



TANQIDIY NAZAR, TAHLILIY TAFAKKUR VA INNOVATION G'UYALAR



C++ DA SINIF VA OBYEKTAR BILAN ISHLASH

РАБОТА С КЛАССАМИ И ОБЪЕКТАМИ В C++

WORKING WITH CLASSES AND OBJECTS IN C++

Shermatova Xilola Mirzayevna

FarDU Axborot texnologiyalari kafedrası dotsenti

shermatovahilola1978@gmail.com

<https://orcid.org/0000-0001-5014-9549>

998904099095

Sharoffidinov Shahobiddin Norinboy o'g'li

FarDU Axborot tizimlari va texnologiyalari yo'nalishi 1-kurs talabasi

sharofiddinovshahobiddin062@gmail.com

Annotatsiya: C++ dasturlash tilida sinflar va obyektlar bilan ishlashning asosiy tamoyillari va amaliy jihatlari ko'rib chiqiladi. Maqolada, sinf yaratish, uning a'zolariga murojaat qilish, konstruktorlar va destruktorlar ishlatish, hamda obyektlar bilan ishlashning o'ziga xos xususiyatlari yoritiladi. Shuningdek, polimorfizm, enkapsulyatsiya va meros kabi ob'jektga yo'naltirilgan dasturlashning asosiy tushunchalari ham ko'rib chiqiladi. Ushbu maqola C++ dasturlash tilida ob'jektga yo'naltirilgan dasturlashni o'rganishni boshlagan dasturchilar uchun foydali qo'llanma bo'lishi mumkin. Dasturlashdagi sinf va obyektlar tushunchasini yaxshiroq tushunish, dasturchilar uchun yuqori darajada samarali va keng qamrovli dasturlar yaratish imkoniyatlarini yaratadi. Shuningdek, ushbu maqola, sinflar yordamida kodning modulini oshirish, xatoliklarni kamaytirish va kodni yanada optimallashtirishni ta'minlaydi. Polimorfizm, enkapsulyatsiya va meros kabi tushunchalar orqali dasturlashda yuqori darajadagi moslashuvchanlik va kengaytirilgan imkoniyatlar yaratish mumkin.

Аннотация: В данной статье рассматриваются основные принципы и практические аспекты работы с классами и объектами в языке программирования C++. В статье рассматриваются создание классов, доступ к их членам, использование конструкторов и деструкторов, а также особенности работы с объектами. Также затрагиваются такие ключевые концепции объектно-ориентированного программирования, как полиморфизм, инкапсуляция и наследование. Этот материал может быть полезен начинающим разработчикам, изучающим объектно-ориентированное



программирование на C++. Понимание работы с классами и объектами помогает создавать более эффективные и масштабируемые программы, повышая уровень модульности кода, уменьшая количество ошибок и улучшая производительность программ. Использование таких концепций, как полиморфизм, инкапсуляция и наследование, открывает новые возможности для создания гибких и расширяемых решений.

Abstract: *This article discusses the fundamental principles and practical aspects of working with classes and objects in C++ programming language. The article covers class creation, accessing class members, using constructors and destructors, and the unique characteristics of working with objects. It also addresses key concepts of object-oriented programming, such as polymorphism, encapsulation, and inheritance. This article can serve as a useful guide for beginner developers learning object-oriented programming in C++. A better understanding of classes and objects enables developers to create more efficient and scalable applications, improving the modularity of code, reducing errors, and enhancing program performance. Concepts such as polymorphism, encapsulation, and inheritance provide flexibility and expandability, allowing for more adaptable and maintainable code in complex systems.*

Kalit soʻzlar: *C++, sinf, obyekt, konstruktor, destruktor, polimorfizm, enkapsulyatsiya, meros, ob'ektga yo'naltirilgan dasturlash, dasturlash tamoyillari, dastur yaratish, kod modulligi.*

Ключевые слова: *C++, класс, объект, конструктор, деструктор, полиморфизм, инкапсуляция, наследование, объектно-ориентированное программирование, принципы программирования, создание программ, модульность кода.*

