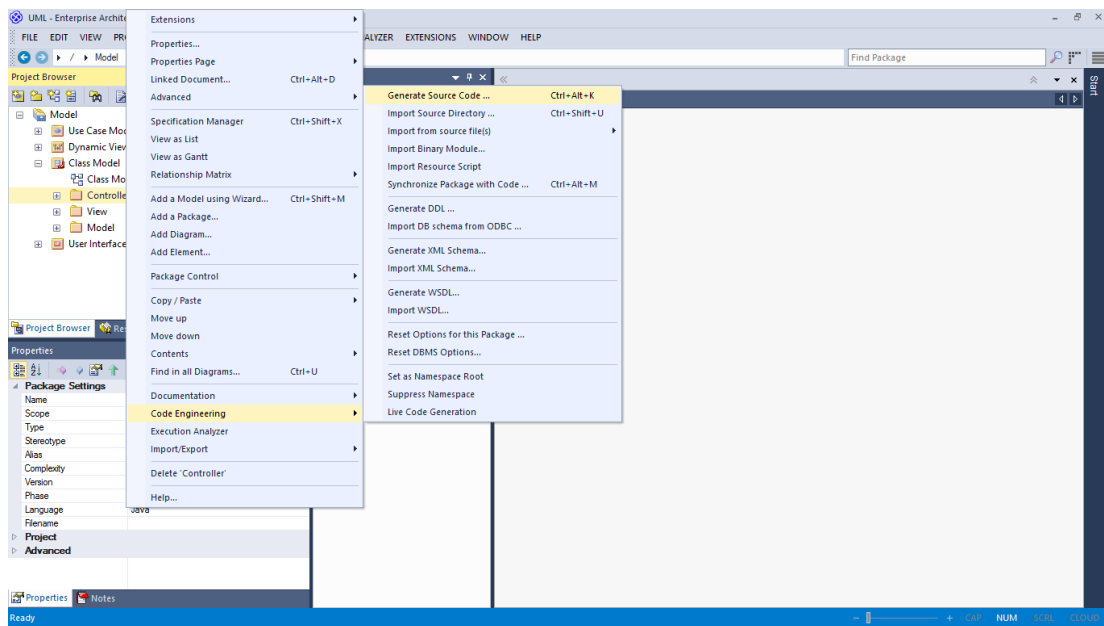


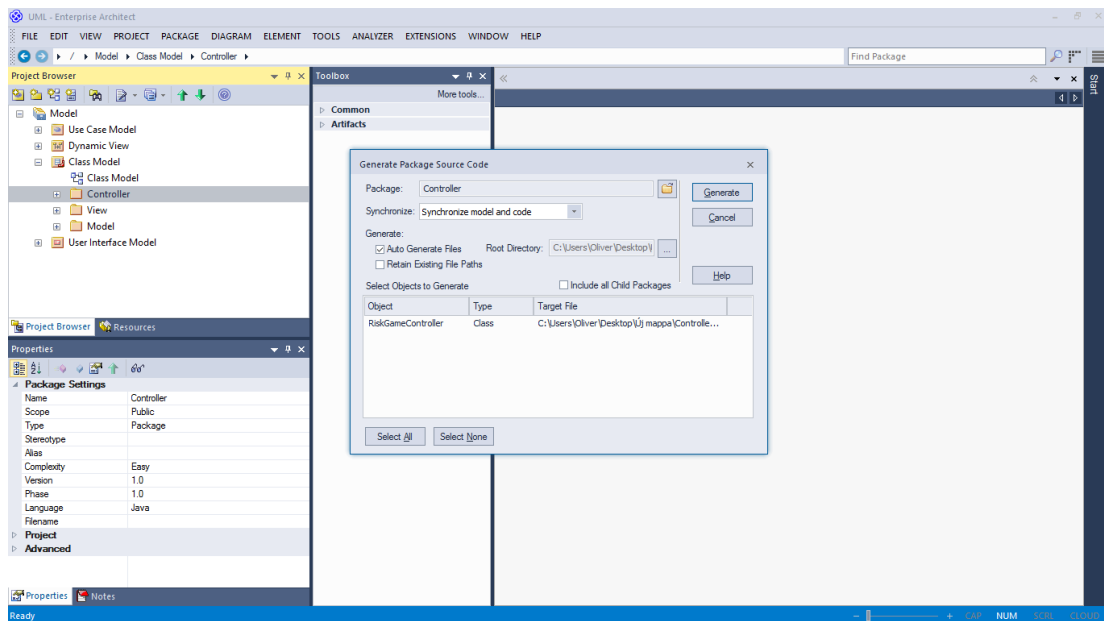
Kódgenerálás és szinkronizáció az Enterprise Architect-ben

