```
class FriendList {
    fun friendList() { //Top up függvény
        val bestFriends :ArrayList<String> = arrayListOf("ryan")
        var bestFriendInput: String? //? elvis az értéke lehet null is
        var deleteBestFriendsInput: String?
        var chooseMenu: String?

        while (true) {
            //menü kiíratása
            println("\nChoose an option:")
            println("1. Add friend to list") // Elírás javítva és szöveg
egyszerűsítve
            println("2. Delete Best Friend from list")
            println("3. View friend list")
            println("4. Exit")
            print("Enter your choice: ")
            chooseMenu = readlnOrNull() // Beolvasás a cikluson bel
            //ugrás a menüre switch case a javában
            //itt nincs szügség a break-re mindig:
            when (chooseMenu) {
                "1" -> {
                    println("Add to the list : ")
                    bestFriendInput = readlnOrNull()
                    if (!bestFriendInput.isNullOrBlank()) {
                        bestFriends.add(bestFriendInput)
                        println("$bestFriendInput - added to the list. ")
                        println(bestFriends.joinToString("\n"))
                    } else {
                        println("cannot add to the list")
                    }
                }
                "2" -> {
                    print("Name of Best Friend to delete: ")
                    deleteBestFriendsInput = readlnOrNull()
                    if (!deleteBestFriendsInput.isNullOrBlank()) {
                        if (bestFriends.remove(deleteBestFriendsInput)) {
                            println("$deleteBestFriendsInput removed.")
                        } else {
                            println("$deleteBestFriendsInput not found.")
                        }
                    } else {
                        println("Name to delete cannot be empty.")
                    }
                }

                "3" -> {
                    //println(bestFriends.joinToString("\n"))
                    //vagy for lista
                    for (friend: String in bestFriends)
                        println(friend)
                }
```

```
                "4" -> {
                    println("Exiting program")
                    break //kilépés a while ciklusbol
                    //System.exit(0)
                }
                else -> {
                    println("Invalid choice. Please select from 1 to 4.")
                }
            }//when vége
        }//while vége
    }
}
        fun main() {
            val friendListManager = FriendList()
            friendListManager.friendList()
        }
```

elemzés:

Nézzük át részletesen a kódodat, Ati!

Áttekintés

Ez egy egyszerű Kotlin CLI program, amely barátokat kezel egy listában. A funkciói:

- Barát hozzáadása
- Barát törlése
- A barátlista megtekintése
- Kilépés a programból

Elemzés

1. Osztály, metódus, változók

```
class FriendList {
    fun friendList() {
        val bestFriends :ArrayList<String> = arrayListOf("ryan")
        var bestFriendInput: String?
        var deleteBestFriendsInput: String?
        var chooseMenu: String?
```

- Az osztály és függvény elnevezése világos.
- A `bestFriends` lista eleinte tartalmaz egy "ryan" nevű barátot.
- `String?` típusú változók a felhasználói bemenethez — jó, mert kezelhető a `null` érték.

---

2. Menü és ciklus

```
while (true) {
    println("\nChoose an option:")
    println("1. Add friend to list")
    println("2. Delete Best Friend from list")
    println("3. View friend list")
    println("4. Exit")
    print("Enter your choice: ")
    chooseMenu = readlnOrNull()
```

- Végtelen ciklus, amiben a menü ismétlődik.
- `readlnOrNull()` használata helyes, mert kezelhető, ha nincs input.

---

3. Választás kezelése `when`-nel

```
when (chooseMenu) {
    "1" -> {
        println("Add to the list : ")
        bestFriendInput = readlnOrNull()
        if (!bestFriendInput.isNullOrBlank()) {
            bestFriends.add(bestFriendInput)
            println("$bestFriendInput - added to the list. ")
            println(bestFriends.joinToString("\n"))
        } else {
            println("cannot add to the list")
        }
    }
    "2" -> {
        print("Name of Best Friend to delete: ")
        deleteBestFriendsInput = readlnOrNull()
        if (!deleteBestFriendsInput.isNullOrBlank()) {
            if (bestFriends.remove(deleteBestFriendsInput)) {
                println("$deleteBestFriendsInput removed.")
            } else {
                println("$deleteBestFriendsInput not found.")
            }
        } else {
            println("Name to delete cannot be empty.")
        }
    }
    "3" -> {
        for (friend: String in bestFriends)
            println(friend)
    }
    "4" -> {
        println("Exiting program")
        break
    }
    else -> {
        println("Invalid choice. Please select from 1 to 4.")
    }
}
```

