

# Úton a szakköralkalmazás felé a ProgCont API-val

Kádek Tamás<sup>1</sup>, Biró Piroska<sup>2</sup>

{<sup>1</sup>kadek.tamas, <sup>2</sup>biro.piroska}@inf.unideb.hu  
DE IK

**Absztrakt.** A ProgCont automatikus megoldáskiértékelő rendszert a Debreceni Egyetem Informatikai Karán már számos különböző számonkérésen, helyi, országos és nemzetközi versenyen alkalmaztuk sikeresen. A pozitív tapasztalatainkból kiindulva úgy gondoljuk, hogy a rendszert érdemes szélesebb körben elérhetővé tenni. Ez adta az ötletét az általunk újonnan kifejlesztett ProgCont API szolgáltatásnak, melyet elsősorban a középiskolában tanító pedagógusok számára kívánunk felajánlani. Az API felhasználásának első kísérleteként egy szakkörököt támogató alkalmazás fejlesztése mellett döntöttünk. Cikkünkben az API-ban és a segítségével készülő alkalmazásban lévő lehetőségeket szeretnénk bemutatni.

**Kulcsszavak:** ProgCont rendszer, automatizált kiértékelés, programozás, szakköralkalmazás

## 1. Bevezetés

A ProgCont rendszer, mely egy automatizált megoldáskiértékelő rendszer, 2011 óta használjuk és fejlesztjük a Debreceni Egyetem Informatikai Karán [1], [2], [3], elsősorban programozási feladatok kiértékelésére, számonkérések és programozási versenyek lebonyolítására, támogatására készült.

A legújabb fejlesztési irányunk a ProgCont API fejlesztése és az ehhez kapcsolódó szakköralkalmazáshoz tartozó kliens program megtervezése és implementálása. Ezt a fejlesztési irányt a GDPR megjelenése és a GDPR által támasztott követelményeknek való megfelelési kényszer szülte. A megfelelés leggyorsabb módja az volt, hogy egyszerűen eltávolítottunk a szoftverből minden személyes adatot, melynek eredményeképpen bejelentkezés nélkül vált használhatóvá a rendszer. Előtte csak a Debreceni Egyetem Informatikai Karának hallgatói tudták használni hálózati azonosítóval, és az általunk szervezett verseny résztvevői a számukra létrehozott felhasználói fiókkal. Ezután a korábban csupán egy zárt közösség számára elérhető alkalmazásunk nyitottá vált. Azóta előállt a helyi GDPR-ral kompatibilis szabályozásnak megfelelő verzió is, de szoftver regisztráció nélküli használható változatát a pozitív visszajelzések miatt megtartottuk.

Jelen pillanatban a ProgCont rendszert bárki szabadon használhatja az elérhető feladatok megoldásainak kiértékelésére. Az, hogy a rendszerünk regisztráció nélkül kiértékeli a korábban karunkon összeállított programozási feladatok megoldásait, még nem feltétlenül teszi csábítóvá a középiskolák számára. Annak érdekében, hogy érdembe bevonjuk a rendszer használatába a középiskolákban tanító tanárokat is, kialakítunk a ProgCont rendszerből egy, kifejezetten a szakkörök szervezését támogató alkalmazást is.