1. Kódgenerálás

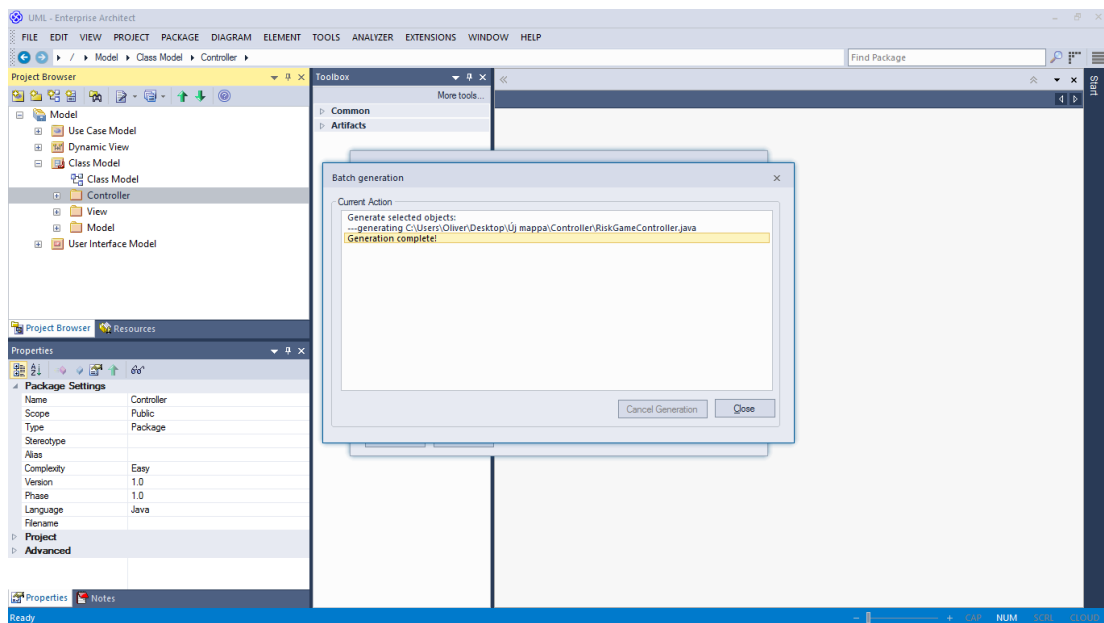
Kódgenerálás a belső Code Generator feature segítségével történik. A komplett Class model legenerálásához a Class Model-ben található package-eket kell kigenerálni. Ha a Controller package-t szeretnénk legeneráltatni, ahhoz a Controller -en jobb egérgomb -> Code Engineering -> Generate Source Code menüpontot válasszuk.



