

KOMPONENTA TUSHUNCHASI VA XUSUSIYATLARI

Toshkent Davlat Iqtisodiyot Universiteti
Andijon fakulteti II bosqich AATT 70/23 guruh
talabalari **Tursunov Ziyodillo**

To'ychiyev Bexruz

Ilmiy rahbar: **Ismailov.A.SH**

COMPONENT CONCEPT AND PROPERTIES

Tashkent State University of Economics
Andijan faculty II stage AATT 70/23 group
student **Tursunov Ziyodillo**

Toychiyev Bekhruz

Scientific supervisor: **Ismailov.A.SH**

Abstract: This article provides detailed information about the concept of component in C++ programming language, its features, advantages and disadvantages, as well as the different types of components. Examples of how to create components in C++ and how to use them in program development are given, explaining how they affect the programming process.

Components exist not only in software, but also in physical systems, and their integration contributes to successful project implementation in many ways. We will also look at some examples to better understand the concept of a component.

Annotatsiya: Bu maqolada C++ dasturlash tilida komponenta tushunchasi, uning xususiyatlari, afzalliklari va kamchiliklari, shuningdek, komponentlarning turli xil turlari haqida batafsil ma'lumot berilgan. C++ da komponentlarni qanday yaratish va ularni dasturlarni ishlab chiqishda qanday ishlatish mumkinligi misollari keltirilib, ularning dasturlash jarayoniga qanday ta'sir qilishi tushuntiriladi. Komponentlar nafaqat dasturiy ta'minotlarda balki fizik tizimlarda ham mavjud bo'lib, ularning integratsiyasi ko'p jihatdan muvaffaqiyatli loyiha amalga oshirishga xizmat qiladi. Shuningdek, biz komponenta tushunchasini yaxshiroq tushunish uchun ba'zi misollarni ham ko'rib chiqamiz.

Keywords: Component, C++, Modularity, Reusability, Interface, Services, Software, Architecture.

Kalit so'zlar: Komponenta, C++, Modullik, Qayta ishlatish, Interfeys, Xizmatlar, Dasturiy ta'minot, Arxitektura.

KIRISH

Zamonaviy dasturlashda murakkab dasturlarni yaratish juda tez va samarali bo'lishi kerak. Bu vazifani bajarishda C++ kabi kuchli dasturlash tillari muhim rol o'ynaydi. C++ da dasturni tez va samarali yaratish uchun ishlatiladigan asosiy tushunchalardan biri - komponenta tushunchasi hisoblanadi. Dasturlash dunyosida murakkab dasturlarni yaratish uchun biz mayda-mayda qismlarga bo'lib, ularni birlashtirib foydalanamiz. Bu qismlarga biz "komponenta" deb ataymiz. Komponenta - bu C++ da qayta ishlatiladigan va bir-biri bilan o'zaro bog'liq bo'lgan mustaqil modullar hisoblanadi. Ular ma'lum bir vazifani bajaradi va boshqa dasturlar bilan osongina integratsiyalanadi. Komponentlar dasturlash jarayonini soddalashtiradi, dasturni tezroq ishlab chiqishga imkon beradi va dastur kodini qayta ishlatish samaradorligini oshiradi. Komponentalarni turli xil dasturlash tillarida, shu jumladan C++, JavaScript, Python, Java va boshqalarda qo'llash mumkin. Ular veb-saytlar, mobil ilovlar, stol ilovlari va boshqa turli xil dasturlarni yaratish uchun ishlatilishi mumkin. Ushbu maqolada C++ da komponenta tushunchasini chuqurroq o'rganamiz, uning asosiy xususiyatlarini, afzalliklari va kamchiliklari, shuningdek, komponentlarning turli xil turlari haqida batafsil ma'lumot beramiz.

C++ da Komponenta Ta'rifi

C++ da komponenta - bu qayta ishlatiladigan va mustaqil kod bo'lagi bo'lib, ma'lum bir vazifani bajaradi va boshqa dasturlar bilan integratsiyalanadi. C++ ob'ektga yo'naltirilgan dasturlash tili sifatida, komponentlar odatda sinflar (class) va interfeyslar (interface) yordamida amalga oshiriladi.

Visual C++ dasturlash tilida komponentalar dasturiy ilovalarni loyihalashda muhim elementlardir. Komponentlar - bu dasturiy kod va interfeys qismlarini bir butun sifatida ifodalaydigan obyektlardir. Komponentalar, odatda, UI elementlar (masalan, tugmalar, oynalar, list box'lar), tizim resurslari (masalan, vaqt o'lchovchi, fayl menejeri), va boshqa turli dasturiy elementlarni o'z ichiga oladi.

Komponentalar asosiy dasturiy qismning ba'zi funktsiyalarini soddalashtirish, shuningdek, ularni qayta foydalanishga imkoniyat yaratish maqsadida qo'llaniladi. Ular yordamida dasturiy loyihani modullarga ajratish va kodning qayta ishlatilishini ta'minlash mumkin. Visual C++ dasturiy muhitida komponentalar foydalanuvchi interfeysini loyihalashda va tizim bilan integratsiyada keng qo'llaniladi.

