

A ProgCont rendszer jelene és jövője

Pánovics János¹, Kádek Tamás², Biró Piroska³

{panovics.janos¹, kadek.tamas², biro.piroska³}@inf.unideb.hu

Debreceni Egyetem, Informatikai Kar^{1,2,3}

Sapientia Erdélyi Magyar Tudományegyetem, Csíkszeredai Kar³

Absztrakt. A ProgCont rendszert 2011 óta használjuk és fejlesztjük a Debreceni Egyetem Informatikai Karán. A rendszer legfontosabb szolgáltatása a programozási feladatokra beküldött forráskódok automatikus értékelése. Az első pár évben még csak programozói versenyek lebonyolítására használtuk, majd folyamatosan integráltuk az oktatásba, alkalmassá téve számonkérések lebonyolítására és gyakorló feladatsorok közzétételére. Az utóbbi pár évben felértékelődött az online oktatástámogató eszközök szerepe, és ennek megfelelően átalakult a ProgCont rendszer szolgáltatási palettája is. Cikkünkben rövid áttekintést nyújtunk ezen szolgáltatásokról, melyek egyben a későbbi fejlesztési irányt is megszabják.

Kulcsszavak: ProgCont rendszer, automatikus kiértékelés, programozás, szakkör alkalmazás

1. Bevezetés

Az elmúlt három évben az oktatási rendszer számos új kihívással találta szembe magát. A pandémia hirtelen átállást kényszerített ki a jelenléti oktatásról az online eszközök irányába. A Debreceni Egyetem Informatikai Karán a programozási nyelvek oktatásában, elsősorban számonkérésekre és versenyfeladatok kiértékelésére használt ProgCont rendszer alkalmasnak tűnt a távolléti oktatás kihívásaira való gyors reagálásra [1], [2], [3].

A rendszer előnye, hogy a Biró és Mester szoftverekhez hasonlóan [4] a különböző forrásnyelveken beküldött feladatok megoldását automatikusan online értékeli. Így a laborgyakorlatokon kiválthatja a szemináriumvezető munkájának egy részét azzal, hogy ítéletet mond az elkészített feladatmegoldások felett.

Természetesen ez nem tudja helyettesíteni a hibákat tartalmazó megoldások problémáinak felderítésében játszott oktatói szerepet. A rendszer jellegéből adódóan a hibás megoldások értékelése igen csak szűkszavú volt, a visszautasításának okát mindössze az alábbi kategóriákra bontottuk:

- **Szintaktikai hiba:** azt jelzi, hogy a feltöltött forrásszöveg szintaktikailag hibás. Tekintettel arra, hogy a benyújtott megoldás ebben az esetben nem hajtható végre, a fordítóprogram hibaüzenetén kívül más információ nem áll rendelkezésre (1. táblázat).
- **Szemantikai hiba:** azt jelzi, hogy a program hibaüzenettel megállt (E-Run), nem fejeződött be időben (E-Tme), vagy nem a megfelelő kimenetet produkálta (E-Res, E-Pre). A beküldött megoldást tipikusan több tesztesettel teszteljük, így a visszajelzés futtatásról futtatásra változhat (2. táblázat).