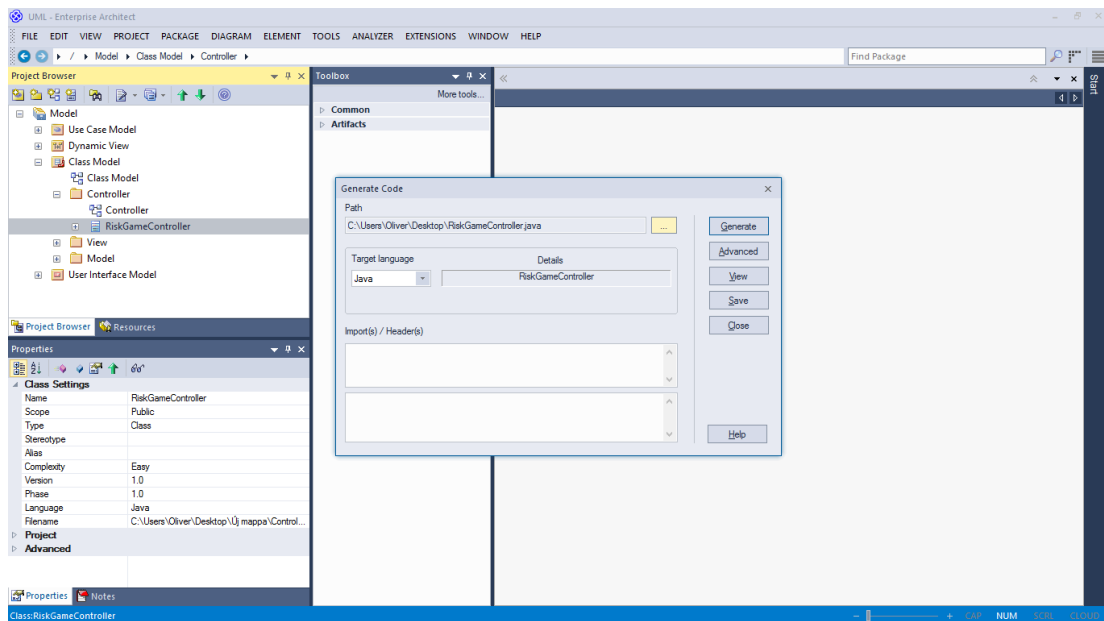
Ekkor egy új ablak nyílik meg, ahol a Generate alatti „Auto Generate Files” bepipálása majd a generált file-ok helyének megadása után már csak a Generate gombra kell kattintanunk. Amennyiben nincs olyan objektum, melyet ki lehetne generálni, akkor az alsó lista üres. Ez azt eredményezi, hogy nincs mit kigenerálnia a Code Generator-nak. Ellenőrizzük, hogy a megfelelő package-t választottuk ki (és nem a komplett Class Model-t) illetve, hogy az tartalmaz objektumokat.



A Generate gombra kattintás után a Batch generation ablak fog előjönni, mely tájékoztat az aktuális objektumok kódjainak legenerálásáról.



Amennyiben ez nem működik, akkor nyissuk le az aktuális package-t, és egy objektumon kattintsunk jobb egérgombbal, majd Code Engineering -> Generate Code. Ekkor már ki bírjuk választani a programozási nyelvet, melyre az aktuális package-t le szeretnénk generáltatni. Ezután kattintsunk a Generate gombra.



