

Somkutas Péter

GONDOLATOK EGY CSATLAKOZÓFELÜLETET ÉRINTŐ JOGVITA KAPCSÁN – ORACLE V. GOOGLE

Amikor a szoftverek szerzői jogi védelmét az irodalmi művekével megegyező módon szabályozták,¹ viták kísérték mind a döntést, mind az odavezető utat. A nemzetközi viták azóta legfeljebb csendesedtek, de nem múltak el, és az elmúlt másfél évtizedben ismét többször kérdőjelezték meg ennek az oltalmi formának a hatékonyságát. Ebben a cikkben két számítástechnikai óriás, az Oracle America Inc. (Oracle) és a Google Inc. (Google) közötti jogvita kapcsán felmerült szerzői jogi kérdések közül a csatlakozófelülettel (ezen belül is az API-val) foglalkozunk, bemutatjuk az API-k iparági fontosságát, és áttekintjük az ügyben hozott amerikai döntések legérdekesebb – és akár a hazai jog szempontjából is elgondolkodtató – részeit. Kísérletet teszünk egyben annak bemutatására is, hogy a hatályos szerzői jog egy ennyire mélyen technológiai kérdést is képes érdemben megragadni, így ez az oltalmi forma hatékony lehet a XXI. században is.

Mielőtt a konkrét jogvitát ismertetnénk, szükséges kitérni magára a *csatlakozófelületre*. A csatlakozófelület a magyar és a közösségi szabályozásban csak említés szintjén szerepel. Hazai szerzői jogi törvényünkben a csatlakozófelület egyetlen helyen jelenik meg,² eszerint a csatlakozófelület alapját képező elv, ötlet nem lehet tárgy szerzői jogi védelemnek. Ez összhangban áll a 2009/24/EK irányelvvel,³ amely irányelv egyben tisztázza is a „csatlakozási felületek” fogalmát:

„A számítógépi program funkciója abban áll, hogy a számítógépes rendszer többi elemével és a felhasználókkal kapcsolatba lépjen és azokkal együtt működjön[...] ... A programnak a szoftver- és hardverelemek közötti *efféle összekapcsolódást és kölcsönhatást biztosító részei általában*⁴ *'csatlakozási felületek'* (interfész) néven ismertek.” [2009/24/EK (10), kiemelés a szerzőtől.]

Az irányelv rögzíti továbbá, hogy

¹ TRIPS, WIPO.

² 58. § (1) Az 1. § (6) bekezdésében foglalt rendelkezést alkalmazni kell a szoftver csatlakozó felületének alapját képező ötletre, elvre, elgondolásra, eljárásra, működési módszerre vagy matematikai műveletre is.

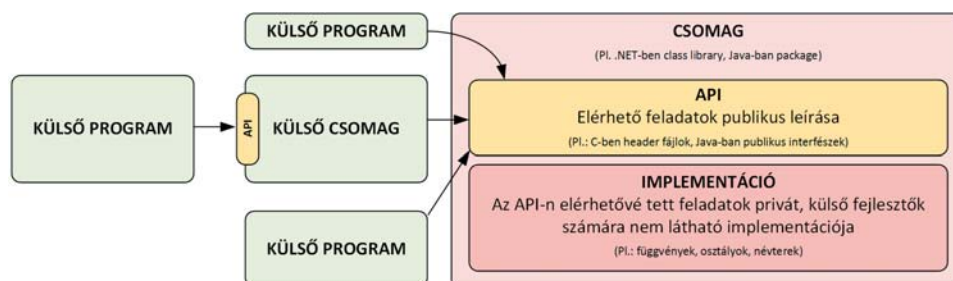
³ <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:111:0016:0022:HU:PDF>.

⁴ A jogalkotó az „általában” kitételrel minden bizonnyal azt akarta elkerülni, hogy a más terminus technicus alá eső, de hasonló feladatot ellátó szoftverrészek kikerülhessenek a szabályozás alól (pl. public facade).

„... csak a számítógépi program kifejezési forma részesül védelemben, és hogy a program valamely elemének – beleértve a csatlakozási felületet – alapjául szolgáló ötletek és elvek ezen irányelv értelmében nem állnak szerzői jogi védelem alatt. E szerzői jogi alapelvnek megfelelően, *amennyiben ezek az ötletek és elvek a logika, az algoritmusok és a programnyelv alapjául szolgálnak*, úgy ezen irányelv értelmében *nem állnak védelem alatt.*” [2009/24/EK (11), kiemelés a szerzőtől.]