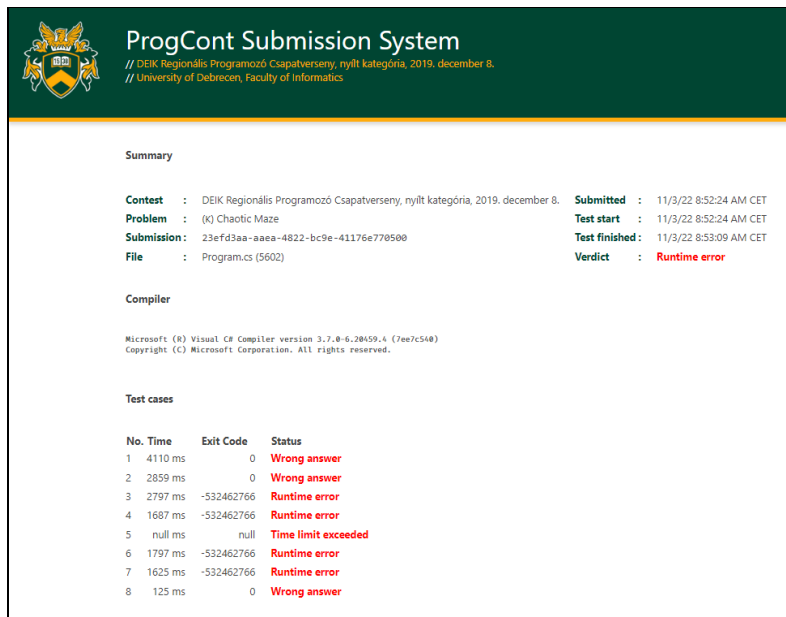
Visszajelzés	Magyarázat
E-Cmp	Fordítási hiba: A beadvány szintaktikailag hibás. Nem tudjuk lefordítani és lefuttatni a benyújtott programot, így nem tudjuk kiértékelni a teszteseteket rajta.

1. táblázat: Szintaktikai hibák

Visszajelzés	Magyarázat
E-Run	Futtatási hiba: A program végrehajtása meghiúsult, pl. hibáüzenettel megszakadt.
E-Tme	Időkorlát-túllépés: A program végrehajtása a megadott időkorláton belül nem fejeződött be, ezért a futása megszakításra került.
E-Res	Rossz válasz: A program futtatásának kimenete eltér az elvárt eredménytől.
E-Pre	Prezentációs (formátum-) hiba: A program futtatásának kimenete eltér az elvárt eredménytől, de az eltérés csak a szóköz karakterekből adódik.

2. táblázat: Szemantikai hibák

Annak érdekében, hogy a visszajelzések minél több információt nyújtsanak a hibakeresés során, kihasználva a rendszer azon tulajdonságát, hogy egy-egy megoldás ellenőrzéséhez több tesztet is lefuttat, kézenfekvőnek tűnt az értékelést tesztetenkénti bontásban is megadni.



ProgCont Submission System
 // DEK Regionális Programozó Csapatverseny, nyílt kategória, 2019. december 8.
 // University of Debrecen, Faculty of Informatics

Summary

Contest : DEK Regionális Programozó Csapatverseny, nyílt kategória, 2019. december 8. **Submitted** : 11/3/22 8:52:24 AM CET
Problem : (K) Chaotic Maze **Test start** : 11/3/22 8:52:24 AM CET
Submission : 23efd3aa-aaaa-4822-bc9e-41176e778500 **Test finished** : 11/3/22 8:53:09 AM CET
File : Program.cs (5602) **Verdict** : **Runtime error**

Compiler

Microsoft (R) Visual C# Compiler version 3.7.0-6.28459.4 (7e07c548)
 Copyright (C) Microsoft Corporation. All rights reserved.

Test cases

No.	Time	Exit Code	Status
1	4110 ms	0	Wrong answer
2	2859 ms	0	Wrong answer
3	2797 ms	-532462766	Runtime error
4	1687 ms	-532462766	Runtime error
5	null ms	null	Time limit exceeded
6	1797 ms	-532462766	Runtime error
7	1625 ms	-532462766	Runtime error
8	125 ms	0	Wrong answer

1. ábra: Megoldás értékelése tesztetenként

Az 1. ábrán egy olyan beküldött megoldás látható, mely mind a nyolc teszten elbukott, három alkalommal hibás választ adott, négyszer futási hibával állt meg, és egy tesztet esetében túllépte a megengedett időkeretet és ezért leállításra került. A ProgCont rendszer ez esetben a futási hiba (E-Run) értékelést adja (mert ez a legmagasabb prioritású hiba). A tesztetenkénti eredményből azonban már az is látszik, hogy még azon esetekben is, amikor sikerült lefutnia a programnak, hibás eredményt produkál.

