

## KOTLIN DASTURLASH TILIDA KLASSLAR VA OB'YEKTLAR TUSHUNCHASI

*X.SH.Musayev*

*o'qituvchi, TATU Farg'ona filiali, Farg'ona*

*Z.Q.Ermatova*

*o'qituvchi, TATU Farg'ona filiali, Farg'ona*

*M.I.Abdurahimova*

*talaba, TATU Farg'ona filiali, Farg'ona*

**Abstract.** This article explains how the Kotlin programming language, like other object-oriented programming languages, uses classes to create objects. An object is a derivative of a class.

**Key words and phrases:** concept of class, Person class, properties and methods, sayHello method, go method, personToString method.

**Аннотация:** В этой статье объясняется, как язык программирования Kotlin, как и другие языки объектно-ориентированного программирования, использует классы для создания объектов. Объект является производным от класса.

**Ключевые слова и фразы:** понятие класса, класс Person, свойства и методы, метод sayHello, метод go, метод personToString

**Annotasiya.** Ushbu maqolada Kotlin dasturlash tili boshqa ob'yektga yo'naltirilgan dasturlash tillari kabi ob'yekt yaratish uchun klasslardan foydalanilishi haqida ma'lumot keltirilgan. Ob'yekt klassning hosilasi hisoblanadi.

**Tayanch so'z va iboralar:** class tushunchasi, Person klassi, xususiyatlar va metodlar, sayHello metodi, go metodi, personToString metodi

**Kotlin** dasturlash tili ob'yektga yo'naltirilgan dasturlash tillari sarasiga kiradi. Ya'ni, bu tilda yaratilgan dasturlar bir-biri bilan ob'yektlar yordamida bog'lanadi. **Kotlin** dasturlash tili boshqa ob'yektga yo'naltirilgan dasturlash tillari kabi ob'yekt yaratish uchun klasslardan foydalanadi. Ob'yekt klassning hosilasi hisoblanadi. Masalan, mashina ob'yektini ko'riladigan bo'lsak, u quyidagilardan iborat. Korpus, g'ildiraklar, rul va boshqalar. Bu sanab o'tilgan ma'lumotlar ob'yektning xususiyatlari hisoblanadi.

Bu sanab o'tgan ma'lumotlarni bitta umumiy ma'lumotga birlashtirilganda klass tushunchasi kelib chiqadi. Mashinalar bir-biriga o'xshamaydigan ko'rinshlarga ega. Qaysidir mashinaning korpusi boshqasidan farq qilsa, boshqasining g'ildiraklari farq qiladi. Shuning uchun ob'yekt yaratilayotgan vaqtda uning xususiyatlariga turlicha ma'lumotlarni kiritish mumkin. Klass qanday yaratiladi? Shu savolga quyidagicha javob berish mumkin.

**Kotlin** dasturlash tilida klass yaratish uchun **class** hizmatchi so'zi foydalaniladi. Bu hizmatchi so'zdan so'ng yaratilayotgan klassning nomi keltiriladi va shundan keyin figurali qavslar orasida klassning xususiyatlari yoki maydonlari

keltiriladi. Kotlin dasturlash tilida klassning tana qismi mavjud bo'lmasa, figurali qavslarni yozish shart emas. Masalan, quyida insonni ifodalovchi **Person** nomdagi klass keltirilgan:

```
class Person  
// yoki quyidagi ko'rinishda  
class Person { }
```

Klass boshqa dasturlash tillaridagi kabi Kotlin dasturlash tilida ham foydalanuvchi tomonidan yaratilgan ma'lumotlar turi hisoblanadi. Shuning uchun, yaratilgan klassni ma'lumotlar turi sifatida ishlatish va u yordamida o'zgaruvchilar e'lon qilish mumkin.

```
fun main() {  
    val tohir: Person  
    val bobur: Person  
    val aziza: Person  
}
```

```
class Person
```

Yuqorida keltirilgan dasturning asosiy funksiyasida uchta o'zgaruvchi foydalanuvchi tomonidan yaratilgan ma'lumotlar turi yordamida yaratilmoqda. Shuni ta'kidlash lozimki Kotlin dasturlash tilining asosiy funksiyasi C# va Java dasturlash tillaridagi kabi klasslarda emas balki barcha yaratilgan klasslardan tashqarida yoziladi.