Keywords: *C++, class, object, constructor, destructor, polymorphism, encapsulation, inheritance, object-oriented programming, programming principles, software development, code.*

KIRISH

C++ dasturlash tili, ob'ektga yo'naltirilgan dasturlash (OOP) paradigmalaridan foydalangan holda kuchli va samarali dasturlar yaratish imkonini beradi. Sinf va obyektlar bu paradigmaning asosiy tarkibiy qismlari bo'lib, dasturlashda yuqori darajadagi modullik, qayta foydalanish va kengaytirilgan imkoniyatlarni ta'minlaydi. Sinflar, ma'lumotlar va ularga tegishli amallarni bir joyda birlashtirgan holda dasturdagi kodni yanada tartibga solish imkonini beradi. Obyektlar esa sinfdan yaratilgan konkret nusxalar bo'lib, ular sinfnings barcha atributlari va metodlarini o'z



ichiga oladi. Sinflar yordamida kodning o'qilishi osonlashadi va dasturchilar orasida hamkorlikni soddalashtiradi. Obyektga yo'naltirilgan dasturlash tushunchalari, shuningdek, dasturlarni kengaytirish va saqlashda yuqori darajada moslashuvchanlikni ta'minlaydi. Bu esa dasturchilarga mavjud kodni qayta ishlatish va uni yangi shartlarga moslashtirish imkoniyatlarini beradi. Obyektga yo'naltirilgan dasturlashni to'liq o'zlashtirish dasturchilarga ilg'or dastur yaratish imkoniyatlarini taqdim etadi.

C++ dasturlash tili nafaqat yuqori samarali dasturlar yaratish imkonini beradi, balki uning kuchli xususiyatlari orqali kompleks tizimlarni modulli tarzda tashkil etish, ularga o'zgarishlar kiritish va boshqarish jarayonlarini soddalashtirish mumkin. Sinflar va obyektlar bilan ishlashda dasturchilar o'zlarining dasturlarini ancha intuitiv va moslashuvchan tarzda qurishlari mumkin. Misol uchun, polimorfizm va enkapsulyatsiya kabi tushunchalar, dasturchilarga o'zgaruvchilarni xavfsiz boshqarish va ularning qiymatlarini tashqi ta'sirlardan himoya qilish imkonini beradi.

Bundan tashqari, C++ dasturlash tili keng qo'llaniladigan va ko'plab sohalarda ishlatiladigan tildir. U tizim dasturlari, o'yinlar, tarmoq dasturlari va ma'lumotlar bazalari bilan ishlashda juda samarali hisoblanadi. Ushbu maqola C++ tilida sinflar va obyektlar bilan ishlashning asosiy tamoyillari va ob'ektga yo'naltirilgan dasturlashning afzalliklarini batafsil ko'rib chiqadi. Maqola, yangi boshlovchilar va C++ tilini chuqur o'rganayotgan dasturchilar uchun foydali bo'lishi mumkin. Obyektga yo'naltirilgan dasturlash konseptlarini o'zlashtirish, dasturlashni ancha samarali va professional darajada amalga oshirish imkonini beradi.

Dasturning kodi :

