12

O‘N IKKINCHI MA’RUZA · 1-QISM

Dasturlash 1 · C++ · 8-mavzu (2 qismdan 1-qism)

# Funksiyalar

Funksiya tushunchasi · Parametrlar va standart qiymatlar · Qayta yuklash va shablonlar

int f(int x)

Overloading

Rekursiya

Lambda

ISINISH

# Nima uchun choyxonada har safar yangi samovar yasalmaydi?

Bir marta tayyorlangan “qurilma” qayta-qayta ishlatiladi.

1

Dasturda bir xil kod 5 joyda takrorlansa, xatoni tuzatish qanchalik qiyin?

2

sqrt(16) ni chaqirganda uning ichida nima bo‘layotganini bilishimiz shartmi?

3

Kod blokiga “ism” berib, uni istalgan joydan chaqira olsak-chi?

O‘QUV MAQSADLARI

# Bu ma’ruza oxirida siz quyidagilarni bajara olasiz...

1. **Farqlash** — funksiya e’loni (prototip), aniqlanishi va chaqiruvini
2. **Yozish** — qiymat qaytaruvchi va void funksiyalarni
3. **Tanlash** — parametr uzatish usulini: qiymat, havola, const havola, massiv
4. **Qo‘llash** — standart (default) qiymatli parametrlarni
5. **Yaratish** — qayta yuklangan va shablon (template) funksiyalarni

01

# Funksiya tushunchasi

Umumiy ko‘rinish · prototip · parametr va argument · void · return

Prototip | Parametr | Argument | void

ANATOMIYA

# qaytariladigan_tur nom(parametrlar) { ... }

| Tushuncha | Izoh | Misol |
|---|---|---|
| **E’lon (prototip)** | Nom, tur va parametrlarni oldindan bildiradi; `;` bilan tugaydi | `int kvadrat(int x);` |
| **Aniqlash (ta’rif)** | Funksiyaning tanasi bilan to‘liq yozilishi | `int kvadrat(int x) { return x * x; }` |
| **Parametr** | E’londagi o‘zgaruvchi | `int x` |
| **Argument** | Chaqiruvdagi aniq qiymat | `kvadrat(7)` dagi 7 |
| **void** | Hech narsa qaytarmaydi | `void salom();` |
| **return** | Natijani qaytaradi va funksiyadan chiqadi | `return x * x;` |

Funksiya main() dan keyin yozilsa, oldin prototip kerak — aks holda “was not declared in this scope” xatosi.

# E’lon oldin, ta’rif keyin

```
int kvadrat(int x);
int kattasi(int a, int b);
int main() {
    cout << kvadrat(7) << endl;
    cout << kattasi(kvadrat(3), 10) << endl;
}
int kvadrat(int x) { return x * x; }
int kattasi(int a, int b) {
    if (a > b) return a;
    return b;
}
```

▶ NATIJA

```
49
10
```

**`int kvadrat(int x);`**

prototip — kompilyatorga “bunday funksiya bor”

**`kattasi(kvadrat(3), 10)`**

bir funksiya natijasi boshqasiga argument: 9 va 10

**`return a;`**

funksiyadan darhol chiqish

# Tub son va EKUB

```
bool tubmi(int n) {
    if (n < 2) return false;
    for (int i = 2; i * i <= n; i++)
        if (n % i == 0) return false;
    return true;
}
int ekub(int a, int b) {
    while (b != 0) { int t = a % b; a = b; b = t; }
    return a;
}
int main() {
    for (int i = 2; i <= 20; i++) if (tubmi(i)) cout << i
            << " ";
    cout << endl << ekub(84, 36);
}
```

▶ NATIJA

```
2 3 5 7 11 13 17 19
12
```

**bool tubmi**

rost/yolg‘on qaytaradi — shartda ishlatish qulay

**return false**

bo‘luvchi topilishi bilan chiqish

**ekub**

Evklid algoritmi: 84, 36 → 36, 12 → 12, 0

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 1-bo‘lim yakuni: Funksiya tushunchasi

1

Hech narsa qaytarmaydigan funksiya qaysi tur bilan e’lon qilinadi?

Savol 1

2

Parametr va argument farqi nima?