Klassning ob'yektini yaratish uchun ushbu klassning konstruktorini chaqirish kerak bo'ladi. Konstruktor klass nomi bilan yuritiladigan funksiya hisoblanadi. Agar foydalanuvchi tomonidan klassning konstruktori aniqlanmasa, kompiyatorning o'zi bo'sh konstruktor yaratadi va undan foylanish mumkin:

```
fun main() {  
    val tohir: Person = Person()  
    val bobur: Person = Person()  
    val aziza: Person = Person()  
}
```

```
class Person
```

Yuqorida keltirilgan dasturdagi o'zgaruvchilar bu holdan so'ng o'zida ob'yektlarni saqlaydi.

**Xususiyatlar va metodlar.** Har bir klass o'ziga kerakli ma'lumotlarni va holatlarni xususiyatlarda saqlaydi. Xususiyatlar klass darajasida val yoki var kalit so'zlari bilan belgilanadigan o'zgaruvchilarni ifodalaydi. Agar xususiyat val yordamida aniqlansa, unda bu xususiyatga bir marta qiymat tayinlanadi, ya'ni bu xususiyat o'zgarimas bo'ladi. Agar xususiyat var kalit so'zi yordamida aniqlansa, u holda bu xususiyatning qiymatlarini o'zgartirish imkoniyati mumkin bo'ladi.

Xususiyatlar boshlang'ich qiymatga ega bo'lishi mumkin. Masalan, quyida boshlang'ich qiymatga ega bo'lgan xususiyatlarga misol keltirilgan:

```
class Person {  
    var name: String = "Noma'lum"
```

```
var age: Int = 18
}
```

Bu yerda, insonni ifodalovchi Person nomidagi klass yaratilgan bo'lib, uning ikkita xususiyati mavjud. Bu xususiyatlardan bir name – insonning nomini, ikkichisi age – insonning yoshini ifodalaydi. Bu ikki xususiyatlarga boshlang'ich qiymatlar berilgan. Bu ikki xususiyatlar var kalit so'zi bilan aniqlanganligi sababli ularning qiymatini dasturning ixtiyoriy joyida o'zgartirish mumkin:

```
fun main() {
    val bobur: Person = Person() // объект яратиш
    println(bobur.name)          // Noma'lum
    println(bobur.age)           // 18
    bobur.name = "Bobur"
    bobur.age = 25
    println(bobur.name)         // Bobur
    println(bobur.age)          // 25
}
class Person{
    var name: String = "Noma'lum"
    var age: Int = 18
}
```

Xususiyatlarga murojaat qilish uchun ob'yektni ifodalovchi o'zgaruvchining nomi, ob'yektning ichidagi ma'nosini berish uchun nuqta (.) va nuqtadan so'ng xususiyatning nomi ko'rsatiladi. Masalan, o'zgaruvchiga ob'yektning xususiyatini qiymatini olish quyidagicha ifodalanadi:

```
val personName : String = bobur.name
```

Agar ob'yektning xususiyatiga qiymat berish kerak bo'lsa, quyidagicha ma'lumotni uzatish mumkin:

```
bobur.name = "Bobur"
```

