

**MOBIL ILOVALAR ISHLAB CHIQISHDA
FUNKSIYALARDAN FOYDALANISH**

A. X. Yuldoshov

Toshkent axborot texnologiyalari universiteti
Samarqand filiali katta o'qituvchisi
azizbektayloq@gmail.com

O. P. Jivanov

Toshkent axborot texnologiyalari universiteti
Samarqand filiali katta o'qituvchisi
01051987oybek@gmail.com

H. A. Majidov

Toshkent axborot texnologiyalari universiteti
Samarqand filiali katta o'qituvchisi
hmajidov17@gmail.com

Annotatsiya: Bu maqolada mobil ilovalar ishlab chiqishda ishlatiladigan funksiyalar haqida tushunchalar va ulardan foydalanish tamoyillari haqida ma'lumotlar keltirilgan.

Kalit so'zlar: Mobil ilova, funksiya, dart dasturlash tili, dry prinsipi, argument, parametrli funksiya, lokal va global funksiyalar, nomlangan funksiyalar, rekursiv funksiya, interaktiv ta'lim, mobil o'yin, math, date.

Bugungi kunda mobil ilovalar ishlab chiqishda funksiyalardan foydalanish alohida ahamiyat kasb etadi. Sababi ko'plab dunyo reytingida IT firmalar dasturchilarni kuzatish jarayonidan ular yaratgan loyihalari va yozgan dastur kodlarini tahlil qiladi. Yozilgan dastur kodlarida "DRY" prinsipidan foyalanilganligiga alohida e'tibor qaratishadi.

"DRY" – don't repeat yourself hisoblanib, dasturlashda bir marta foydalangan kodingizni yana qayta yozmaslik uchun ishlatiladi. Shu maqsadda mobil ilovalarni dart dasturlash tilida ishlatishda funksiyalardan keng foydalanish imkonini beradi[1-3].

Dasturlash tilida funksiya tushunchasi bu yozgan kod faylimizda bir necha bor chaqirish imkoni bo'lgan va ma'lum jarayonni bajarib beruvchi kod qismi hisoblanadi hamda ifodalar to'plamini o'zida saqlaydi.

Dart dasturlash tilida funksiyalar quyidagicha e'lon qilinadi:

```
Ma'lumot_turi funksiya_nomi(){  
// funksiya tanasi  
}
```

Masalan, ikkita sonni qo'shuvchi funksiya yaratamiz:

```

1 void main() {
2   print(qushish());
3 }
4 int qushish(){
5   int a=5,b=6;
6   return a+b;
7 }
    
```

The screenshot shows the DartPad interface with a code editor on the left and a console on the right. The code defines a function `qushish()` that returns the sum of 5 and 6. It is called from `main()` without any arguments. The console shows the output `11`.

1-rasm. Funksiya ishlatilishiga misol.

Bu yerda int funksiya turini butun turga tegishli ekanligini ko'rsatsa, **qushish()** bu funksiya nomi hisoblanadi. Shuningdek funksiya tanasida qiymat qaytarish uchun **return** kalit so'zidan foydalaniladi[4]. Yuqorida keltirilgan funksiya parametrsiz funksiya hisoblanadi. Paramterli funksiya quyidagicha e'lon qilinadi:

```

Ma'lumot_turi funksiya_nomi(parametr){
// funksiya tanasi
}
    
```

Ikkita sonni qo'shuvchi funksiyaning parameter yordamida bajarsak quyidagi ko'rinishga ega bo'ladi:

```

1 void main() {
2   print(qushish(5,10));
3 }
4 int qushish(int a, int b){
5   return a+b;
6 }
    
```

The screenshot shows the DartPad interface with a code editor on the left and a console on the right. The code defines a function `qushish(int a, int b)` that returns the sum of `a` and `b`. It is called from `main()` with arguments `5` and `10`. The console shows the output `15`.

2-rasm. Parametrlil funksiya uchun namuna

Bu yerda a va b parametr sifatida ishlatilmoqda. Parametr qiymatlarini kiritish esa majburiy qilib qo'yildi[5-7].

Agar funksiya tanasida bitta operator ish bajarsa u holda funksiyaning quyidagicha e'lon qilish ham mumkin:

```

void main() {
  print(sayHi());
}
sayHi()=> 'Salom';
    
```

Funksiya parametrlari uch turga bo'linadi:

1. **Postional**

parametrlar:

Positional parametrlar nomsiz aniqlanadi va “[]” qavslar ichida yoziladi. Ularning joylashgan o'rnini bevosita e'lon qilinish tartibiga bog'liq bo'ladi.

```

void main() {
  //functions
}
    
```

```

print(sayHello('Alisher', 20,50));
print(sayHello('Akobir', 60));
}
String sayHello(String name, [int age=10, double weight=40]){
return 'Salom $name. Mening yoshim ${age} da vaznim: ${weight} kg';
}

