



MAGDALENA ANDRZEJEWSKA

Przyczyny niepowodzeń edukacyjnych studentów informatyki na wstępnych kursach programowania

Factors Affecting of Computer Science Students' Failure in an Introductory-level Programming Courses

Doktor inżynier, Uniwersytet Pedagogiczny im. Komisji Edukacji Narodowej w Krakowie, Wydział Matematyczno-Fizyczno-Techniczny, Instytut Informatyki, Zakład Badań Edukacyjnych i Nowych Mediów, Polska

Streszczenie

Nauczanie i uczenie się programowania to trudny proces, a wskaźniki niepowodzeń edukacyjnych na kursach z podstaw programowania są wysokie. W artykule zaprezentowano wyniki sondażu przeprowadzonego wśród studentów I roku informatyki, którego celem było poznanie opinii respondentów na temat czynników wpływających na te niepowodzenia. Uzyskane wyniki wskazują, że decydujące znaczenie ma motywacja studentów oraz trudności związane z nabywaniem umiejętności programistycznych.

Słowa kluczowe: informatyka, nauczanie i uczenie się programowania, podstawy programowania

Abstract

Teaching and learning fundamental programming is a difficult process and the failures rates of students on these subjects are high. The article describes the results of a survey conducted among CS1 students in order to get to know their opinion on the factors that affect these failures. In the respondents' opinion, the motivation of students and difficulties related to the acquisition of programming skills are decisive ones.

Keywords: computer science, teaching and learning programming, introduction to programming

Wstęp

Zapotrzebowanie rynku pracy na specjalistów z branży IT jest bardzo wysokie, a umiejętność programowania ma w tym zawodzie znaczenie kluczowe. Na systemach edukacyjnych spoczywa więc duża odpowiedzialność za kształcenie ekspertów z tej dziedziny. Jednak problemy z edukacją programistyczną utrzymują się i pojawiają ponownie w każdym kolejnym pokoleniu studentów kierunków informatycznych i im pokrewnych. Doświadczenia nauczycieli akade-

mickich i badania prowadzone na przestrzeni kilku ostatnich dziesięcioleci pokazują, że nauka programowania na wstępnym jej etapie jest trudnym procesem, a wielu studentów nie odnosi w tym zakresie sukcesów edukacyjnych (Mendes, Paquete, Cardoso, Gomes, 2012). Bennedsen i Caspersen (2007) podają, że średni wskaźnik rezygnacji ze studiów związany z niepowodzeniami na wstępnych kursach programowania wynosi 33%.

Jakie są przyczyny tego zjawiska? Jednym z podstawowych aspektów jest złożoność samego procesu nabywania umiejętności programowania. W ujęciu psychologicznym jest to aktywność poznawcza wymagająca różnego rodzaju modeli mentalnych (Robins, Rountree, Rountree, 2003, s. 149–151). Fundamentalne znaczenie ma model związany z mechanizmem rozwiązywania problemów, konstrukcją i reprezentacją algorytmów. Ale nauka programowania to nie tylko nabywanie lub rozwijanie kompetencji z zakresu rozwiązywania problemów, to również konieczność poznania wielu abstrakcyjnych pojęć związanych z mechaniką programowania, składnią języka – syntaktyką i semantyką tworzonego kodu (Ala-Mutka, 2004, s. 2–6). Lahtinen, Mutka i Jarvinen (2005) twierdzą, że największym problemem początkujących programistów nie jest rozumienie podstawowych pojęć, ale umiejętność ich stosowania. Kolejną istotną i od kilku dziesięcioleci kontrowersyjną kwestią jest wybór pierwszego języka programowania. Znaczenie tego problemu zaowocowało nie tylko rozszerzoną dyskusją, ale też szeregiem badań nad skutecznością różnych paradygmatów i języków programowania stosowanych w kursach wprowadzających. W jednym z najnowszych badań na tym zagadnieniu (Xinogalos, Pitner, Ivanović, Savić, 2018) nie wykazano jednak różnic w efektach kształcenia oraz ocenie poziomu trudności przedmiotu przez grupy studentów, które uczyły się podstaw programowania z wykorzystaniem dwóch różnych języków i proceduralnego paradygmatu programowania.