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
class Talaba {
```

```
private:
```

```
    string ism;
```

```
    int yosh;
```

```
    double ball;
```

```
public:
```

```
    Talaba() {
```

```
        ism = "";
```

```
        yosh = 0;
```

```
        ball = 0.0; }
```

```
void malumotlarniKirit(string i, int y, double b) {ism = i;
```



```
yosh = y;  
ball = b; }  
void malumotlarniChiqar() {  
    cout << "Talaba ismi: " << ism << endl;  
    cout << "Yoshi: " << yosh << endl;  
    cout << "Ballari: " << ball << endl;}};  
int main() {  
    Talaba talaba1;  
    string ism;  
    int yosh;  
    double ball;  
    cout << "Talaba ismini kiriting: ";  
    getline(cin, ism);  
    cout << "Yoshini kiriting: ";  
    cin >> yosh;  
    cout << "Ballarini kiriting: ";  
    cin >> ball;  
    talaba1.malumotlarniKirit(ism, yosh, ball);  
    cout << "Talaba ma'lumotlari:";  
    talaba1.malumotlarniChiqar();  
    return 0;}
```

Dastur tahlili

#include <iostream> - Bu kutubxona fayli konsolga ma'lumot chiqarish (cout) va konsoldan ma'lumot olish (cin) imkoniyatlarini beradi. Bu satr dasturda ishlatilgan barcha konsol bilan aloqador funktsiyalarni chaqirishi uchun zarur.

#include <string> - Bu kutubxona fayli string turini ishlatishga imkon beradi. string o'zgaruvchisi matnli ma'lumotlarni saqlash uchun ishlatiladi.

using namespace std; - Bu satr std nomli kutubxonani dasturga kiritadi. std namespace ichidagi barcha elementlardan bevosita foydalanish imkonini beradi. Masalan, cout, cin, string kabi elementlar to'g'ridan-to'g'ri chaqiriladi va "std::" prefiksini qo'llash kerak emas.

class Talaba - Bu satr yangi Talaba nomli sinfni (klassni) yaratishni boshlaydi. Bu sinf talaba haqidagi ma'lumotlarni saqlash va ularni chiqarish uchun ishlatiladi.

Private - Bu yerda private bo'limi sinf ichidagi a'zolari faqat sinf ichidan kirishga ruxsat beradi. private a'zolar tashqaridan (dasturning boshqa qismlaridan) bevosita kirish mumkin emas.



string ism;, int yosh;, double ball; - Bu satrlar sinfdagi Talaba ob'ekti uchun kerakli o'zgaruvchilarni e'lon qiladi:

public - Bu yerda public bo'limi sinfdagi a'zolari tashqi dastur kodidan ham ishlatishga imkon beradi. public a'zolari sinf ob'ekti orqali chaqirish mumkin.

Talaba() { ... } - Bu yerda Talaba sinfining konstruktorini ko'rsatyapsiz. Konstruktor sinf ob'ekti yaratilganda avtomatik ravishda chaqiriladi va sinfdagi o'zgaruvchilarga boshlang'ich qiymatlar beradi:

ism	=	""	—	boshlang'ichda	ism	bo'sh	bo'ladi.
yosh	=	0	—	boshlang'ichda	yosh	0	bo'ladi.
ball = 0.0 — boshlang'ichda ball 0.0 bo'ladi.							

void malumotlarniKirit(string i, int y, double b) { ... } - Bu funksiya talaba ma'lumotlarini kiritish uchun ishlatiladi. U uchta parametr oladi (i — ism, y — yosh, b — ball), va ularni sinfdagi mos o'zgaruvchilarga tayinlaydi:

void malumotlarniChiqar() { ... } - Bu funksiya talaba ma'lumotlarini ekranga chiqaradi. cout yordamida talabaning ismi, yoshi va ballari chiqariladi:

int main() { ... } - Dastur bajarilishi shu yerda boshlanadi. main() funksiyasi — dasturda asosiy funksiya bo'lib, barcha kod bu funksiyada bajariladi.

Talaba talaba1; - Bu satr Talaba sinfidagi bir ob'ekt yaratadi, bu ob'ektga talaba1 nomi berilgan. Bu ob'ekt orqali sinfnining metodlariga (funksiyalariga) murojaat qilish mumkin bo'ladi.