```

2. Nomlangan parametrlar:

Nomlangan parametrlar nom bilan aniqlanadi va ularning nomlari yordamida ishlatiladi. Ulardan foydalanishda parametrlarni joylashga o'рни ahamiyatga ega bo'lmaydi:

```

void main() {
//functions
print(sayHello('Alisher', age:10, weight:50));
print(sayHello('Akobir',weight: 60));
}
String sayHello(String name, {int age=18, required double weight}){
return 'Salom $name. Mening yoshim ${age} da vaznim: ${weight} kg';
}

```

3. Optional parametrlar:

Parametrlarni kvadrat qavs ichiga olish orqali ixtiyoriy qilishingiz mumkin.

```

void printName({String firstName, String lastName}) {

print('$firstName $lastName');

}

```

Dart dasturlash tilida, boshqa ko'plab dasturlash tillarida bo'lgani kabi, local va global o'zgaruvchilar bor. Har qanday funktsiyadan tashqarida e'lon qilingan o'zgaruvchilar global o'zgaruvchilar hisoblanib, ushbu o'zgaruvchilarga dart faylining istalgan joyidan murojat qilish mumkin[1,5].

```
int globalVariable = 10;

void main() {

    print(globalVariable); // Erishsa bo'ladi chunki global scopeda

}
```

Funksiya doirasida e'lon qilingan o'zgaruvchilar funksiya doirasidadir. Ularga faqat shu funksiya doirasida kirish mumkin. Shuning uchun bu o'zgaruvchilar lokal o'zgaruvchilar hisoblanadi:

```
void lokalFunction() {

    int lokalVar = 5;

    print(lokalVar); // Function ichida foylanaish mumkin.

}

void main() {

    lokalFunction();

}
```

O'z-o'zini qayta qayta chaqirib ishlatadigan funksiyalar rekursiv funksiyalar hisoblanadi. Masalan n! hisoblovchi dasturda rekursiv funksiyadan foydalanish quyidagicha:

```
int factorial(int n) {

    if (n <= 1) {

        return 1;

    } else {

        return n * factorial(n - 1);

    }

}

void main() {

    print(factorial(5)); // Output: 120 (5! = 5 * 4 * 3 * 2 * 1 = 120)

}
```

Dart dasturlash tilida vaqt bilan ishlashga oid DateTime funksiyasidan foydalaniladi.

```
main(){
```

```
DateTime now=DateTime.now();
int currentDay=now.day;
int currentMonth=now.month;
int currentYear=now.year;
int currentWeek=now.weekday;
print("Bugungi kun: $currentDay");
print("Bugungi kun: $currentMonth");
print("Bugungi kun: $currentWeek");
}
```

Bugungi kunda mobil ilovalardan foydalanishda funksiyalardan foydalanish ortib bormoqda. Xususan flutter freymworkida barcha jarayonlar widgetlar asosida qurilgan bo'lib, widgetlarning ishlatilishi esa bevosita funksiyalardan foydalanishga olib keladi.

Foydalanilgan adabiyotlar:

1. Адама Порта. Шарифа Хашеми. "Programming Mobile Applications for Android Handheld Systems: Part 1".
2. Ted Schadler, Josh Bernoff, Julie Ask. The Mobile Mind Shift: Engineer Your Business to Win in the Mobile Moment. 2014.
3. Ёулдошов А.Х., Ходжаев Т.Т., Эрмаматов С.С. Мобильное приложенияев повышении математической грамотности учащихся начальной школы. Сборник докладов научно-практической конференции “Современные информационно-педагогическиетехнологии в цифровизации образования: проблемы и решения”. 11-12 мая 2023 года. 45-48 ст.
4. A. Yuldoshov, Sh. Khodzhaev, T. Khodzhaev. Mathematical Model for Assessing the Reliability of the Functioning of a Distribution Gas Supply Network as a Queuing System. INTERNATIONAL JOURNAL OF THEORETICAL AND APPLIED ISSUES OF DIGITAL TECHNOLOGIES. 2023/3/19. 45-53 pages.
5. Aziz Khujamurodovich Yuldoshov. INFORMATION AND ANALYTICAL ASSESSMENT OF THE FUNCTIONING OF THE GAS SUPPLY NETWORK IN THE EVENT OF EMERGENCY SITUATIONS. Современные инструментальные системы, информационные технологии и инновации. 2022 г. 19-23 ст.
6. Khodzhaev Shukhrat Tolibovich, Yuldoshov Aziz Khujamurodovich, Khodzhaev Tolib Tohirovich. Program for Calculation of the Optimal Distribution of the Planned Amount of Gas on the Gas Supply Network, 2021 International Conference on Information Science and Communications Technologies (ICISCT). 2021/11/3, 1-4 pages.
7. Sh. Khodzhaev, A. Abdukarimov, A. Yuldoshev, T. Khodzhaev. Technology for Digitalization of Research and Evaluation of the Functioning of Territorial Gas Supply Networks. AIP Conference Proceedings. 3147, 030024 (2024).