

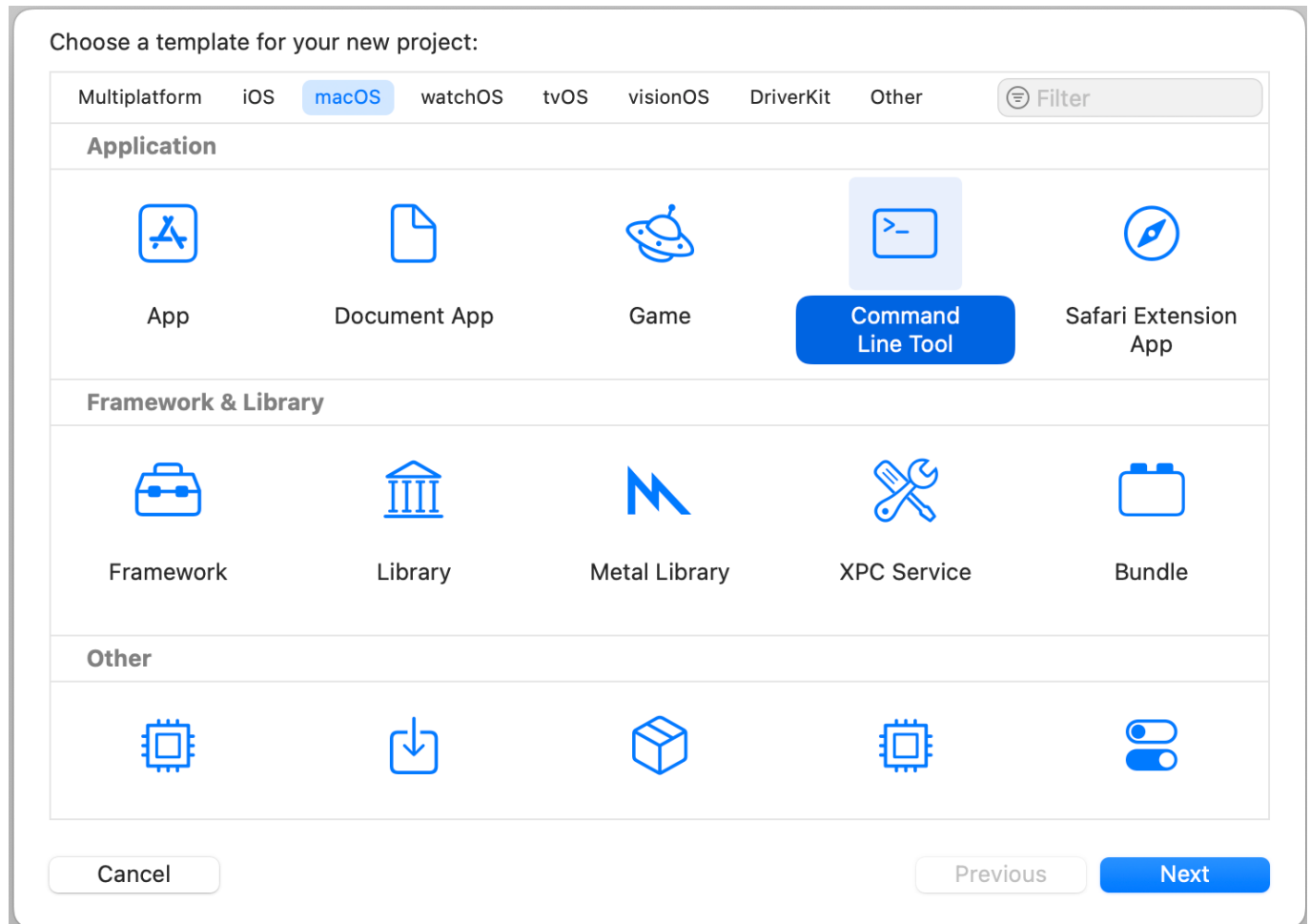
iOS Labor 2

Labor során a *csodálatos* Playground helyett egy macOS-es konzol alkalmazást fogunk készíteni, hogy gyakorolni tudjuk a Swift nyelvet.

Nyisd meg az Xcode-ot.

Hozz létre egy új projektet vagy nyitóképernyőn, vagy a *File > New > Project...* lehetőséget választva.

A macOS-es sablonok közül válaszd a macOS fület, majd a *Command Line Tool* opciót.