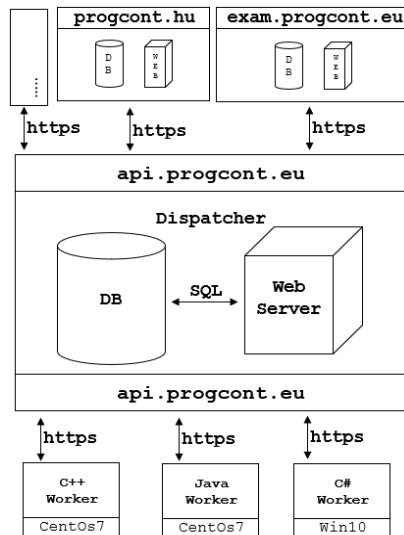
A ProgCont rendszert számos feladat megoldásra használjuk, ezek a feladatok nem jelentenek egyenletes terhelést a háttérben meglévő infrastruktúra számára. Például a nagy érdeklődést kiváltó programozási versenyek, illetve a különböző programozási tárgyak számonkérésének ideje alatt mutatkozó terhelés mellett eltérően az, ami hétköznapi használat során tapasztalható. A ritkán jelentkező, de intenzív terhelést jelentő időszakok közt más is hozzáférhetne és használhatná a rendelkezésre álló erőforrásokat akár saját szakkörök, számonkérések szervezése céljából. Köztudott, hogy a középiskolák infrastruktúrája változó, ezért számukra fejezetten hasznos lehet a ProgCont szabad kapacitásainak kihasználása. Célunk, hogy támogassuk a szakkör szervezésének lehetőségét azáltal, hogy biztosítjuk az alkalmazás használatát a középiskolában tanító kollegák számára.

## 2. ProgCont API

A fejlesztés alapvető eleme a ProgCont API. Ezzel létrehozunk egy alkalmazói interfészt, amelyen keresztül az automatikus megoldáskiértékelő rendszert hozzáférhetővé tettük. Ezután következhetett a meglévő felhasználói interfészek (webalkalmazások) átalakítása úgy, hogy az új API szolgáltatásain keresztül lássák el feladatukat. Ennek segítségével még házon belül tesztelhetjük az alkalmazói interfészt.

### 2.1. A ProgCont API felépítése

Az API egy online elérhető eszközt definiál, amely a programkódok automatikus fordításáért, futtatásáért és a kiértékeléséért felelős. Nem csupán a meglévő alkalmazásaink kaptak ezzel egységes háttérszolgáltatást, hanem megnyílt a lehetősége, hogy hasonló alapokon újabb funkciókkal bővüljön a ProgCont eszköztára, amelyek segítségével bármely érdeklődő – akár középiskolai informatika tanár – továbbfejlesztesse és saját elképzelései alapján testre szabhassa.



1. ábra: A ProgCont API rendszersémája [3]

A ProgCont API elérhető a következő link segítségével: <https://api.progcont.eu/>

Az API központi eleme a Dispatcher szolgáltatás, ez az a réteg, amellyel közvetlenül kapcsolatba kerül az, aki a ProgCont API-n fejlesztést akar végezni, ez tulajdonképpen a beérkező programkódok fordítás, futtatás feladatát elosztja azon worker-nek nevezett gépek között, amelyek képesek a beérkező programkódokat elemezni, fordítani, futtatni különböző programozási nyelveken (például: C, C++, C#, Java, Pascal, Python).

A rendszert modulárisra terveztük, az alábbi rétegeket megkülönböztetve egymástól:

- a programkódok fordításáért és futtatásáért felelős worker-ek,
- a feladatok kiosztását felügyelő Dispatcher alkalmazás (ennek funkciói jelennek meg a ProgCont API szolgáltatásaiként), és
- a felhasználói interfészek (webalkalmazások) a különféle igények kielégítésére.

Ha valakinek az általunk felajánlott kapacitás nem elegendő, akkor ezek a részek önállóan is telepíthetők saját infrastruktúrára. Természetesen ez sokkal több konfigurációs beállítást igényel, mint az általunk biztosított ProgCont API végpont – az `api.progcont.eu` – felhasználása.

## 2.2. A ProgCont API szolgáltatásai

A ProgCont API szolgáltatásai: a programozói feladatokhoz készült forráskódokat képes lefordítani különböző programozói nyelven (C, C++, C#, Java, Pascal, Python), ezen lefordított kódokat futtatja és vizsgálja a keletkezett kimenetet, a program kimeneti kódját, hiba kimenetét, standard outputját és a program futási idejét.