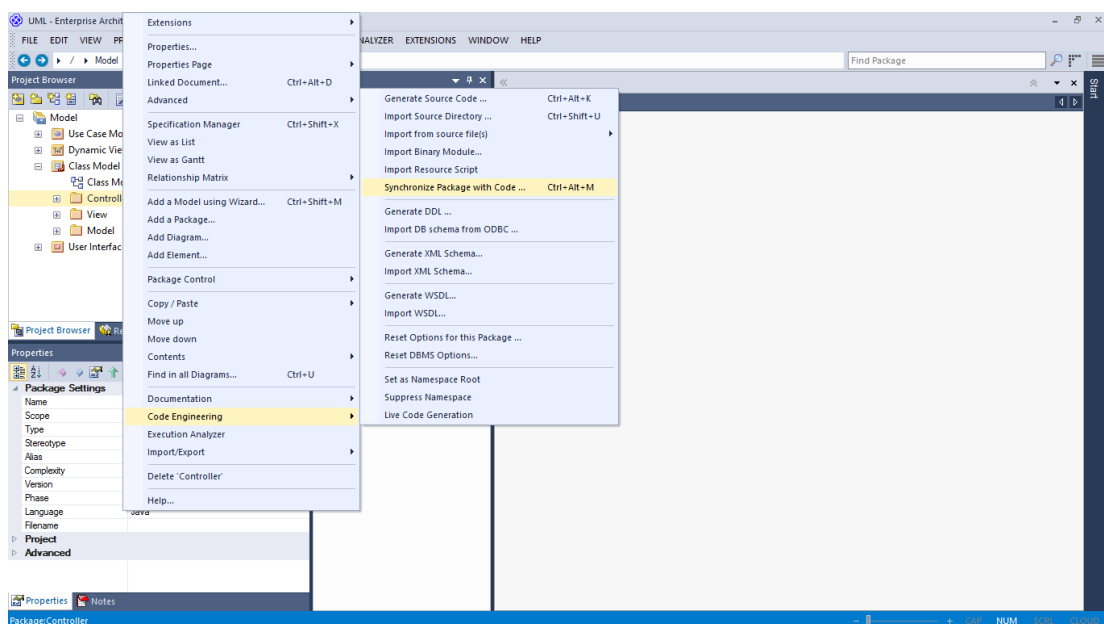
Ismételjük meg a kódgenerálást, míg az összes package-t ki nem generáluk!

2. Kód visszaalakítása a modellben

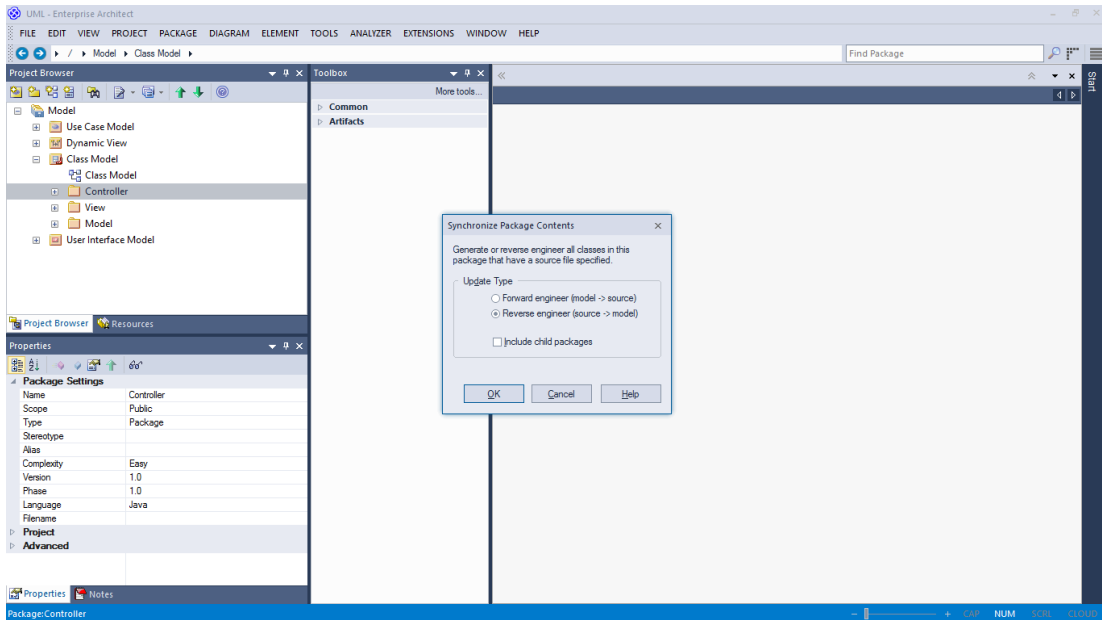
A kódgenerálás után lehetőségünk van a kód visszaalakítására azaz a modell és a legenerált kód szinkronizációjára. Ezt kétféleképpen tehetjük meg:

- Az aktuális package-t szinkronizáljuk, ezzel az összes objektumot frissítjük
- Csak az objektumot szinkronizáljuk.