Az, hogy a programnyelv alapját képező elvek nem állhatnak szerzői jogi védelem alatt, sőt, hogy esetleg maga a programnyelv önmagában sem állhat ilyen oltalom alatt (és bár a közvélekedés⁵ szerint a WPL v. SAS-ügyben⁶ a brit High Court ezen oltalom meglétéről is döntött, e sorok szerzője ezzel nem ért egyet⁷), nem érinti a jogvita tárgyát. A Java API, bár a Java programnyelven megírt forráskódi programban használjuk, nem része a Java programnyelvnek,⁸ azt kizárólag néhány tucat kulcsszó, szintaxis és szemantika alkotja.⁹

De mi is az API, az *application programming interface* (alkalmazásprogramozási felület)? Ezek az esetek túlnyomó többségében és a konkrét ügyben is olyan nyilvános szoftveres csatlakozófelületek, amelyeket kizárólag másik szoftver érhet el abból a célból, hogy az API „mögött” megvalósított (implementált) funkcionalitást kihasználja. Első közelítésben érdemes az API-kra úgy tekinteni, mint csomagok felirataira. Pontosan leírják, hogy egy-egy csomagban mi van, címkeként részletezik a csomagokban rendelkezésre álló funkcionalitást (például „Töröld le a merevlemeztől a paraméterként kapott elérési úton található fájlt!”).



Szoftvermodul csomagjának felépítése és a funkcionalitását elérhetővé tevő API-k elhelyezkedése, lehetséges megjelenési formájukkal példaként

⁵ <http://www.wired.co.uk/news/archive/2012-05/03/programming-language-copyright>.

⁶ SAS v. WPL-ítélet: <http://teamwpc.co.uk/docs/SAS-v-WPL-Final-Judgment.pdf>.

⁷ A jogvitának nem képezte az a kérdés a tárgyát, hogy a programozási nyelv mint önálló mű vajon jogosult-e a szerzői jogi védelemre, lásd ugyanezen ítélet (22).

⁸ A szintaxis leírását lásd: http://en.wikipedia.org/wiki/Java_syntax.

⁹ http://en.wikipedia.org/wiki/Programming_language.

Miért fontosak az API-k? Mint az előbbieken szó volt róla, a programozási nyelv önmagában néhány kulcsszó és egy nyelvtan, hasonlóan egy természetes nyelvhez. Az API az, amelyen keresztül az adott programnyelven írt programok elérhetik az előre elkészített („instant”) programkódokat, az előre gyártott funkciókat, függvényeket, ezek a funkciók és függvények pedig újabb API-kon keresztül vehetnek igénybe újabb, mélyebb rétegekben futó funkciókat. Az előre legyártott és a fejlesztők rendelkezésére bocsátott csomagok azok, amelyek megkímélik a fejlesztőket egyes rutinok és funkcionálisok újra- és újraismételt felesleges megírásától, és egyben megkímélik a felhasználókat a diverzifikáció okozta károsztól a számítógépükön. Az API-k azok a „ragasztófelületek”, amelyek a sok kisebb-nagyobb építőközből a számítógépes szoftverek összetett rendszerének egységes működését biztosítják. Legyen ez ablak megjelenítése egy operációs rendszerben vagy fájl felöltése az Amazon Cloudba, minden esetben nyilvános API mondja meg a hívóprogramnak azt, hogy *milyen* szolgáltatásokat kínál a számára.

Hangsúly van a *milyenen*: az API sosem árulja el, *hogyan* oldja meg a feladatot, hanem kizárólag azt, hogy milyen bemenő adatokat használ egy-egy művelethez, és ennek eredményeképpen milyen kimenő adatokat ad vissza. Ha úgy tetszik, az API a vezérlőpult, a mögöttes működés azonban a hívó fél számára fekete doboz, kizárólag a bemenet és a kimenet ismert. Olyan ez a programok szempontjából, mint a hivatali ügyintézés az ügyfél szempontjából. Tudja, hogy hova, milyen formanyomtatványon mit adhat be (input), és ezért milyen eredményt vagy eredményeket várhat (output). A belső működés többnyire ismeretlen, és nem is szükséges, hogy ismert legyen (sőt, ennek megismerése a szerzők számára kifejezetten ártalmas lehetne).

A félreértések elkerülése érdekében megjegyezzük, hogy amikor szoftverek közötti csatlakozófelületről beszélünk, nem soroljuk ide a fizikai eszközök (akár szabadalommal védett) csatlakozófelületeit (pl. CPU), és ebben a cikkben a szoftverszabadalmak – közösségi jogban amúgy is csak nagyon szűken elismert – témáját sem érintjük.

Fontos azonban aláhúzni, hogy ha az olvasó úgy érzi, az API felépítése, működése és célja nagyban hasonlít az általában szabadalmakkal oltalmazott megoldásokéhoz, akkor nem téved. A számítógépi programok mindig valamilyen feladat ellátására jönnek létre, mindig funkcionálisak, és éppen ez a tulajdonságuk az, amely miatt kicsit kakukktojások a szerzői jogban, és visszatérően sugallják a szabadalmi védelem lehetőségét. Az, hogy az szoftverek csatlakozófelületei – vagy akár a szoftverek más részei, kifejeződései formái – jogosultak lennének, lehetnének-e szabadalmi oltalomra, már egy másik cikk témája lehetne.

MÉLYVÍZ, AZ API MEGJELENÉSI FORMÁI

Reméljük, keveseket rettentünk el a következő fejezettel, amelyben valós, forráskódban megjelenő API-építésbe kezdünk. Ha a kedves olvasót inkább a jogi háttér további részletei érdeklik, ugorjon bátran a *Oracle v. Google-jogvita* fejezetre.