Természetesen az, hogy milyen jellegű hibák fordultak elő, a teszteset tartalmának ismerete nélkül továbbra sem segíti hatékonyan a hiba felderítését. A tesztesetek nyilvánosságra hozatala ugyanakkor kerüendő, mert ennek eredményeképp olyan programok születhetnének, amelyek csak és kizárólag a tesztesetekre adnak helyes megoldást.

Ezen ellentmondás feloldásaként vezettük be a teszteset-annotációs módszert, amely az egyes tesztesetek tartalmi különbségeit igyekszik kiemelni a konkrét teszteset felfedése nélkül [5], [6], [7].

INBPM9931 Magas szintű programozási nyelvek 3,
2021. május 12., Sz12 ZH
2021. május 12. 12:00 – 2021. május 12. 14:00

Feladatok ■ Kijelentkezés

10: ERES

all_numbers_between_bounds: false

A teszttárolmány szöveges (nem bináris) adatokat tartalmaz.

lower_bound: 1

A teszttárolmány sorainak hossza legfeljebb 10 Kbyte.

number_of_lines: 1K

separator_space: NONE

sorted: RANDOM

A teszttárolmány több tesztesetet is tartalmaz.

test_cases: 44

upper_bound: 44

zero_numbers: false

2. ábra: Megoldás értékelése annotációkkal

A 2. ábrán egy beküldött megoldás értékelésének részlete látható, mely szerint a 10-es számú teszteseten hibás kimenetet állított elő az alkalmazás. A teszteset tartalma ugyan továbbra sem elérhető, az azonban látszik, hogy milyen tesztadatok kerültek a 10-es számú tesztesetbe. Összehasonlítva a többi tesztesettel, így már sok esetben beazonosítható, hogy melyik részfeladat megoldása okozta a kihívást.

2. Teszteset-annotációk

A probléma érzékelésére nézzük meg a következő feladatot, mely a *Magas szintű programozási nyelvek 1* vizsgához tartozó feladatsor tagja volt, majd a vizsga után gyakorló feladatként jelent meg a ProgCont rendszerben.

FELADAT — Időpontok¹

Írjon programot, amely a standard bemenetről állományvégeig (EOF) „HH.MM” alakú, 24 órás intervallumon értelmezett, érvényes időpont adatokat olvas! A program írja a standard kimenetre az időpont 12 órás intervallumon értelmezett angol megfelelőjét úgy, hogy a 0–12 óra közötti időpontokat „am” utótaggal, a 12–24 óra közöttieket pedig „pm” utótaggal látja el, arra is ügyelve, hogy az angol időpontok között nem szerepelhet 0 órás időpont. A 10-nél kisebb órákat egy számjeggyel, a percekét viszont mindig két számjeggyel kell megadni. Például: 0.02-ből 12.02am, 11.58-ból 11.58am, 12.32-ből 12.32pm, 13.29-ből 1.29pm, 22.17-ből 10.17pm lesz.

2.1. Értékelés annotációk nélkül

Erre a feladatra már több mint 1300 beküldés érkezett. Kezdetben két tesztet volt a feladathoz. Az egyik egy rövid minta volt, amely a feladat leírásában is szerepel. A második tesztet az összes lehetséges bemenetet tartalmazta, összesen 1 440 sorból állt, amelyek mindegyike egy-egy feladatot jelentett. Az idők rendezetlenül jelentek meg a tesztfájlban.

Hasonló problémákat már korábban is sokféleképpen elemeztünk. A beküldött megoldások 30%-a oldotta meg helyesen a feladatot és 13%-a a két tesztet közül csak az egyikben működött helyesen. Mivel a második tesztet tartalmazta az összes lehetséges bemenetet, nem nehéz kitalálni, hogy ezek a programok ezen a második teszten buktak meg.

2.1. Értékelés annotációkkal

Az eredeti visszajelzésekből a hiba okára vonatkozó következtetéseket nem lehet levonni, ugyanakkor a beküldött megoldások bizonyos részfeladatokat (bizonyos időintervallumokba eső értékek konverzióját) helyesen oldották meg. Ezért 9 új tesztet bővítettük a feladatot, melyeket az alábbi annotációkkal láttunk el.