Az API központi eleme egy XML állomány, amely tartalmaz egy feldolgozási feladatot, a feldolgozási feladat pedig tartalmazhat néhány a feltöltött forráskódtól független részt, például hogyan akarunk értesülni a felhasználás eredményéről, vagy ha több felhasználó is használja a rendszert, akkor milyen prioritásba kerüljön a feldolgozás.

Az API két különböző XML sémát támogat:

1. az *egyszerűsített kiértékelést* leíró állomány egy CodeRunner<sup>1</sup> szerű [4] szolgáltatásokat leíró XML, amely tartalmazza a:
  - a. forrásállományt, amelynek tesztelését el kívánjuk végezni,
  - b. teszteseteket:
    - i. a standard bemenettel,
    - ii. a parancssori argumentumokkal,
    - iii. a program futtatásakor elérhető állományokkal,
    - iv. az elvárt kimenettel vagy kimenetekkel,
2. a *szakértői kiértékelést* leíró állomány, amely a fentiek mellett tartalmaz:
  - a. fordítási opciókat,
  - b. interpreter beállításokat,
  - c. kimenetet ellenőrző programkódot,

amelyek közül a nyilvánosság számára az egyszerűsített változat az elérhető. Ez az, ami a worker-ek működésével kapcsolatos háttértudást – ilyenek lehetnek az operációs rendszer és a fordítóprogram verziófüggő sajátosságai – nem igényel.

A ProgCont API egyszerűsített kiértékelést leíró XML sémája nyilvános, a következő címen érhető el: <https://api.progcont.eu/docs/job.xsd>.

A séma részletes bemutatása helyett álljon itt egy példa egy kiértékelési feladat leírására (`job_in.xml`). Ebben egy egyszerű ANSI C program – amely egy darab `main.c` forrásállományban kapott helyet – kiértékelését kértük. Az állomány esetében épp úgy, ahogy később a standard bemenet és az elvárt kimenet leírásánál ASCII kódolású (`target-encoding="US-ASCII"`) szöveges állományokat használunk, ahol a sorok végét az operációs rendszernek megfelelő sorvégejelek zárják (`line-endings="platform-dependent"`). A fordítóprogramtól függően a worker kerülhet akár Linux-os vagy Windows-os gépre is, az ebből adódó különbségeket azonban kiküszöböltük. A kiértékelés eredményében a fordítóprogram kimenetét is fel szeretnénk tüntetni (`report-output="yes"`). Egyetlen teszteset segítségével ellenőrizzük a programot (`test-case-id="t1"`), amely esetében legfeljebb 5 másodperces futási időt

---

<sup>1</sup> A CodeRunner a Moodle beépített plugin-ja, mely segítségével a diákok által különféle programozási feladatok megoldásaként készített programkódok futtathatók és értékelhetők [4].

engedélyeztünk (`time-limit="5000ms"`). A program akkor helyes, ha a megadott elvárt kimenetet produkálta.

Példa egy kiértékelési feladat leírására (`job_in.xml`):

```
<job-request job-id="j1" source-language="C">
  <source-code>
    <file file-name="main.c">
      <text target-encoding="US-ASCII" line-endings="platform-
        dependent">
        #include &lt;stdio.h>;
        int main(){
          int szam, osszeg = 0;
          while (scanf("%d", &szam) > 0) {
            osszeg += szam;
          }
          printf("%d", osszeg);
          return 0;
        }
      </text>
    </file>
  </source-code>
  <compile-with report-output="yes"/>
  <test-cases>
    <test-case test-case-id="t1">
      <execute-with report-output="yes" time-limit="500ms">
        <input>
          <text target-encoding="US-ASCII" line-endings=
            "platform-dependent">5 4 9 7 4</text>
        </input>
      </execute-with>
      <validate-with report-output="yes">
        <answer>
          <text target-encoding="US-ASCII" line-endings=
            "platform-dependent">29</text>
        </answer>
      </validate-with>
    </test-case>
  </test-cases>
</job-request>
```