A kitartó olvasók egy dokumentumokat kezelő szoftvermodul API-ját ismerhetik meg. Cél, hogy dokumentumokat tudjunk menteni, kiolvasni és szűrőfeltételek szerint listázni, és ezt a dokumentumkezelő csomagot más szoftverfejlesztő cégeknek kívánjuk értékesíteni, hogy ne kelljen saját rendszereikhez ezt a funkcionalitást újra megírniuk. Ez az egyszerű dokumentumkezelési funkcionalitás – amely önmagában természetesen érdemtelen a szerzői jogi oltalomra –, mint látni fogjuk, szinte végtelen számú API-t eredményezhet.¹⁰

Első változat

API-k két helyről ismerhetőek meg: *a)* dokumentációból (amely a szoftverfogalom részeként szintén oltalmat élvez¹¹) és *b)* magából a forráskódból, ha az utóbbi rendelkezésre áll. Mindkettő az API megjelenési formájának tekinthető, teljesen identikus. Eltérés annyiban van, hogy amíg a dokumentáció egyértelműen a fejlesztőknek szól, a forráskódi API már mind a számítógépi használatot, mind a hozzáértőbb szoftverfejlesztők tájékozódását is szolgálhatja.

A fenti írt funkcionalitás megvalósítása történhet az alábbi elnevezésekkel és hierarchiában:

```
namespace Documents;

interface DocumentManager
{
    Document GetDocument(string filename);
    SaveDocument(Document document);
    List<Document> SearchDocuments(Filter filter);
}
```

Mindez annyit jelent, hogy a kért funkcionalitás a *Documents* névtérben (mint gyűjtőben) elhelyezett *DocumentManager* interfészen keresztül e három függvénnyel (*GetDocument*, *SaveDocument*, *SearchDocuments*) érhető el az API-n keresztül.¹² Paraméterként fájlnevet (*string filename*), egy konkrét dokumentumot (*Document document*) vagy szűrőt kaphatnak

¹⁰ Kérdés lehet, hogy minek ide API, hiszen a szoftver felhasználója csak megnyomja a felhasználói felület megfelelő gombjait, és meg is történhetne az elvárt eredmény (például a fájl törlése vagy zene lejátszása). Általános gyakorlat – és az így van a Java esetében is – hogy a szoftverek nem monolitok, hanem sok-sok modulból épülnek fel. A modulok közötti API az a moduloknak, ami a felhasználó számára a felhasználói felület: míg a felhasználó képernyőn keresztül párhuzamosan Excelben, Wordben, levelezőalkalmazásban dolgozik, és maga vezérli a munkát, a szoftvermodulok az API-n keresztül ismerik meg és használják, vezérlik egymást. Hasonlattanálva: ami a felhasználónak a képernyőn megjelenő felület, az a szoftverrendszerek számára egymás között az API.

¹¹ Sztj. 1. § c).

¹² Az implementálás, vagyis az elvárt funkcionalitás megvalósítása már sokkal több forráskódi program megírását igényelné, célunk azonban most kizárólag az API építésének a bemutatása.

(*Filter filter*). A *GetDocument* egy konkrét fájlt, a *SearchDocuments* pedig a szűrőfeltételnek megfelelő dokumentumok listáját adja vissza.

Az API fejlődése, fejlesztése során – például titkosítás, felhőbeli tárolás bevezetésével – annak összetettsége hatványozottan növekszik, egyre több feladatot kell továbbra is átlátható, érthető módon az API használói számára elérhetővé tenni.

Második változat

Úgy tűnhet, mindez magától értetődő, és nincs rá ok, hogy ezt bárki bármikor másképp csinálja. A következő, szintén valós példában látható lesz azonban, hogy mindez könnyedén lehetséges:

```
namespace Enterprise.Store.Documents;  
  
interface DocumentQuery  
{  
    List<Document> QueryBy(Filter filter);  
}  
  
interface DocumentStore  
{  
    Document Find(int fileId);  
    int Save(Document document);  
}
```

A fenti, sokkal jobban tagolt API két diszjunkt interfészen keresztül biztosítja a hozzáférést a már ismert funkcionalitáshoz: egyrészt a kereséssel (*DocumentQuery*) és a mentéssel, beolvasással (*DocumentStore*) egy az előzőtől egészen eltérő névtérben (*Enterprise.Store.Documents*). Az olvasót ez a két példa talán meggyőzi két fő állításunkról: egyrészt aki az első változatot ismeri, az nem fog azonnal kiigazodni a másodikon és fordítva. Másrészt, az API tervezése során azonos funkcionalitás gyökeresen eltérő módszerek szerint épített API-kon át is elérhető, megvalósítható.

Lehetséges lenne harmadik, negyedik, ötödik változat is. Ha azonban egy már létező API-val megegyezőt szeretnénk készíteni, akkor az eltérés nem lehetséges – és ez az állítás az Oracle v. Google-ügy kapcsán még fontos lesz.