### Klass metodlari

Klass ichida o'zgaruvchi e'lon qilib ishlatish kabi klassning ichida funksiyalar e'lon qilish ham mumkin. Klassda e'lon qilingan o'zgaruvchilar maydon yoki xususiyat deb nomlanganidek, klassga tegishli bo'lgan funksiyalarni metodlar deb yuritiladi. Metodlarning vazifasi klassdan olingan ob'yektlarning xatti-harakatini belgilab beradi. Umumiy olganda klassning xususiyatlari va metodlari klass a'zolari deb nomlanadi. Masalan, klass metodi bilan ishlash quyidagi dasturda keltirilgan:

```
class Person{
    var name: String = "Undefined"
    var age: Int = 18
    fun sayHello(){
        println("Salom, mening ismim $name")
    }
    fun go(location: String){
```

```
        println("$name $location \bga bormoqda")
    }
    fun personToString() : String{
        return "Ismi: $name    Yoshi: $age"
    }
}
```

Klassning metodlari oddiy funksiyalar kabi aniqlanadi. Hususan, yuqorida keltirilgan Person klassida 3 ta metod mavjud bo‘lib, bu metodlardan birinchisi sayHello. Bu metod konsolga salomlash va ob‘yektning ismini chiqarishga hizmat qiladi. Ikkinchi metod go metodi bo‘lib, ob‘yektning xarakatini ya‘ni qaysi manzilga borishini ko‘rsatib beradi. Joylashuv manzilini metodga location parametri orqali uzatiladi. Uchinchi metod personToString bo‘lib, ob‘yekt haqidagi ma‘lumotlarni satr ko‘rinishiga keltirib, qiymat sifatida uzatish uchun hizmat qiladi.

Klassning metodlarida klass xususiyatlaridan foydalanish to‘g‘ridan-to‘g‘ri amalga oshiriladi. Yuqorida keltirilgan Person klassida aniqlangan ikki xususiyat mavjud. Bular name va age xususiyatlaridir. Yuqorida klassning o‘zini yaratish ko‘rsatib o‘tilgan. Bu klassdan foydalanish quyidagi to‘liq dasturda keltirilgan:

```
fun main() {
    val tohir = Person()
    tohir.name = "Tohir"
    tohir.age = 37
    tohir.sayHello()
    tohir.go("do'kon")
    println(tohir.personToString())
}
class Person{
    var name: String = "Undefined"
    var age: Int = 18
    fun sayHello(){
        println("Salom, mening ismim $name")
    }
    fun go(location: String){
        println("$name $location \bga bormoqda")
    }
    fun personToString() : String{
        return "Ismi: $name    Yoshi: $age"
    }
}
```

Бу дастурни натижаси консолда қуйидагича бўлади.

```
Salom, mening ismim Tohir
Tohir do'konga bormoqda
Ismi: Tohir    Yoshi: 37
```

**Foydalanilgan adabiyotlar ro‘yxati**

1. R. Zulunov. Preparing the educational process for the era of artificial intelligence. The journal of integrated education and research, Volume 1, issue 4, September 2022, p.261-263
2. R. Zulunov. Use of artificial intelligence technologies in the educational process. Web of Scientist: International Scientific Research Journal (WoS), Volume 3, Issue 10, Oct., 2022, p. 764-770.
3. R. Zulunov. Подготовка образовательного процесса к эпохе искусственного интеллекта. Periodica Journal of Modern Philosophy, Social Sciences and Humanities, 2022, Oct., 11, p. 81-83.
4. Солиев Б. Н. Проблемы моделирования электронных торговых процессов на основе местных характеристик //Исследования молодых ученых. – 2020. – С. 8-11.
5. Halimovich T. T. et al. Monte Carlo method for constructing an unbelised assessment of diffusion problems //European science review. – 2020. – №. 1-2. – С. 7-12.
6. Tozhiev, Tokhirjon Halimovich, et al. "MONTE CARLO METHOD FOR CONSTRUCTING AN UNBELISED ASSESSMENT OF DIFFUSION PROBLEMS." European Science Review 1-2 (2020): 7-12.
7. Muminjonovich, Hoshimov Bahodirjon, and Uzokov Barhayot Muhammadiyevich. "Teaching Children to Programming on the Example of the Scratch Program." Eurasian Scientific Herald 9 (2022): 131-134.