- A menü opciók jól kezeltek.
- Hozzáadásnál és törlésnél ellenőrzöd, hogy ne legyen üres vagy `null` a bevitel.
- A törlésnél visszajelzel, ha a név nem található.
- A barátlista megjelenítése korrekt.
- Kilépésnél a `break` jó megoldás.
- Hibás választás esetén üzenet, így user-friendly.

---

4. `main` függvény

```
fun main() {
    val friendListManager = FriendList()
    friendListManager.friendList()
}
```

- Egyszerű indító metódus, amely létrehozza az osztályt és elindítja a menüt.

Összegzés

- **Minden funkció rendben van, jól működő, stabil kód.**
- Hibakezelés megfelelő.
- A menü világos, logikus.
- Kotlin nyelvi funkciók jól használva (`ArrayList`, `readlnOrNull()`, `when`).
- A kód könnyen érthető és bővíthető.

Tippek / fejlesztési ötletek

- A `bestFriends` lista legyen osztály szintű változó, ha a barátlista megőrzése az osztály életciklusán túl is fontos (most a `friendList()` lokális).
- A `readlnOrNull()` miatt `null` esetére érdemes lehet ciklust vagy ismétlést bevezetni, hogy ne menjen tovább, ha érvénytelen input van.
- Ha akarod, később mentheted fájlba, vagy betöltheted onnan.
- Esetleg a nevek összehasonlításánál használhatsz kis/nagybetű érzéketlenséget (`equalsIgnoreCase`).

```
fun main (){
    //példányosítások
    val bestFriends = arrayListOf("ryan")
    var bestFriendInput: String =""
    var deleteBestFriendsInput : String =""
    var optionPicked : String =""

    while (optionPicked != "3") {
        println("1.Add friedns to list.")
        println("2.Delete Best Friends to list.")
        println("3.Exit")
        print("Choose an option: ")
        optionPicked = readLine() ?:""
```

```
        if (optionPicked == "1"){
            bestFriendInput ="placeholder"
            while (bestFriendInput.isNotBlank()){
                print("What's your BestFriends?: ")
                bestFriendInput = readLine() ?:""
                if (bestFriendInput.isNotBlank()){
                    bestFriends.add(bestFriendInput)
                }
            }
            println("Your Best Friends are: ")
            for (friend in bestFriends){
                println(friend)
            }
        }else if (optionPicked == "2"){
            println("Name of Best Friend to delete: ")
            deleteBestFriendsInput = readLine() ?:""
            if (bestFriends.remove(deleteBestFriendsInput)){
                println("$deleteBestFriendsInput removed.")
            }else{
                println("$deleteBestFriendsInput not found.")
            }
        }else if  (optionPicked == "3") {
            println("Exiting program")
            break
            //System.exit(0)
        }else {
            println("Invalid option")
        }
        println()//üres sor
    }



}
```

elemzés a második verziot:

Rendben, nézzük át részletesen a mostani verziódat!