Savol 2

3

Prototip qachon kerak bo‘ladi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

# 1-topshiriq: Tub sonlar oralig‘i

`bool tub(int n)` funksiyasini yozing. `main()` tayyor: u `[a, b]` oralig‘idagi tub sonlarni funksiya yordamida chiqaradi.

**Kirish:** Ikki butun son `a` va `b` ($1 \le a \le b \le 100000$).

**Chiqish:** Oraliqdagi tub sonlar bo‘sh joy bilan; bo‘lmasa — `-1`.

| MISOL · KIRISH | MISOL · CHIQISH |
|---|---|
| `10 30` | `11 13 17 19 23 29` |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

02

# Parametrlar va standart qiymatlar

Qiymat · havola · const havola · massiv · default argumentlar

int &x

const string&

int a[]

= 0

QAYSI USUL?

# Parametrni qanday uzatish kerak?

| Holat | Tavsiya |
|---|---|
| **Kichik tur (`int`, `double`) faqat o‘qiladi** | Qiymat bo‘yicha: `void f(int x)` |
| **Funksiya argumentni o‘zgartirishi kerak** | Havola: `void f(int &x)` |
| **Katta obyekt (`string`, `vector`) faqat o‘qiladi** | Konstant havola: `void f(const string &s)` |
| **Massiv** | Ko‘rsatkich + o‘lcham: `void f(int a[], int n)` |

Massiv hech qachon nusxalanmaydi — funksiya ichidagi o‘zgarish asl massivga ta’sir qiladi. Himoya: `const int a[]`.

AMALDA

# Qiymat, havola va massiv

```
void oshirQ(int x) { x += 10; }
void oshirH(int &x) { x += 10; }
int yig(const int a[], int n) {
    int s = 0;
    for (int i = 0; i < n; i++) s += a[i];
    return s;
}
void ikki(int a[], int n) { for (int i = 0; i < n;
        i++) a[i] *= 2; }
int main() {
    int x = 5, a[] = {1, 2, 3};
    oshirQ(x); cout << x << " ";
    oshirH(x); cout << x << endl;
    ikki(a, 3); cout << yig(a, 3);
}
```

▶ NATIJA

```
5 15
12
```

**oshirQ(x)**

nusxa o‘zgardi — x = 5 qoldi

**oshirH(x)**

havola — x = 15

**ikki(a, 3)**

asl massiv o‘zgardi: 2 4 6 → yig‘indi 12

STANDART QIYMATLAR

# Argument berilmasa — default ishlatiladi

```
void chiziq(int n = 10, char c = '-') {
    for (int i = 0; i < n; i++) cout << c;
    cout << endl;
}
double narx(double s, double chegirma = 0, int soni = 1) {
    return s * (1 - chegirma / 100) * soni;
}
int main() {
    chiziq();
    chiziq(5, '*');
    cout << narx(50000) << " " << narx(50000, 10, 3);
}
```

▶ NATIJA

```
----------
*
50000 135000
```

**chiziq()**

n = 10, c = '-'

**narx(50000)**

chegirma = 0, soni = 1

**Qoida**

standart qiymatlar faqat o‘ngdan: `f(int a = 1, int b)` — XATO

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 2-bo‘lim yakuni: Parametrlar va standart qiymatlar

1

Qaysi e’lon to‘g‘ri: `f(int a = 1, int b)` yoki `f(int a, int b = 2)`?

Savol 1

2

Katta string ni faqat o‘qish uchun qanday uzatish kerak?

Savol 2

3

Massiv funksiyaga nusxalanadimi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

# 2-topshiriq: Standart parametr: daraja

`double daraja(double x, int n = 2)` funksiyasini sikl bilan yozing (`pow` ishlatmang). `main()` uni bitta argument bilan (standart `n = 2`) va ikkita argument bilan chaqiradi.

**Kirish:** Haqiqiy son `x` ($|x| \le 10$) va butun son `n` ($0 \le n \le 15$).

**Chiqish:** Ikki qator: `x²` va `xⁿ`, verguldan keyin 3 xona.

| MISOL · KIRISH | MISOL · CHIQISH |
|---|---|
| `3 4` | `9.000`<br>`81.000` |

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

03

# Qayta yuklash va shablonlar

Overloading · kompilyator qanday tanlaydi · template <typename T>

Overloading | Aniq moslik | template | T

QAYTA YUKLASH

# Bir nom — bir nechta funksiya

### 1 Shart

Parametrlar soni yoki turlari farq qilishi kerak. Faqat qaytariladigan tur farqi — XATO.