string ism;, int yosh;, double ball; - Bu yerda lokal o'zgaruvchilar e'lon qilinadi, ular foydalanuvchidan talaba ma'lumotlarini olish uchun ishlatiladi.

cout << "Talaba ismini kiriting: "; - Konsolga foydalanuvchidan talabaning ismini kiritishni so'raydi.

getline(cin, ism); - Foydalanuvchidan ismni olish uchun ishlatiladi. getline funksiyasi butun qatorni o'qib beradi, shuning uchun foydalanuvchi bo'sh joylar bilan ism kiritsa ham to'liq o'qiladi.

cout << "Yoshini kiriting: "; - Foydalanuvchidan talabaning yoshini kiritishni so'raydi.

cin >> yosh; - Foydalanuvchidan yoshni olish uchun ishlatiladi. Bu yerda integer turidagi ma'lumot olinadi.

cout << "Ballarini kiriting: "; - Foydalanuvchidan talabaning ballarini kiritishni so'raydi.

cin >> ball; - Foydalanuvchidan ballarini olish uchun ishlatiladi. Bu yerda o'nlik son (double) turi olinadi.



talaba1.malumotlarniKirit(ism, yosh, ball); - Bu satr malumotlarniKirit metodini chaqiradi va foydalanuvchidan olingan ma'lumotlarni (ism, yosh, ball) talaba1 ob'ektiga o'tkazadi.

cout << "Talaba ma'lumotlari:"; - Bu satr ekranga "Talaba ma'lumotlari:" yozuvini chiqaradi.

talaba1.malumotlarniChiqar(); - Bu satr malumotlarniChiqar metodini chaqiradi va talaba1 ob'ekti orqali talaba ma'lumotlarini ekranga chiqaradi.

return 0; - return 0 satri dasturning to'liq yakunlangani va xatolik yo'qligini bildiradi.

XULOSA

C++ dasturlash tili ob'ektga yo'naltirilgan dasturlash (OOP) paradigmalarini qo'llash orqali dasturlarni modulli, kengaytiriladigan va o'qilishi oson tarzda yaratish imkoniyatini taqdim etadi. Sinflar va obyektlar bu paradigmaning asosiy tarkibiy qismlari bo'lib, dasturchilarga ma'lumotlarni va amallarni bir joyda jamlash imkonini beradi. Maqolada C++ tilida sinf va obyektlar bilan ishlashning asosiy tamoyillari, konstruktorlar, destruktorlar, polimorfizm, enkapsulyatsiya va meros kabi muhim tushunchalar yoritildi. Sinflar yordamida kodning qayta foydalanish imkoniyatlari oshadi, dasturdagi xatoliklar kamayadi va dastur samaradorligi ortadi. Shuningdek, ob'ektga yo'naltirilgan dasturlash konseptlarini o'zlashtirish orqali dasturchilar samarali va yuqori sifatli dasturlar yaratish imkoniyatiga ega bo'lishadi. C++ dasturlash tili yordamida sinf va obyektlar bilan ishlashni o'rganish, zamonaviy dasturlash amaliyotida muhim ko'nikmalarni rivojlantirishga yordam beradi. Bu maqola, dasturlashni o'rganayotgan yoki o'z bilimlarini yanada mustahkamlashni xohlagan dasturchilar uchun foydali qo'llanma bo'lishi mumkin.



FOYDALANILGAN ADABIYOTLAR:

- 1) Stroustrup, B. (2000). *The C++ Programming Language*. Addison-Wesley.
- 2) Deitel, H. M., & Deitel, P. J. (2017). *C++ How to Program*. Pearson.
- 3) Schildt, H. (2019). *C++: The Complete Reference*. McGraw-Hill Education.
- 4) Bjarne Stroustrup. (2014). *The Design and Evolution of C++*. Addison-Wesley Professional.
- 5) Lippman, S. B., Lajoie, J., & Moo, B. E. (2005). *C++ Primer*. Addison-Wesley.
- 6) Kernighan, B. W., & Ritchie, D. M. (1988). *The C Programming Language*. Prentice Hall.
- 7) Meyers, S. (2005). *Effective C++: 55 Specific Ways to Improve Your Programs and Designs*. Addison-Wesley.
- 8) Baldwin, C. (2016). *C++ Programming for Beginners: The Ultimate Beginner's Guide to Learning C++ Programming*. Independently published.