Példa a kiértékelési feladat eredményére (`job_out.xml`):

```
<job-response job-id="j1">
  <compile exit-code="0"> <stdout/> <stderr/> </compile>
  <test-result test-case-id="t1">
    <execution exit-code="0" output-size="2" execution-time="6ms">
      <stdout> <text target-encoding="US-ASCII">29</text> </stdout>
      <stderr/>
    </execution>
    <validation verdict="PASS"/>
  </test-result>
</job-response>
```

A kiértékelés eredménye (`job_out.xml`) egyfelől tanúskodik arról, hogy a fordítás sikeresen lezajlott, és a program futása a megadott tesztesetre (`t1`) is sikeres volt (`exit-code="0"`). A futás eredménye a standard kimeneten 2 karakter (`output-size="2"`), amely 6 milliszekundum alatt állt elő (`execution-time="6ms"`) úgy, hogy a hibakimenet üres volt. A kimenet megegyezett az elvárt kimenettel, így a program átment a teszten (`verdict="PASS"`).

### 3. Szakköralkalmazás

Az alkalmazás nem más, mint a ProgCont API segítségével fejlesztett informatika szakkörök lebonyolítását segítő webalkalmazás.

A fejlesztés szerepe kettős:

1. szeretnénk beemelni a középiskolákat a ProgCont rendszer felhasználóinak táborába, alternatívát biztosítva Moodle CodeRunner kiegészítője mellé [4], bízva abban, hogy a szűkebb célcsoport igényeit jobban ki tudjuk szolgálni;
2. forráskódját publikálva bemutatjuk, hogy miként lehet a ProgCont API szolgáltatásait felhasználó alkalmazásokat fejleszteni.

#### 3.1. Kliens program

Az alkalmazás első elkészült komponense a ProgCont API kliens, amely a felhasználói interfész szolgáltatásainak eléréséért felelős. Egy olyan Java nyelven íródott mintakód, amely segítségével feltölthetők a kiértékelendő programozási feladatok megoldásai és tesztesetei, továbbá letölthetők a kiértékelés eredményei.

Az API felhasználásához egy felhasználóra és a hozzá tartozó két regisztrációs kulcsra van szüksége, amelyeket a kliens programba kell beépíteni. Innentől kezdve a kommunikáció a ProgCont szolgáltatásaival hiteles és titkos, annak érdekében, hogy érzékeny környezetben – például számonkérések vagy versenyek lebonyolítása során – is alkalmazható legyen.

#### 3.2. Szakkörök támogatása

A felhasználói webinterfésze jelenleg tervezési szakaszban jár.

A tervezés során elsősorban a már meglévő szoftvereink funkciói közül merítettünk ötleteket:

1. `exam.progcont.eu` egyetemi programozás kurzusok számonkéréseinek menedzselésére készült zárt webalkalmazás, mely az ELTE Bíró [8] nevű webfelületéhez hasonló,
2. `deikrpcs.progcont.eu` a Debreceni Egyetem Regionális Programozó Csapatversenyéhez készült webalkalmazás, ami a PC<sup>2</sup> [6], illetve a Mooshak [5] szoftverek ötletét gondolta tovább a helyi igényekhez alkalmazkodva,
3. `progcont.hu` nyilvánosan is elérhető gyakorlófelület, amely az ELTE Mester [8] nevű és az UVa Online Judge [7] számunkra fontos jellemzőit ötvözi.

A Moodle CodeRunner [4] kiegészítőjéhez hasonló alkalmazást tervezünk megvalósítani, ebben a Moodle-ben már jól bevált alábbi funkciókkal, például:

- felhasználói fiókok kezelése,
- tananyagok feltöltésének lehetősége,
- programozói feladatok kérdésbankja,
- jelentések és statisztikák megjelenítése.