AZ ORACLE V. GOOGLE-JOGVITA

Az ügy e cikkben elemzett tárgya az alábbi: állhat-e a (Java) API, pontosabban annak szerkezete, sorrendje és elrendezése (angol szakkifejezésként SSO) szerzői jogi oltalom alatt? Mielőtt azonban erre a meghozott ítélet vagy az uniós jogrend alapján választ keresnénk, röviden bemutatjuk a jogvitához vezető utat és annak hátterét.

Iparági háttér

Az Android felemelkedése

A ma már a mobiltelefonok és táblagépek meghatározó részén futó Android operációs rendszert a Google 2005-ban vásárolta meg és fejlesztette tovább. 2007 végén az Androidhoz megjelent fejlesztői készlet (SDK) a Java SE API egy részét implementálva adott lehetőséget a fejlesztőknek androidos programok készítésére, anélkül, hogy ehhez az Oracle (amely cég a Java feletti jogokat addigra a Suntól megszerezte) hozzájárult volna. A Google és az Oracle ezt követően hosszas és eredménytelen tárgyalások során próbálta rendezni a Java platform egy részének Androidon való megjelenését.

Egy biztos, a választás a Google-t igazolta. Ma egy mobiltelefontól vagy egy táblagéptől messze többet várnak el a felhasználók, mint a telefonálás vagy a könyvolvasás, a sikerhez elengedhetetlen a nagy mennyiségű és jó minőségű programokkal feltöltött szoftverbolt, ehhez pedig sok és jó fejlesztő szükséges. Fejlesztői közösséget a nulláról felépíteni azonban sok év kemény és drága munkája,¹³ a Google nem véletlenül tette le a garast a Java (pontosabban a Java egy része, 37 csomagjának API-ja) mellett. A korábban Javán nevelkedett és azt jól ismerő fejlesztők milliói (!)¹⁴ az Android világában is azonnal otthonosan érezhették magukat, nagyon meredek tanulási görbével kezdhették el a mobilalkalmazások fejlesztését.

A Google 2014. év végi adatai szerint ma az Androidnak 1 milliárd aktív felhasználója van. Ennek a sikertörténetnek nem egyedüli, de fontos része volt a fejlesztők megnyerése, és ehhez a Java API-n keresztül kényelmes út vezetett.

A Java API

A Java API az, amelyen keresztül a Java programnyelv használói az Oracle (illetve később a Google) által biztosított és előre megírt funkciókat (matematikai függvényeket, hardverelérést, szövegműveleteket stb.) elérhetik. Hosszú éveken át fejlesztett, széles körben ismert, hozzáértéssel szervezett-szerkesztett rendszerről van szó, amelynek a kiterjedtsége a Java ökoszisztéma egyik nagy erőssége.

A jogvita tárgya a Java 37 csomagjának API-ja, megközelítőleg 7000 sor forráskód. És bár mindez elenyésző (3%) a teljes implementációhoz képest, ne felejtjük el, hogy ebben az esetben kizárólag a két konkrét API-t, az Oracle- és a Google-féle Java API 7000-7000 sorát kell összevetnünk szerzői jogi jogsértés szempontjából.

¹³ Lásd például a Microsoft .NET fejlesztői platform népszerűvé tételéhez szükséges többéves erőfeszítéseket.

¹⁴ <http://jaxenter.com/how-many-java-developers-are-there-10462.html>.

A PER

A sikertelen egyeztetések után az Oracle 2010 augusztusában beperelte a Google-t a Javát érintő szerzői jogi és szabadalmi jogsértések miatt az USA bírósága előtt. Jelen cikkben kizárólag szerzői jogi nézőpontból tekintünk a jogvitára, így sem a fair use, sem az interoperabilitási¹⁵ kérdésekre nem térünk ki, mert mindez nem érinti azt, hogy az API jogosult-e a szerzői jogi oltalomra, vagy sem.

Elsőfokú ítélet

Első fokon William Alsup bíró tárgyalta az ügyet, aki a helyes döntés meghozatala érdekében tett erőfeszítéseiért – megtanult például Javában programozni – nagy népszerűségre tett szert az IT-s közösségekben. Döntése lényegében az Oracle teljes vereségét jelentette.

Ítéletében¹⁶ a 37 érintett Java API-t *ebben* ez ügyben¹⁷ szerzői jogi védelemre nem tartotta jogosultnak. Ítélete az alábbiakban foglalható össze:¹⁸

- amíg az implementációs (a funkcionalitás *mikéntjét* megvalósító) kód eltérő, addig bárki jogosult ugyanazt a funkcionalitást megvalósítani és deklarálni (vagyis ahhoz API-t létrehozni) a Copyright Act rendelkezései szerint. Nem számít, hogy az API deklarációs forráskódja megegyezik az eredeti példányéval,
- a Java szabályai szerint ezektől a deklarációktól eltérni nem is lehet, és mivel az eltérés nem lehetséges, ezért ezt az egyetlen létező kifejezési módot senki sem sajátíthatja ki,¹⁹
- és bár az Android saját API-ja használhatna saját nevezéktant, a szerzői jog ilyen rövid kifejezésekre és szavakra nem terjedhet ki, továbbá
- a parancsok és azok hierarchiájának azonos felépítése szükséges az interoperabilitás megtartása érdekében.