Alapvetően kétféle annotáció különböztethető meg: olyan, amelyik nem használja ki a feladat jellegzetességeit, hanem általános problémákra hívja fel a figyelmet (feladatfüggetlen annotációk, 3. táblázat), és olyan, amely kifejezetten a feladatleírásból fakadó különféle sajátosságokra koncentrál (feladatspecifikus annotációk, 4. táblázat).

Annotációk	Leírás
<code>empty input = true</code>	a tesztet egy üres bemeneti fájlból áll
<code>input type = text</code>	a tesztetben szöveges inputot kell feldolgozni
<code>line length = max 10</code>	a bemeneti sorok hossza legfeljebb 10 karakter lehet
<code>lines = none</code>	a tesztet nem tartalmaz feladatokat
<code>lines = more</code>	a tesztet egynél több feladatot tartalmaz

3. táblázat: Feladatfüggetlen annotációk

Általában a szélsőséges input adat (pl. az üres bemenet) kihívások elé állíthatja a feladatmegoldót.

¹ <https://progcont.hu/progcont/100029/?pid=200502>

Annotációk	Leírás
hour = 0	a teszteset csak olyan feladatokat tartalmaz, amelyekben az egyes időpontok óráértéke 0; azaz a helyes kimenet 12.MM am.
hour = 1-11	a teszteset csak olyan feladatokat tartalmaz, amelyekben az egyes időpontok óráértéke 1–11; azaz a helyes kimenet 1–11.MM am.
hour = 12	a teszteset csak olyan feladatokat tartalmaz, amelyekben az egyes időpontok óráértéke 12; azaz a helyes kimenet 12.MM pm.
hour = 13-23	a teszteset csak olyan feladatokat tartalmaz, amelyekben az egyes időpontok óráértéke 13–23, azaz a helyes kimenet 1–11.MM pm.
minute = 10-59	a teszteset csak olyan feladatokat tartalmaz, amelyekben a perc két számjegyű.
minute = 0-9	a teszteset csak olyan feladatokat tartalmaz, amelyekben a perc egyjegyű; ezért a helyes kimenet perc része 0-val kezdődik.

4. táblázat: Feladatspecifikus annotációk

Ezek segítségével megállapíthattuk, hogy a részben jó megoldások esetében többségében a 0 óra és az egy számjegyű perc értékek megjelenítése okozta a legnagyobb kihívást.

Ami ennél is fontosabb, hogy az annotációk segítségével a részben jó megoldást készítő diákok be tudták azonosítani azt a részfeladatot, melyet még nem teljesítettek sikeresen, és ez nagyban megkönnyítette a hibakeresést, ami így önállóan is elvégezhető volt.

3. Szakkör alkalmazás

A ProgCont rendszer fejlesztésében fontos feladat a hozzáférhetőség javítása. Nemcsak házon belüli számonkéréseken és versenyeken szeretnénk használni a szoftvert, hanem nemzetközi versenyek lebonyolításában, nyári programozó szakkörökön, és úgy általában programozási feladatok automatikus kiértékelésében akár házon kívül is [8].

A rendszerben lévő feladatok majdnem teljes egészében a külvilág számára is korlátozások nélkül elérhetők a **progcont.hu** weboldalon, ahol korábbi versenyünk, számonkéréseink, gyakorló feladatsoraink nagy számban állnak az érdeklődők rendelkezésére (47 versenyfeladatsor, 253 zárthelyi dolgozat, összesen 1795 feladat).

A ProgCont rendszer ezen feladatokra fogadja a megoldásokat, melyeket mindenféle regisztráció nélkül, folyamatosan bővülő programozásnyelv-kínálat mellett értékel (jelenleg C, C++, C#, Java, Pascal, Python és Racket nyelvű megoldásokat tud tesztelni).