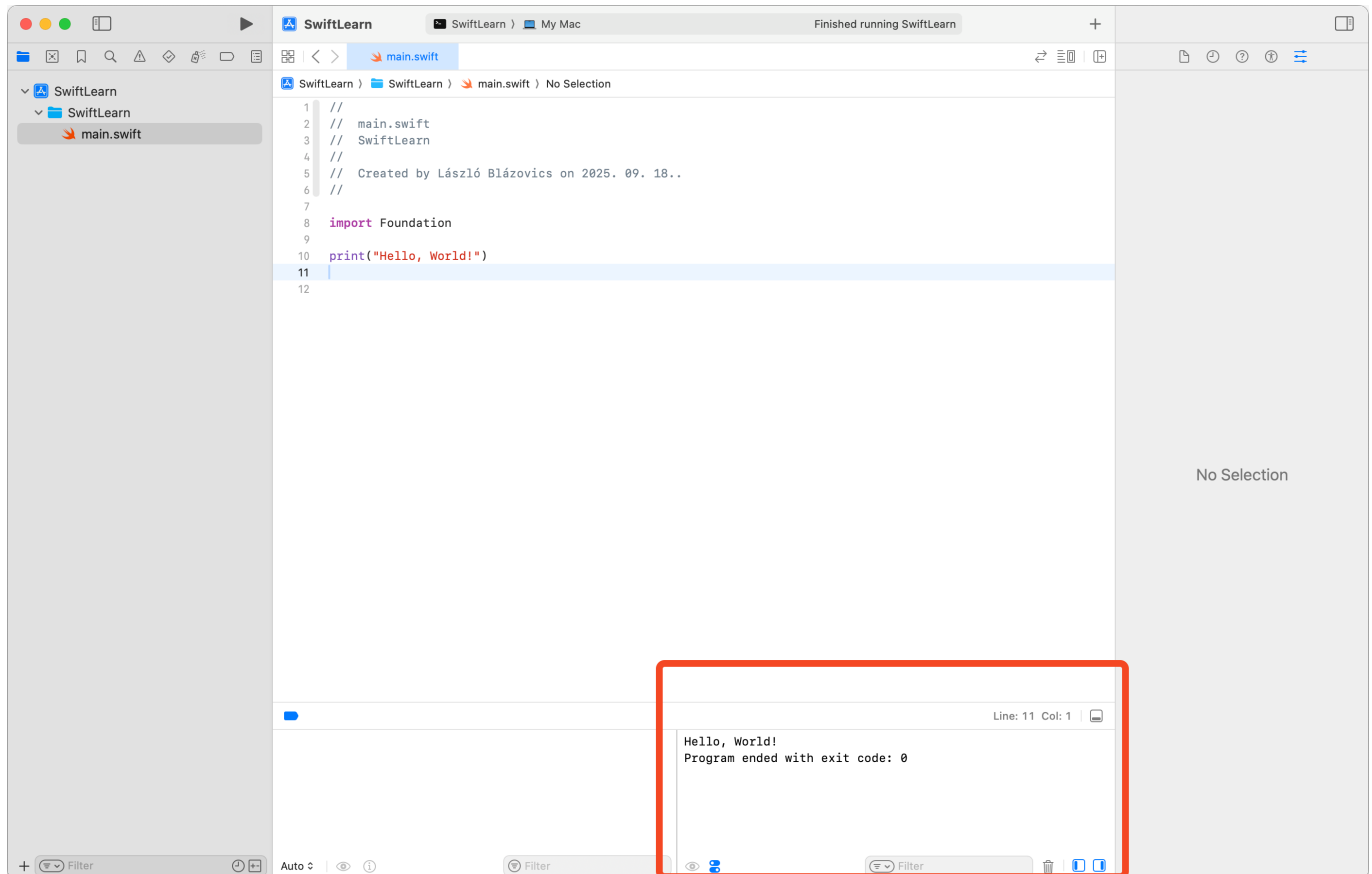


Add meg a projekt nevét (pl. SwiftPractice), és győződj meg róla, hogy a *Language* lenyíló menüben a **Swift** van kiválasztva.

Kattints a Next, majd a Create gombra a projekt létrehozásához.

A bal oldali navigációs sávban (Project Navigator) nyisd meg a `main.swift` fájlt. Ide fogjuk írni a kódunkat.

A terminál az alábbi képen látható bekeretezve:



1. feladat - "Gondoltam egy számra" játék

Generálj egy véletlenszerű számot 1 és 100 között.

Ehhez használd ki az `Int` alábbi metódusát, ami egy `Range`-et (tartományt) vár bemeneti paraméternek.

```
Int.random(in: ...)
```

Kérd meg a felhasználót, hogy tippelje meg a számot.

Ehhez használd a `print()`, valamint a `readLine()` metódusokat.

Adj Visszajelzést, hogy a tipp túl alacsony, túl magas, vagy helyes.

Valósítsd meg, hogy addig ismétlődjön a tippelés, amíg a felhasználó el nem találta a számot.

Helyes tipp esetén írd ki, hogy hány próbálkozásból sikerült kitalálni a számot.

