++ tili. “Voris-nashriyot” MCHJ, Toshkent 2013. 488 b.
5. Horstmann, Cay S. C++ for everyone / Cay S. Horstmann. Printed in the United States of America - 2nd ed. 2010. – P. 562. 579
6. Horton I. - Beginning Visual C++ 2012 / I.Horton. Published simultaneously in Canada. – 2012. –P. 988.