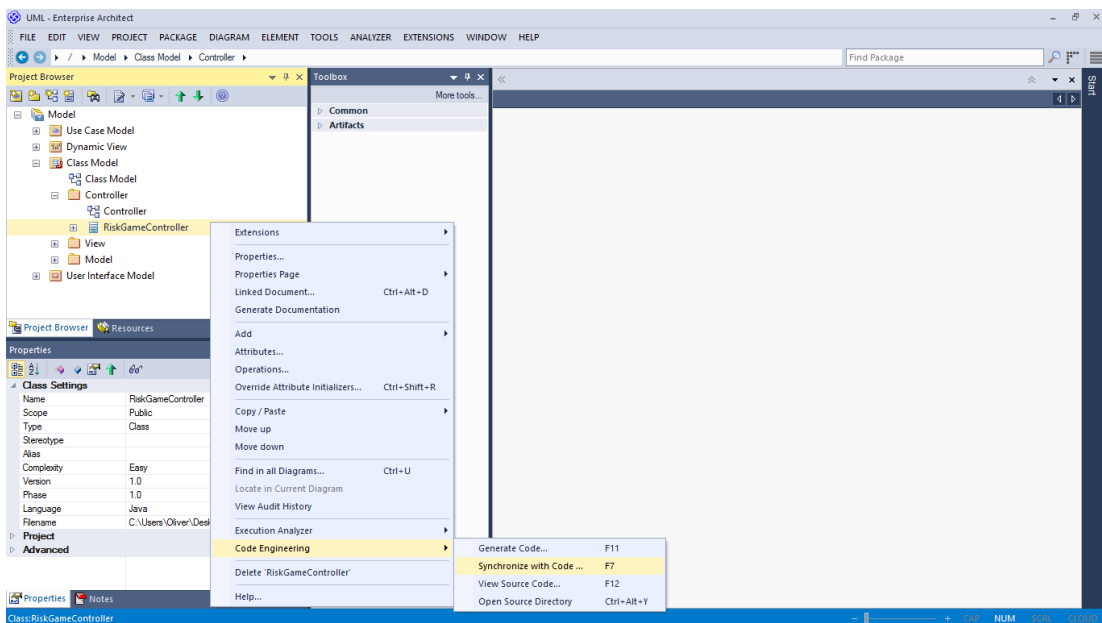
Az a) esetben a kiválasztott package-n jobb egérgomb -> Code Engineering -> Synchronize Package with Code.



Az előugró Synchronize Package Contents ablakban választhatjuk ki, hogy melyik irányba szeretnénk szinkronizálni. Válasszuk a Forward Engineer radiobutton-t, ha model-t szeretnénk a forráskóddal szinkronizálni. Amennyiben a forráskódot szeretnénk a model-el szinkronizálni, válasszuk a Reverse Engineering radiobutton-t. Mivel most vissza szeretnénk alakítani a forráskódot modellé, válasszuk ki ezt, ha nem ez van kiválasztva alapértelmezetten. Az OK gomb lenyomása után végrehajtódik a szinkronizáció.