Elemzés: Kotlin CLI barátlista kezelő program

```
fun main (){
    // példányosítások
    val bestFriends = arrayListOf("ryan")
    var bestFriendInput: String =""
    var deleteBestFriendsInput : String =""
    var optionPicked : String =""

    while (optionPicked != "3") {
        println("1.Add friends to list.")
        println("2.Delete Best Friends from list.")
```

```
        println("3.Exit")
        print("Choose an option: ")
        optionPicked = readLine() ?: ""

        if (optionPicked == "1"){
            bestFriendInput ="placeholder"
            while (bestFriendInput.isNotBlank()){
                print("What's your BestFriends?: ")
                bestFriendInput = readLine() ?: ""
                if (bestFriendInput.isNotBlank()){
                    bestFriends.add(bestFriendInput)
                }
            }
            println("Your Best Friends are: ")
            for (friend in bestFriends){
                println(friend)
            }
        } else if (optionPicked == "2"){
            println("Name of Best Friend to delete: ")
            deleteBestFriendsInput = readLine() ?: ""
            if (bestFriends.remove(deleteBestFriendsInput)){
                println("$deleteBestFriendsInput removed.")
            } else {
                println("$deleteBestFriendsInput not found.")
            }
        } else if  (optionPicked == "3") {
            println("Exiting program")
            break
        } else {
            println("Invalid option")
        }
        println() // üres sor a tisztább megjelenésért
    }
}
```

Mit csinál a program?

- Egy barátlista kezelésére alkalmas egyszerű parancssoros alkalmazás.
- Alapból tartalmaz egy `"ryan"` nevű barátot.
- Menüben 3 lehetőséget kínál: barát hozzáadása, barát törlése, kilépés.
- A barátok listáját egy `ArrayList<String>`-ben tárolja.

Jó megoldások, erősségek

- **Egyszerű és érthető logika**: könnyen követhető a menü kezelése.
- **Null biztonság**: a `readLine() ?: ""` használatával nullát biztonságosan üres stringgé alakítja.
- **Ismétlődő barát hozzáadás ciklus**: a 1-es opció alatt addig kér új barátokat, amíg nem üres inputot kap, ami jó UX (addig adhatsz új barátot, ameddig akarsz).
- **Törléskor ellenőrzés**: kiírja, hogy sikerült-e törölni vagy nem talált bejegyzést.
- **Érvényes opciók kezelése**: a nem létező választásokra üzenetet ír ki.

Amit érdemes lehet javítani vagy átgondolni

1. **A barát hozzáadás ciklusában a "placeholder" érték használata kicsit furcsa:**
   - A `bestFriendInput ="placeholder"` célja, hogy a ciklus elinduljon, mert `bestFriendInput` nem üres.
   - Alternatív megoldás lehet egy do-while vagy más megközelítés, hogy ne kelljen felesleges értéket beállítani.

   Például:

   ```
   do {
       print("What's your BestFriend?: ")
       bestFriendInput = readLine() ?: ""
       if (bestFriendInput.isNotBlank()) {
           bestFriends.add(bestFriendInput)
       }
   } while (bestFriendInput.isNotBlank())
   ```

2. **Írásjelek és kis-nagybetűk a menüpontoknál:**
   - Pl. `"Add friedns to list."` elírás → `"Add friends to list."`.
   - `"Delete Best Friends to list."` helyett inkább `"Delete Best Friend from list."`.
   - Ez csak esztétikai és helyesírási kérdés, de jobb a felhasználói élmény.
3. **Több szóból álló barát nevek esetén a törlés működik, mert `remove()` pontos egyezést keres, de:**
   - Célszerű lehet megmutatni a listát törlés előtt, hogy a felhasználó tudja, mit törölhet.
   - Esetleg kezelni lehetne kis-nagybetű érzékenységet (pl. `bestFriends.removeIf { it.equals(deleteBestFriendsInput, ignoreCase = true) }`).
4. **Program bővíthető fájlkezeléssel (mentés/betöltés), ahogy korábban említetted.**
5. **Kód olvashatósága érdekében érdemes lehet a menü kezelését külön függvénybe vagy osztályba szervezni.**

Összefoglalás

- Ez a verzió már jól működő, tiszta és hibakezelés szempontjából is rendben lévő alap parancssori alkalmazás.
- Az `ArrayList` dinamikusan bővíthető és módosítható, inputok ellenőrzöttek.
- Néhány kisebb javítással (pl. do-while, helyesírás) még jobb lesz a program.
- Ha tovább fejleszted (pl. fájlkezelés, több funkció, GUI), akkor az alap jó kiindulópont.