## 4. Összefoglalás

Cikkünkben bemutatásra került a ProgCont alkalmazás legújabb fejlesztési iránya, amely reményeink szerint kitágítja a felhasználók körét. Egy egyszerű példa segítségével röviden bemutattuk, hogy milyen szolgáltatásokat nyújt a ProgCont API a lehetséges felhasználók számára. Az itt bemutatott változat már egy hosszabb fejlesztési szakasz eredménye, amellyel egy logikusan felépülő és könnyen használható interfész jött létre. Szintén bemutattuk a kliens program első változatát, amely leegyszerűsíti a felhasználók számára az API szolgáltatásainak használatát.

A fejlesztés itt nem fejeződött be, a közeljövőben olyan webalkalmazásokat szeretnénk készíteni, amelyek példaként állnak az API-t felhasználni kívánók előtt. Első ilyen alkalmazásunk a középiskolákban tanító informatikatanárok számára nyújt segítséget programozó szakkörök lebonyolításához.

## 5. Köszönetnyilvánítás

A ProgCont rendszer további fejlesztését az „EFOP-3.4.4-16-2017-00023 AZ MTMI szakokra való bekerülést elősegítő innovatív programok megvalósítása a Debreceni Egyetem vonzáskörzetében” című projekt támogatta.

## Irodalom

1. R. Tóth, M. Kósa, T. Kádek, J. Pánovics: *The Development of Evaluation Systems at the Faculty of Informatics*, University of Debrecen. In: L., Gómez Chova; A., López Martínez; I., Candel Torres – INTED2019 PROCEEDINGS Valencia, Spanyolország; International Academy of Technology, Education and Development (IATED) (2019) 5552–5559.
2. R. Tóth, M. Kósa, T. Kádek, J. Pánovics: *A ProgCont rendszer új szolgáltatásaira épülő alkalmazások*. SZÁMOKT 2018, Tusnádfürdő, Románia: Erdélyi Magyar Műszaki Tudományos Társaság, (2018) 331–336.
3. T. Kádek, P. Bíró: *A ProgCont API: programozási feladatok megoldásainak újszerű kiértékelése*. SZÁMOKT 2019 konferencia, Erdélyi Magyar Műszaki Tudományos Társaság (EMT) kiadó, Temesvár (2019) 191-195.
4. CodeRunner Documentation V3.1.0 (2019)  
<https://coderunner.org.nz/mod/book/tool/print/index.php?id=184> (utoljára megtekintve: 2019.10.31.)
5. M. Rubio-Sánchez, P. Kinnunen, C. Pareja-Flores, Á. Velázquez-Iturbide: *Student perception and usage of an automated programming assessment tool*, Science of Computer Programming, 31, (2014) 453-460.  
<https://doi.org/10.1016/j.chb.2013.04.001>
6. A. S. Arefin, M. A. Rahman, S. A. Sharna, S. Mahmud, D. M. Kaykobad: *Secured Programming Contest, System with Online and Real-time Judgment, Capability*, 8th International Conference on, Computer and Information Technology (ICCIT), IUT, Dhaka, (2005) 584–586.
7. E. Verdú, L. M. Regueras, M. J. Verdú, J. P. Leal, J. P. de Castro, R. Queiros: *A distributed system for learning programming on-line*. Computers & Education. (58) 1, (2012) 1–10.  
<https://doi.org/10.1016/j.compedu.2011.08.015>
8. Győző Horváth, Gyula Horváth, László Zsakó: *A bíró és a mester – az online értékelés szerepe a programozás oktatásában, Mérési és értékelési módszerek az oktatásban és a pedagógusképzésben*. Szerk. Károly Krisztina és Homonnay Zoltán, Diszciplínák tanítása – a tanítás diszciplínái 5, ELTE Eötvös Kiadó, Budapest (2017) 89–103.