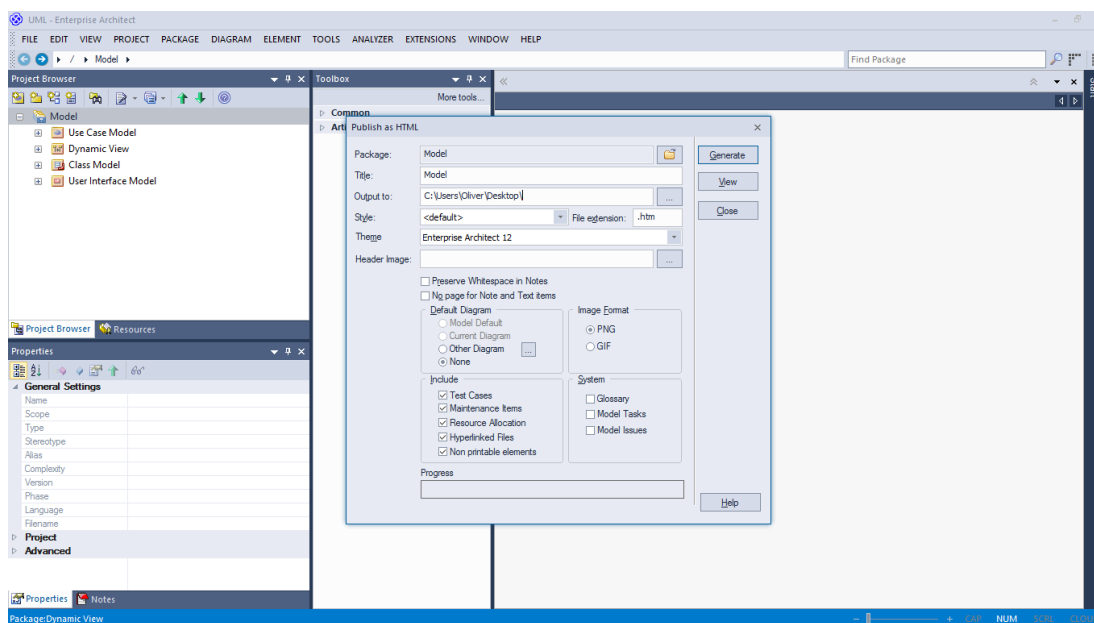
Ha csak egy-egy objektumot szeretnénk szinkronizálni azaz a b) eset áll fenn, akkor a kiválasztott objektumon jobb egérgomb -> Code Engineering -> Synchronize with Code-ot válasszuk ki.




Ezután a model már frissülni is fog a forráskódnak megfelelően.

3. Dokumentáció legenerálása

Dokumentumok generálására számtalan lehetőség van, de a legjobb választás az, ha HTML dokumentációt generáltatunk. HTML dokumentációt generáltathatunk a következőkhöz: package-k, Model-ek és View-ök illetve a komplett projekthez is. Legcélszerűbb a végén az egész projektből generáltatni a dokumentációt. Ehhez kattintsunk a projektre, majd jobb egérgomb -> HTML report. Az előugró Publish as HTML ablakban a Generate gombra való kattintás után már kész is a kigenerált dokumentáció.



Az általunk kiválasztott mappában az index.html megnyitása után láthatjuk a dokumentációt, mely minden adatot tartalmaz, többek között a megjegyzéseket vagy a készítőket nevét is. Célszerű Mozilla Firefox böngészővel vagy Chrome-tól eltérő más böngészővel megnyitni, mert a Chrome blokkolja az Active-X vezérlőket.

ENTERPRISE
ARCHITECT

Model

- Model
 - Use Case Model
 - Dynamic View
 - Class Model
 - Controller
 - RiskGameController**
 - View
 - User Interface Model

RiskGameController : Public Class
Created: 2017.03.01. 10:08:31
Modified: 2017.03.08. 22:49:09
Project: Szabó Dávid
Author: 1.0
Version: 1.0
Phase: Proposed
Status: Easy
Complexity: Easy
Advanced: (82D4333E-ECDC-4044-96E6-F164ED310FAE)
UUID: Controller
Appears In: Controller

AttributesOperationsAssociations FromOther Links

Operation
Public
actionPerformed(@ e: ActionEvent,
}
Details:
Sequential
Notes: A nézetben történt eseményeket (button click, stb) a controller kezeli le.
Public
finalize(): void
Details:
Sequential
Tags: throws=Throwable
Public
RiskGameController():
Details:
Sequential