Komponenta Xususiyatlari

- **Modullik:** Komponentlar dasturni kichikroq, mustaqil bo'limlarga ajratadi. Bu kodni tushunishni, saqlashni va yangilashni osonlashtiradi.

- **Qayta ishlatish:** Komponentlar turli xil dasturlarda qayta ishlatilishi mumkin, bu esa ishlab chiqish vaqtini va xarajatlarni kamaytiradi.
- **Interfeys:** Har bir komponenta aniq interfeysga ega bo'lib, bu boshqa komponentlarga uning xizmatlaridan qanday foydalanishni ko'rsatib beradi.
- **Xizmatlar:** Komponentlar aniq vazifalarni bajaradigan xizmatlar to'plamini taqdim etadi.
- **Mustaqillik:** Komponentlar bir-biridan mustaqil ravishda ishlaydi va o'zaro bog'liqliklarni minimallashtiradi.

Visual C++ muhitida ishlatiladigan komponentalar quyidagi asosiy toifalarga bo'linadi:

Standart Komponentalar: Bu guruhga *Button*, *Label*, *TextBox*, *ListBox* kabi asosiy foydalanuvchi interfeysi elementlari kiradi. Ular dasturiy ilovada foydalanuvchi bilan muloqot o'rnatishda keng qo'llaniladi. Ushbu komponentalar asosan GUI (Graphical User Interface) elementlarini loyihalash uchun ishlatiladi.

Button: Bu foydalanuvchi interfeysidagi bosish uchun mo'ljallangan tugma.

![Button Komponentasi](file-WvVj9ARAw7lwI0D7MMRGJD0X)

TextBox: Bu matn kiritish uchun mo'ljallangan komponenta bo'lib, foydalanuvchidan ma'lumot olish uchun qo'llaniladi.

Additional Komponentalar: Bu guruhga maxsus foydalanish uchun mo'ljallangan komponentalar kiradi. Misol uchun, *Timer*, *ProgressBar*, va *PictureBox* kabi komponentalar qo'shimcha interfeys imkoniyatlarini taqdim etadi.

Timer: Bu komponenta vaqtni o'lchash va ma'lum bir vaqt oralig'ida avtomatik ravishda ma'lum bir funksiyani bajarish uchun ishlatiladi.

![Timer Komponentasi](file-WvVj9ARAw7lwI0D7MMRGJD0X)

ProgressBar: Bu komponenta biror vazifaning bajarilish darajasini ko'rsatish uchun qo'llaniladi.

System Komponentalari: Bu guruhga *File Dialog*, *Print Dialog*, *System Tray Icon* kabi tizim bilan bog'liq bo'lgan komponentalar kiradi. Bu komponentalar dasturiy ilova va operatsion tizim o'rtasidagi o'zaro aloqani ta'minlash uchun ishlatiladi.

File Dialog: Bu komponenta foydalanuvchi tomonidan faylni tanlash yoki saqlash uchun ishlatiladi.

![File Dialog](file-WvVj9ARAw7lwI0D7MMRGJD0X)

System Tray Icon: Bu komponenta ilovaning kichik ikonasi tizim paneliga joylashtirish imkoniyatini beradi.

C++ da Komponenta Afzalliklari

- Tez ishlab chiqish: Komponentlardan foydalanish dasturni tezroq va samarali ishlab chiqishga imkon beradi.
- Qayta ishlatish: Komponentlar qayta ishlatilishi mumkin, bu esa ishlab chiqish vaqtini va xarajatlarni kamaytiradi.
- Ko'p platformali: C++ komponentlarini turli xil platformalarda ishlatish mumkin.
- Jamoaviy ishni osonlashtiradi: Komponentlar dasturchilar guruhi tomonidan birgalikda ishlab chiqilishi va ishlatilishi mumkin.

C++ da Komponenta Kamchiliklari

- Murakkablik: Komponentlarni yaratish va ularni boshqarish murakkab bo'lishi mumkin.
- Qaramlik: Komponentlar boshqa komponentlarga qaram bo'lishi mumkin, bu esa muammolarga olib kelishi mumkin.
- Integratsiya: Komponentlarni boshqa tizimlarga integratsiyalash qiyin bo'lishi mumkin.
- Xavfsizlik: Komponentlar xavfsizlik muammolariga moyil bo'lishi mumkin.

Visual C++ Muhitida Komponentalar bilan Ishlash

Visual C++ muhitida komponentalar bilan ishlash uchun Toolbox menyusidan kerakli komponentani tanlash va Form Designer oynasiga qo'shish kifoya. Komponentni formaga joylashtirgandan so'ng, unga tegishli xususiyatlar va metodlarni sozlash mumkin. Har bir komponentning o'ziga xos Properties bo'limi mavjud bo'lib, unda komponentning hajmi, ranglari, va boshqa xususiyatlarini o'zgartirish mumkin.