A szakkör alkalmazás keretein belül megvalósult legfontosabb fejlesztés a ProgCont rendszerben a szoftver modularizálása volt. Ennek keretén belül alapjaiban gondoltuk újra, hogy az automatikus értékelő szolgáltatás miként alakítható át úgy, hogy számos különböző felhasználási esetben egyaránt használható legyen. Végeredményként egy rugalmas ProgCont API került kidolgozásra [9], mely egységes keretbe foglalja a különféle kiértékelési feladatokat. A szakkör alkalmazás valójában nem egyetlen alkalmazás, hanem alkalmazáskomponensek gyűjteménye (diák felület, zsűri weblap, adminisztrátori oldalak), melyeket egymáshoz kapcsolva kínálnak egyre több és egyre inkább testre szabható lehetőséget a ProgCont erősségeinek kihasználására.

4. Összefoglalás

A pandémia okozta kényszerek új lendületet adtak az online oktatástámogató eszközök fejlesztésének. Felértékelődtek az önálló felkészülést támogató szoftverek. Ugyanakkor az is látszik, hogy ahhoz, hogy a ProgCont rendszer ezen a téren is sikeres legyen, további fejlesztésekre van szükség. Az utóbbi két év tapasztalataiból kiindulva, két fő fejlesztési irány rajzolódik ki.

Egyfelől fejleszteni kell a rendszer visszajelzéseit annak érdekében, hogy a legalább részben jó megoldások esetében iránymutatást adjon a lehetséges hibák okainak felderítéséhez.

Másfelől ki kell terjeszteni a rendszer elérhetőségét, kihasználva a moduláris felépítésben rejlő azon lehetőséget, hogy a különböző felhasználói körök számára testre szabott felületeket tud kínálni.

Irodalom

1. P. Biró, T. Kádek: *Automatic evaluation of programming tasks at the University of Debrecen*, in INTED2020 Proceedings, 2–4 March, 2020, pp. 3522–3527, DOI:10.21125/inted.2020.0994.
2. P. Biró, T. Kádek: *How can the ProgCont system help with programming education during a pandemic?* In 15th International Conference on Economics and Business: Challenges in the Carpathian Basin : Global Challenges - Local Answers. Interdependencies or Slobalisation? Cluj-Napoca, Editura Risoprint (2021), pp. 179-191.
3. P. Biró, T. Kádek: *The Mathability of Computer Problem Solving with ProgCont*. Acta Polytechnica Hungarica 2022, 19(1) pp. 77-91. <https://doi.org/10.12700/APH.19.1.2022.19.6>
4. G. Horváth, G. Horváth, L. 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, pp. 89–103.
5. P. Balázs, P. Biró, T. Kádek, M. Kósa, J. Pánovics: *Mathability and exploring mathematical skills of programmers with exercises' annotations*. In: Nikodem, Jan; Klempos, Ryszard (eds.) 12th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2021): Proceedings IEEE 2021, pp. 347-350.
6. P. Biró, M. Kósa, J. Pánovics, T. Kádek: *Test case annotation of programming tasks in the ProgCont system for differential education and strengthening stand-alone preparation opportunities*. In EDULEARN21 Proceedings 2021, (pp. 5433-5441). <http://doi.org/10.21125/edulearn.2021.1109>
7. P. Biró, T. Kádek, M. Kósa, J. Pánovics: *A New Method to Increase Feedback for Programming Tasks During Automatic Evaluation*. Acta Polytechnica Hungarica 2022, 19(9) pp. 103-116. <https://doi.org/10.12700/APH.19.9.2022.9.6>
8. T. Kádek, P. Biró: *A távolléti oktatás hatásai a ProgCont rendszerre*. In XXI. Energetika-Elektrotechnika – ENELKO és XXX. Számítástechnika és Oktatás – SzámOkt Multi-konferencia 2020, (pp. 104–109).
9. T. Kádek, P. Biró: *A ProgCont API: programozási feladatok megoldásainak újszerű kiértékelése*. In ENELKO 2019 SzámOkt 2019 (pp. 191–195).