Ítéletének végén Alsup bíró egyértelművé tette továbbá, hogy

„A döntés nem arról szól, hogy a Java API csomagjai mindenki számára szabadon felhasználhatóak lennének licencia nélkül, sem arról, hogy a szerkezete, sorrendje és elrendezése bármely számítógépi programnak szabadon átvehető

¹⁵ Az interoperabilitás, vagyis a rendszerek együttműködése valóban lényeges kérdése a számítástechnikának (ennek fontos része az API is), de érdemes tudni, hogy az androidos Java csupán „részlegesen interoperabilis” az Oracle Javával.

¹⁶ <http://docs.justia.com/cases/federal/district-courts/california/candce/3:2010cv03561/231846/1202>.

¹⁷ William Alsup bíró ítéletében külön ki is hangsúlyozza: „This order does not hold that Java API packages are free for all to use without license”: <http://www.groklaw.net/pdf3/OraGoogle-1202.pdf>, p. 41, Conclusion.

¹⁸ I. m. (17), p. 3, Summary of ruling.

¹⁹ A Merger Doctrine alapján amennyiben az ötlet és a kifejezés mód annyira szoros kapcsolatban van, hogy maga az ötlet és annak egyetlen megjelenési formája egy és ugyanaz, akkor a szerzői jog nem védi a megjelenési formát, mert nem védheti az ötletet.

lenne. A döntés erre a kifejezett esetre vonatkozik, a Google által többszörözött részek mindenki más számára is szabadon felhasználhatóak lennének a Copyright Act alapján.” (P. 41.)

Döntését nagy lelkesedés²⁰ és kritikák²¹ kísérték. Az Oracle természetesen fellebbezett, és az ennek eredményeként született ítéletet hasonlóan heves kommentárok kísérték.

Másodfokú ítélet

A másodfokú bíróság (Federal Circuit) ugyanis lényegében az elsőfokú ítélet minden érdemi döntését megváltoztatta, ráadásul 180 fokkal eltérő irányba. A döntésben a bíróság mély, a szoftverfejlesztést értő, azt a jog nyelvére kiválóan szintetizáló hozzáértése tükröződik.

Igazodva az elsőfokú ítélet fenti ismertetéséhez, hasonló felosztásban mutatjuk be a másodfok döntését²² is:

- a döntés egyértelművé tette a Java *programnyelv* és a Java *API* közötti különbséget. A programnyelv bárki számára szabadon felhasználható, azonban a Java programnyelven írt Java API (és maguk a csomagok) nem;
- a Google számára adott volt a lehetőség, hogy az ugyanolyan funkcionalitást tartalmazó csomagokat (és a „tetejükön levő” Java API-t) a Javától eltérő módon építse fel, ám a Google ezt nem tette meg;
- egy számítógépi program szükségszerűen funkcionális, lényege, hogy valamilyen feladatot ellásson. Ha elfogadnánk, hogy a funkcionalitása miatt nem élvezhet szerzői jogi védelmet a program, akkor egyetlen számítógépi program sem lenne rá jogosult;
- önmagukban szavak, rövid kifejezések valóban nem jogosultak oltalomra. Azonban *Charles Dickens Két város regénye* c. műve sem más, mint szavak sorozata, viszont senki sem állítaná, hogy ez a mű nem állhat szerzői jogi védelem alatt csak azért, mert lebontható szavakra és rövid kifejezésekre; továbbá
- az Androidban levő Java *nem* interoperabilis az Oracle Java változatával, ráadásul az interoperabilitás irreleváns a szerzői jogi védelem szempontjából.

A felek egyetértettek továbbá abban,²³ hogy a Java API megfelel a Copyright Act²⁴ 102(a) rendelkezéseinek, tehát főszabályként jogosult a szerzői jogi védelemre. Nem értettek egyet azonban azt illetően, hogy a 102(b) hogyan alkalmazandó. A Google álláspontja szerint míg a 102(a) megadja az oltalmat, addig a 102(b) miatt egyúttal el is veszi azt, hiszen a számítógépi programalkotás mint mű funkcionális elemeket hordoz. Azonban – mutat rá a bíróság – a Kongresszus kiemelte, hogy a 102(b)-nek „nem célja, hogy növelje vagy csökkent-

²⁰ <https://www.eff.org/deeplinks/2012/05/no-copyrights-apis-judge-defends-interoperability-and-innovation>.

²¹ <http://www.fosspatents.com/2012/05/judge-says-google-only-used.html>.

²² <http://cafc.uscourts.gov/images/stories/opinions-orders/13-1021.Opinion.5-7-2014.1.PDF>.

²³ I. m. (22), p. 22.

²⁴ 17 U.S. Code § 102: <http://www.law.cornell.edu/uscode/text/17/102>.

se a szerzői jogi védelem terjedelmét, hanem hogy az ötlet és a kifejezés közötti dichotómiát megtartsa”. Ez a kettősség a Java API érintett 37 csomagja esetében pedig fennáll.

A másodfok lényegében teljes fordulatot jelent az elsőfokú ítélethez képest. A döntés ki mondta, hogy az érintett 37 Java csomag API-jának szerkezete, sorrendje és elrendezése²⁵ szerzői jogi védelemre jogosult.