C++ da Komponenta yaratish va ishlatish misollari

- **UI komponentlari yaratish:**

```
#include <iostream>
#include <string>
class Button {
public:
    void click() {
        std::cout << "Tugma bosildi!" << std::endl;
    }
};

int main() {
    Button button;
    button.click();
}
```

```
return 0;  
}
```

Ma'lumotlar bazasi bilan ishlash:

```
#include <iostream>  
#include <sqlite3.h>
```

```
int main() {  
    sqlite3 *db;  
    char *err;  
    int rc = sqlite3_open("mydatabase.db", &db);  
    // Ma'lumotlar bazasiga kirish va ishlash ...  
    sqlite3_close(db);  
    return 0;  
}
```

Biznes mantig'i komponentlari yaratish:

```
class User {  
public:  
    std::string name;  
    int age;  
  
    void showInfo() {  
        std::cout << "Ism: " << name << ", Yosh: " << age << std::endl;  
    }  
};
```

```
int main() {  
    User user;  
    user.name = "Javohir";  
    user.age = 20;  
    user.showInfo();  
    return 0;  
}
```

Xulosa

Ushbu maqolada biz komponenta tushunchasi va xususiyatlarini chuqurroq o'rganib chiqdik. Komponentalar zamonaviy dasturlashda kodni qayta ishlatishni, loyihalarni boshqarishni osonlashtirishni va samaradorlikni oshirishni ta'minlaydigan muhim

qadamlardir.

C++ da komponenta tushunchasi dasturlarni tez va samarali ishlab chiqishda muhim rol o'ynaydi. U dasturlarni tezroq ishlab chiqish, qayta ishlatishni oshirish va dastur kodini samarali boshqarish imkonini beradi. C++ da komponentlarni tushunish va ularni qanday yaratishni o'rganish har qanday C++ dasturchisi uchun juda muhimdir.

Komponentalar - bu mustaqil ravishda ishlaydigan va qayta ishlatiladigan kod bloklari bo'lib, ular aniq vazifani bajarish uchun mo'ljallangan. Bu ularni dasturlash jarayonida qayta ishlatishga imkon beradi, bu esa vaqt va resurslarni tejashga yordam beradi. Shuningdek, ular kodni tashkil qilish va boshqarishni osonlashtiradi, bu esa katta va murakkab loyihalarda juda muhimdir.

Komponentalar turli xil xususiyatlarga ega bo'lib, ularning funksiyalari va imkoniyatlarini belgilaydi. Komponentalar boshqa kod bloklari bilan bog'liq bo'lmasdan o'zlari mustaqil ravishda ishlay olishi kerak. Komponenta dasturlashning boshqa qismlarida yoki boshqa loyihalarda qayta ishlatilishi mumkin. Komponentalar o'z ichki ma'lumotlar va funksiyalarni yashirib, boshqa qismlarga kirishni cheklashi mumkin. Komponentalar o'z xizmatlarini boshqa qismlarga taqdim qilish uchun aniq interfeysga ega bo'lishi kerak.

Shunga qaramay, komponenta tushunchasini tushunish va ularni to'g'ri qo'llash dasturlashning samaradorligini oshirish uchun muhimdir. Ushbu maqolada keltirilgan ma'lumotlar sizga komponenta tushunchasi va xususiyatlarini chuqurroq tushunishda va uni dasturlash jarayonida samarali qo'llashda yordam berishga mo'ljallangan.

Kelajakda komponenta tushunchasi yanada rivojlanib, dasturlash dunyosida muhim rol o'ynaydigan texnologiyalarning asosiy qismlaridan biri bo'lib qoladi.

FOYDALANILGAN ADABIYOTLAR:

1. Abdurakhmonova, N., Alisher, I., & Toirova, G. (2022, September). Applying Web Crawler Technologies for Compiling Parallel Corpora as one Stage of Natural Language Processing. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 73-75). IEEE.
2. Abdurakhmonova, N., Alisher, I., & Sayfulleyeva, R. (2022, September). MorphUz: Morphological Analyzer for the Uzbek Language. In 2022 7th International Conference on Computer Science and Engineering (UBMK) (pp. 61-66). IEEE.

3. NAZIROVA, E., ABDURAKHMONOVA, N., & ALISHER, I. Exploring Linguistic Roots (stem) and Word Categories in Uzbek Language through Advanced Natural Language Processing Techniques for Text Analysis.
4. Ismailov, A. S., Akbarov, A., Qodirova, G. T. X. Q., & Yigitaliyeva, M. (2023). Dasturiy ta'minotni ishlab chiqish bosqichlari. Science and Education, 4(3), 187-191.
5. Ismailov, A. S. "Algoritmlash va dasturlash asoslari 2". O'quv qo'llanma.