2. feladat - Webshop

A cél egy egyszerű webshop logikájának felépítése

Készíts egy árkezelő rendszert, ami teljesíti az alábbi feltételeket:

- Minden terméknek van neve és ára.
- Képes különböző típusú termékeket (pl. könyv, laptop) kezelni.
- A termékeket kosárba lehet helyezni és ki is lehet őket onnan venni.

- Egy kosárba különböző típusú termékeket lehet helyezni.
- A kosár képes kiszámolni a végösszeget.
- A végösszegre különböző típusú kedvezményeket lehet alkalmazni (pl. százalékos vagy fix összegű).

Protocol

Mielőtt konkrét termékeket hoznánk létre, definiálnunk kell egy közös interfészt, aminek minden eladható terméknek meg kell felelnie. Ezt Swift-ben protokollal tehetjük meg.

Hozz létre egy **Product** nevű protokollt. Majd definiáld az alábbi propertyket és metódust

- **name** (legyen **String** típusú).
- **price** (legyen **Double** típusú).
- **ID** (mivel egyedinek kell lennie, legyen **UUID** típusú).
- **productInfo()**, ami egy **String**-et ad vissza. Ennek a metódusnak a feladata lesz, hogy egy rövid leírást adjon a termékről.

Struct

Most, hogy megvan a szabályrendszer, hozzuk létre a konkrét terméktípusokat. Ezek egyszerű adattárolók lesznek, ezért a **struct** a legalkalmasabb erre. Így már világossá is vált, hogy miért protocol-t és nem össztályt hoztunk létre.

Hozd létre a **Book** nevű **struct**-ot, ami megvalósítja a **Product** protokolt.

Ezután implementálnod kell minden tulajdonságot (nev, ár, termékID) és metódust (termekInfo), amit a protokoll előír. Ebben az Xcode segíteni tud, a hibaüzenet végén a **Fix** felíratra kell csupán kattintani.

Adj hozzá a **Book**-hoz egy extra propertyt is, például **author**, ami legyen **String** típusú.

A **productInfo()** metódusban add vissza a könyv nevét és a szerzőjét.

Ismételd meg ezt a folyamatot egy Laptop nevű struktúrával is. Ennek is tegyél egy egyedi tulajdonságot, például cpu (**String** típusú), és a termékInfo() metódusa ezt az információt is adja vissza.

Enum

A kedvezményeknek több fajtája lehet (százalékos, fix összegű). Ahelyett, hogy ezt szöveggel vagy számokkal próbálnánk kezelni, használjunk egy **enum**-ot, ami sokkal biztonságosabb.

Hozz létre egy **Discount** nevű enumot.

Definiáld benne két case-t:

- Egy **Percentage** case-t, amihez **associated value**-ként kapcsolj egy **Double** típust (ez lesz a kedvezmény mértéke).

- Egy **FixAmount** case-t, amihez szintén egy **Double** értéket kapcsolj (ez lesz a levonandó összeg).

Ezzel létrehoztál egy típust, ami egyszerre képes reprezentálni a kedvezmény fajtáját és annak értékét is.

Class

A bevásárlókosár egy összetettebb entitás, így erre a célra egy **class**-t fogunk létrehozni.

Hozd létre a **Cart** nevű osztályt.

Adj hozzá egy privát attributumot, ami a termékeket fogja tárolni. Ennek a típusa egy tömb legyen, ami a **Product** protokollnak megfelelő elemeket tartalmaz.

Írj egy-egy metódust, melyek a következő feladatokat végzik:

- Hozzáad egy terméket a kosárhoz
- Elvesz egy terméket a kosárból

Végül valósítsuk meg a végösszeg kiszámítását végző metódust:

A metódusnak legyen egy opcionális paramétere: egy **Discount** típus.

- A metóduson belül először számold ki a kosárban lévő összes termék árának összegét. Használd a **reduce** metódusát a tömbnek.
- Ezután vizsgáld meg, hogy a metódus kapott-e kedvezményt. Ha igen, egy **switch** segítségével vizsgáld meg a kedvezmény típusát (szazalekos vagy fix összegű). A **switch** ágain belül a megfelelő logika szerint számold ki a kedvezményes árat az **associated value** felhasználásával.
- Végül a metódus adja vissza a kiszámolt végső árat. Ha nem volt kedvezmény, akkor az eredeti összeget adja vissza.

Teszt

Most, hogy minden elem a helyén van, keltsük életre a rendszert a main.swift fájlban egymás utáni parancsokkal szimulálva a vásárlást.

Hozz létre néhány konkrét termék példányt feltöltve adatokkal. Hozz létre egy **Cart**-ot. Add hozzá a kosárhoz a termékeket. Számold ki a végösszeget kedvezménnyel, illetve anélkül is.