A döntés, akárcsak az első, szintén megosztotta mind a jogász-, mind az IT-szakmát. Egyfelől a szerzői jogi védelem lehetősége sok évtizedes szokásjogot változtatna meg, nagyban növelné a számítástechnikán belüli divergenciát, és gátolhatná a folyamatos, szerves fejlődést. Másrészt azonban az API megalkotása sokszor egy-egy cég legjobb mérnökeinek többhavi vagy -éves alkotómunkájának egyéni-eredeti jellegű eredménye.

AMERIKAI DÖNTÉS, EURÓPAI KÖVETKEZTETÉSEK

Az Európai Unió Bírósága előtt a SAS v. WPL-ügyben Yves Bot főtanácsnok utalt indítványában²⁶ az egyedi elrendezés és felépítés oltalomra érdemes jellegére:

„A programozó képleteket és algoritmusokat használ, melyek önmagukban nem állnak szerzői jogi védelem alatt, hiszen a költő vagy regényíró által az irodalmi alkotáshoz felhasznált szavakhoz hasonlók. Azonban mindezen *elemek elrendezési módja*, mint például a számítógépi program írásmódjának stílusa, *tükrözheti a szerző saját szellemi alkotását, és így védelem tárgya lehet.*” (C-406/10, 55., kiemelés a szerzőtől.)

Ha két több tíz- vagy százezer soros forráskódi program azonos, akkor a másolás – és így felhasználási engedélyek hiányában a szerzői jogi jogsértés – nagyon erősen valószínűsíthető. Azonban a gyakorlatban eszközök tucatjai²⁷ állnak rendelkezésre a forráskód (vagy a tárgykódból visszafejtett forráskód) átalakítására és lefordítására vagy szolgai másolással egyenértékű újírására, így a szinte csupán pusztán szó szerinti vizsgálat nem nyújthat elégséges védelmet a szerzői jogukban megsértett szerzőknek. Ezért amellet, hogy a főtanácsnoki vélemény illeszkedik a Oracle v. Google-ügyben hozott ítélet SSO-ra vonatkozó részéhez – hogy egy rövid kitérőt tegyünk az általános szoftverrel kapcsolatos szerzői jogi

²⁵ Structure, sequence and organization (SSO).

²⁶ C-406/10, 55. bekezdés: <http://curia.europa.eu/juris/document/document.jsf?jsessionid=9ea7d2dc30dd8f3f2a601b1d45c99305f0e8b221aefc.e34KaxiLc3qMb40Rch0SaxuPahz0?text=&docid=115484&pageInde x=0&doclang=HU&mode=lst&dir=&occ=first&part=1&cid=228098>.

²⁷ Legyen az akár a forráskód automata és identikus megváltoztatása [pl. ProGuard (<http://proguard.sourceforge.net/>)] vagy a tárgykódból való visszafejtése és újbóli lefordítása [DCompiler (<http://sourceforge.net/projects/dcompiler/>)], amelyek eredményeképpen a felhasználói funkcionalításban az eredeti kóddal teljesen azonos program jön létre más forráskódi megjelenésben.

esetekhez – ez ha kis lépésekben is, de jelentős elmozdulást jelenthetne a szerzői jogok sérelme esetén a forráskódok gyakorlatilag teljes egyezőségét elváró küszöbhez képest.

Célszerű lenne ezért a szoftver oltalomra jogosult aspektusainak – így például az API-nak – a szerzői jog számára jobban kézzelfogható, már ismert megközelítéseket találni. Az Oracle enyhén vitriolos *Ann Droid*-hasonlata²⁸ – amelyben a Java API lemásolását a *Harry Potter* regény fejezetcímeinek másolásával írt könyv kiadásához hasonlították – könnyen érthető, ám ha az API-hoz mint *gyűjteményes műhöz* közelítünk, jogászként szintén ismerős terepen találjuk magunkat.

API-k mint gyűjteményes művek

Mint a fentiekből látható, egy API alkotóelemei önmagukban nem jogosultak szerzői jogi védelemre: rövid, többnyire néhány szóból álló összetett függvénynevek és paraméterek, komoly szellemi munkával rendszerre szervezve azonban jól átlátható, a fejlesztők által könnyen kiismerhető és használható rendszerre állnak össze. Azt is láthattuk, hogy az oltalom kérdéssége nem az API egy-egy elemére vonatkozik, hanem azok egyedileg elrendezett, szerkesztett egészére.

A gyűjteményes művek (mint például a lexikonok, sajátos szerkesztési elvű jogszabálygyűjtemények vagy akár telefonkönyvek) ebből a szempontból nagyon hasonlóak az API-khoz: a művet alkotó elemek nem feltétlenül részesülnek szerzői jogi védelemben (mint például a „kutyá” szó), azonban a tartalom elrendezésének és szerkesztésének egyéni és eredeti jellege már oltalomképesé teszi a művet.