### 2 Aniq moslik

Tur bir xil: `int → int` — eng yaxshi variant.

### 3 Kengaytirish

`char`, `bool → int`; `float → double`.

### 4 Standart o‘zgartirish

`int → double`, `long → int` va h.k.

### 5 Foydalanuvchi o‘zgartirishi

`const char* → string`.

### 6 Noaniqlik

Ikkita variant teng mos bo‘lsa — “ambiguous” xatosi.

# Qaysi chop() chaqiriladi?

```
void chop(int x) { cout << "int " << x << endl; }
void chop(double x) { cout << "double " << x << endl; }
void chop(const string &s) { cout << "string " << s << endl; }
int main() {
    chop(5);
    chop(2.5f);
    chop('A');
    chop("Salom");
}
```

▶ NATIJA

```
int 5
double 2.5
int 65
string Salom
```

**chop(2.5f)**

float → double (kengaytirish)

**chop('A')**

char → int: 65 chiqadi!

**chop("Salom")**

const char* → string

# template — bitta kod, ko‘p tur

```cpp
template <typename T>
T kattasi(T a, T b) {
    return (a > b) ? a : b;
}
template <typename T>
T yigindi(const T a[], int n) {
    T s = T();
    for (int i = 0; i < n; i++) s += a[i];
    return s;
}
int main() {
    cout << kattasi(3, 9) << " " << kattasi(2.5, 1.5) << endl;
    string s[] = {"C", "++"};
    cout << yigindi(s, 2) << " " << kattasi<double>(3, 4.5);
}
```

▶ NATIJA

```
9 2.5
C++ 4.5
```

**typename T**

tur o‘rnini bosuvchi nom

**T s = T()**

int uchun 0, string uchun ""

**kattasi<double>**

turni aniq ko‘rsatish

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

# 3-bo‘lim yakuni: Qayta yuklash va shablonlar

1

Qayta yuklangan funksiyalar nima bilan farq qilishi shart?

Savol 1

2

`void f(int); void f(double);` bo‘lsa `f('x')` qaysi biri?

Savol 2

3

`kattasi("olma", "nok")` shablonda nima uchun xavfli?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

# 3-topshiriq: Qayta yuklangan kattasi()

Bir xil nomli uchta funksiya yozing: `int kattasi(int, int)`, `double kattasi(double, double)` va `string kattasi(string, string)` (satrlar uchun — **uzunroq**i, teng bo‘lsa birinchisi). `main()` tayyor.

**Kirish:** Birinchi qatorda `q` ($1 \le q \le 100$). Keyingi `q` qatorda: tur (`int`, `double` yoki `string`) va ikki qiymat.

**Chiqish:** Har bir so‘rov uchun bitta qatorda natija (double — verguldan keyin 2 xona).

MISOL · KIRISH

```
3
int 3 7
double 2.5 -1
string olma anor
```

MISOL · CHIQISH

```
7
2.50
olma
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

XULOSA

# Asosiy xulosalar

## 1 Funksiya

`tur nom(parametrlar) { ... return ...; }`; prototip — main dan oldin.

## 2 Uzatish

O‘qish — qiymat yoki `const &`; o‘zgartirish — `&`; massiv — manzil.

## 3 Default

Standart qiymatlar o‘ngdan chapga: `f(int a, int b = 2)`.

TAKRORLASH

# Yakunlashdan oldin oltita savol

1. `void` funksiya `return 5;` qila oladimi?
2. `int f(int x);` — bu nima?
3. Argumentni o‘zgartiruvchi parametr belgisi?
4. `f(int a = 1, int b)` nima uchun xato?
5. Faqat qaytarish turi farq qiluvchi overloading mumkinmi?
6. `template <typename T>` nima uchun kerak?

CHIQISH CHIPTASI

# Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

Ikki sonning o‘rtachasini qaytaruvchi `double ortacha(int a, int b)` funksiyasini yozing.

2

Qiymat bo‘yicha va havola bo‘yicha uzatishning farqini misol bilan tushuntiring.

3

`f(int a, int b = 2)` ni `f(5)` deb chaqirsak, `b` nechaga teng?