Az Szjt.²⁹ a gyűjteményes művekről az alábbi módon rendelkezik a 7. §-ban:

- „(1) Szerzői jogi védelemben részesül a gyűjtemény, ha *tartalmának összeválogatása, elrendezése vagy szerkesztése egyéni, eredeti* jellegű (gyűjteményes mű). A védelem a gyűjteményes művet *megilleti* akkor is, ha annak *részei, tartalmi elemei nem részesülnek, illetve nem részesülhetnek szerzői jogi védelemben.*
- (2) A gyűjteményes mű egészére a szerzői jog a szerkesztőt illeti, ez azonban nem érinti a gyűjteménybe felvett egyes művek szerzőinek és a kapcsolódó jogi teljesítmények jogosultjainak önálló jogait.
- (3) A gyűjteményes mű szerzői jogi védelme nem terjed ki a gyűjteményes mű tartalmi elemeire.” (Kiemelés a szerzőtől.)

Az API-k alkotóit hasonló elv vezeti, mint egy lexikon vagy szótár szerzőjét. A cél a keresett elem (lexikon esetében egy szó és a hozzá tartozó jelentés, míg API esetében például egy függvény) minél gyorsabb és pontos megtalálása. Az erre szolgáló elv természetesen

²⁸ <http://www.groklaw.net/pdf4/OraGoogleAppeal-43.pdf>.

²⁹ http://net.jogtar.hu/jr/gen/hjegy_doc.cgi?docid=99900076.TV.

nem részesülhet szerzői jogi védelemben, azonban ahogyan ez az elgondolás formát ölt, már jogosult lehet erre.

Ha egy regény fejezeteihez hasonlítjuk az API-t – mint tette ezt az Oracle az *Ann Droid*-hasonlatában – akkor ezt az amúgy lényegi kérdést elhanyagoljuk: az API egyik fontos tulajdonsága a szerkezet, az elemek elrendezése. Amikor az API-t használó szoftverfejlesztőknek több száz vagy több ezer lehetséges eljárás, függvény közül kell megtalálniuk a szükségeset, akkor még a lexikonok olvasóinál is nehezebb helyzetben vannak, ugyanis az ábécé szerinti rendezés mit sem ér. Az API-n belül az általánostól a konkrét felé haladva kell megtalálni a keresett függvényeket, névtérről névtérré haladva, valamilyen, az API szerzője által kitalált rendszer mentén.

A hazai jogtudományi álláspont is illeszkedik ehhez³⁰ a gondolatmenethez. Eszerint a gyűjteményes mű szerkesztőjét, amennyiben az elrendezés, szerkesztés, összeválogatás eredménye sajátos, egyéni-eredeti jelleget tükröz, megilleti a szerzői jogi védelem. Megilleti ez akkor is, ha a gyűjteménybe került elemeket önmagukban amúgy nem illeti meg szerzői jogi oltalom. Mindez hasonló a szótár szerzőjének, szerkesztőjének helyzetéhez. A szócikkek szerkezetének kialakítása és a gyűjtőmunka több lehet, mint egyszerű rutintevékenység. Az API szerzője esetében az egyedi függvény- és metódusnevek elnevezése, az egységes nevezéktan kialakítása, ezek osztályokba és hierarchikus szerkezetbe (névterekbe) szervezése szintén komoly szellemi munkát igénylő, alkotó folyamat. Az, hogy nem más, hanem akár saját maga által alkotott elemekből hozza létre az API-t, szintén belefér a gyűjteményes művek definíciójába.

Felvethető – mint tette a Google is a konkrét ügyben –, hogy az API csak és kizárólag egyetlen módon felépíthető, és ebből adódóan nem jogosult a szerzői jogi védelemre. A másodfokú bíróság helyesen ítélte meg, hogy azonos funkcionalitás tetszőlegesen sok API-n keresztül is elérhetővé tehető, ezt mutattuk be mi is korábban dokumentumkezelő rendszerünk API-ján keresztül. A Google lehetősége a Java *programnyelv* választása mellett is megvolt arra, hogy a Java keretrendszerben már létező csomagjai által megvalósított funkcionalitást más elrendezésben, más nevezéktannal, más implementációban, tehát más API-n keresztül tegye elérhetővé a leendő androidos fejlesztők számára.

A gyűjteményes művek megfoghatóvá tehetik a jogászok számára az API-k (de tágabb értelemben a csatlakozófelületek) szerzői jogi védelemre érdemes aspektusait, biztosítva az alkotók által végzett egyéni és eredeti jellegű, komoly és sokszor évekig tartó szellemi tevékenységet igénylő alkotómunka jogi védelmét.

³⁰ Lontai Endre, Faludi Gábor, Gyertyánfy Péter, Vékás Gusztáv: Magyar polgári jog. Szerzői jog és iparjogvédelem. Eötvös József Könyvkiadó Bt., 2012, p. 59.

Az API-k és a hazai joggyakorlat

A gyűjteményes művek oltalma a hazai joggyakorlatban is többször megerősítést nyert. A döntések alapján kijelenthető, hogy az API-k létrehozásának jellemzői (többszerzős művek, egyéni szerkesztés elismerése az önmagukban erre nem jogosult elemekkel együtt, adatbázis és gyűjteményes művek elválasztása) ismertek bíróságaink számára.

A Fővárosi Ítéltábla jogerős ítéletében³¹ határozott arról, hogy több szerző is lehet a mű szerzője, és ekként minősítette többszerzős gyűjteményes műnek egy csapatépítő tréningeket készítő cég ún. „Tematika és emlékeztető” anyagát. Szakvélemény alapján kimondta továbbá, hogy a

„... perbeli team-építő tréning Emlékeztető és Tematika egységes mű, és az Sztj. 7. §-ának (1) bekezdése szerinti *gyűjteményes műnek minősül, mert elrendezése, szerkesztése egyéni, eredeti jelleget mutat*. Részai pedig részben más szerzőktől származnak, részben nem állnak szerzői védelem alatt...” (Kiemelés a szerzőtől.)

Egy másik, a Pécsi Ítéltábla által hozott határozat³² egy interjúkötetnek biztosított annak egyéni, eredeti szerkesztési jellege alapján szerzői jogi védelmet az Sztj. 7. §-ának (1) bekezdése alapján, és kimondta, hogy a szerzői jog annak szerkesztőjét illeti az Sztj. 7. §-ának (2) bekezdése alapján.

Szintén a Fővárosi Ítéltábla jogerős ítéletéről hozott döntésében a Legfelsőbb Bíróság mondta ki,³³ hogy egy műsorújságban a műholdas vételi lehetőségekről rendszeresen tájékoztató ún. frekvenciatáblázatok és áttekintő térképek

„a szerzői jogról szóló 1999. évi LXXVI. törvény (Sztj.) 7. § alapján *gyűjteményes műként szerzői jogi védelemben részesülnek*, amelynek jogosultja az első szerkesztő” (kiemelés a szerzőtől)

mégpedig azok egyedi szerkesztési jellege miatt. Ez az egy egyéni-egyedi szerkesztési jelleg, amely az API-knál is lényegi mozzanat az alkotás folyamatában. Egyszerű, akár önmagukban szerzői jogi oltalomra nem érdemes elemekből épül fel az API saját szerkesztési, rendezési elvek szerint, sőt, mint láttuk már önmagának a nevezéktannak (névtérre, függvénynévre stb.) is lehet egyéni jellege.

A Fővárosi Ítéltábla ugyanitt megállapította, hogy a felperes által rögzített adatok szerzői jogi védelme nem áll meg, és azt sem fogadta el, hogy az áttekintő térkép és a frekvenciatáb-

³¹ Pf.21.671/2011/4.

³² Pf.III.20.168/2009/4.

³³ Pfv.IV.21.208/2008/5.

lázat külön jogi védelem alatt álló adatbázis lenne. Itt jegyezzük meg, hogy az egyéni-eredeti elvek alapján létrejött adatbázissal mint a gyűjteményes művek különleges alcsoportjával nem itt foglalkozunk, mert az jellegéből adódóan más mű, mint az API. Ugyanis míg az adatbázis olyan gyűjtemény, amelynek *adataihoz*, elemeihez külön-külön is hozzá lehet férni, addig az API lényegét tekintve *műveletek* rendezett halmaza.

ZÁRSZÓ ÉS KOGNITÍV DISSZONANCIA

Amennyire megalapozottnak tűnik az erre érdemes API-k szerzői jogi védelme a szerzői jog szempontjából, legalább annyira ellentmondásos a helyzet a gyakorló programtervező matematikus perspektívájából. Az API-k a mai napig rendszerek közötti együttműködések alapjai, és kevés kivételtől eltekintve a szoftveripar szokásjoga alapján ezt eddig senki nem vonta kétségbe. És bár több nagyvállalat³⁴ maga is elkötelezett az API-k külön licenzelés nélküli felhasználása mellett, féltő, hogy precedensértékű ítélet esetén az olykor visszaélés-szerű szabadalmi perek mintájára hasonlóra számíthat több iparági szereplő szerzői jogi területen is. A perekből való félelem már önmagában is visszatartó erő lehet, az API-k diverzifikációja pedig mind fejlesztői, mind felhasználói oldalon jelentős többletterheket jelentene. Az, hogy ezt a helyzetet a fair use vagy az interoperabilitás, esetleg más jogi eszközök oldják fel akár az USA-ban, akár az EU-ban, ma még nem látható.

Másrészt – és mint több nagy szoftverprojekt architektje ezt e sorok szerzője maga is átélte –, egy-egy triviálisnál összetettebb API szervezése, megtervezése, megírása komoly szellemi munka eredménye, így az alkotók hatékony jogi védelem nélkül hagyása méltánytalan lenne. Mindazonáltal biztató, ahogy a szerzői jog a szoftverekre rezonál. Egyre több döntés érinti a számítógépi programokat, és válik szét az eddig a jogalkotók és jogalkalmazók számára néha talán monolit szoftverfogalom egyre több és apróbb, önmagában vizsgálható és a jog számára is megfogható részre.

³⁴ <http://www.developerfusion.com/news/85355/microsoft-says-net-on-android-is-safe-no-litigation-like-oracle/>.