Istotny jest nie tylko wybór pierwszego języka programowania, ale także odpowiedniego środowiska programistycznego, którego użytkowanie może być źródłem dodatkowych trudności, z jakimi będą się mierzyć początkujący programiści (Pears i in., 2007). Kolejnym czynnikiem determinującym zaangażowanie studentów i ich sukces edukacyjny jest niewątpliwie motywacja, którą kierują się, wybierając typ studiów. Badania Bergin i Reilly (2005) pokazały, że tylko 22% studentów I roku studiów informatycznych uczy się programowania ze względu na zainteresowanie tym obszarem, 40% – ponieważ uważa, że umiejętności te będą miały wpływ na ich karierę zawodową, a 35% przyznaje, że tylko dlatego, że jest to przedmiot obowiązkowy. Ponadto w badaniach tych potwierdzono, że wewnętrzna motywacja studentów jest silnie dodatnio skorelowana z efektywnością uczenia się programowania.

Metodologia badań

Cel badań

Celem przeprowadzonego sondażu było poznanie opinii studentów o przyczynach niepowodzeń, które ich dotyczą na kursach z podstaw programowania, oraz zbadanie, czy opinia ta zależy od doświadczenia programistycznego, które posiadają.

Charakterystyka badanej grupy

Grupa respondentów liczyła 80 studentów I roku inżynierskich studiów kierunku informatyka. Wśród nich było 66 mężczyzn (82,5%) oraz 14 kobiet (17,5%), średni wiek badanych to blisko 21 lat ($M = 20,8$; $SD = 1,33$). Studenci odbyli kurs z podstaw programowania w języku C (zakończony egzaminem) w semestrze poprzedzającym termin przeprowadzenia ankiety. Połowa respondentów rozpoczęła naukę programowania na studiach, a ponad 72% z nich nie programowało wcześniej w języku C. Szczegółowe dane zestawiono w tabeli 1.

Tabela 1. Doświadczenie respondentów w programowaniu

| Kiedy rozpoczęłaś/eś naukę programowania? | N | % | Jak długo programujesz w języku C? | N | % |
|---|----|-------|------------------------------------|----|-------|
| w szkole podstawowej | 2 | 2,50 | dłużej niż 5 lat | 1 | 1,25 |
| w gimnazjum | 5 | 6,25 | 3–5 lat | 9 | 11,25 |
| w szkole ponadgimnazjalnej | 33 | 41,25 | 1–2 lata | 12 | 15,00 |
| na studiach | 40 | 50,00 | krócej niż rok | 58 | 72,50 |

Źródło: opracowanie własne.

Charakterystykę respondentów uzupełnia informacja o czasie poświęcanym przez nich (poza planem zajęć) na naukę programowania w języku C. 31% studentów przyznało, że jest to mniej niż 1 godzina, a 38%, że 1–2 godziny w tygodniu. Około 31% badanych przeznacza na naukę programowania ponad 3 godziny w tygodniu. Ponadto w ocenie 55% studentów kurs z podstaw programowania był trudniejszy, a 15% uznało go za łatwiejszy w stosunku do kursów z innych przedmiotów (mediana $MD = 4$ w 5-stopniowej skali oceny).

Wyniki badań

Pytanie, na które badani udzieli odpowiedzi, brzmiało: „Co według Ciebie wpływa na niepowodzenia podczas nauki programowania? Oceń stopień wpływu biorąc pod uwagę własne doświadczenia”. Lista czynników była następująca:

K1: Nauczany język programowania i związane z tym tematy będące treścią kursu

K2: Problemy z nabywaniem umiejętności programistycznych

K3: Metody nauczania i sposób prowadzenia zajęć

K4: Indywidualne nastawienie/motywacja studenta

K5: Środowisko programistyczne

Respondenci wyrażali swoją opinię w oparciu o 5-stopniową skalę Likerta z odpowiednimi rangami (wpływ: 1 – bardzo mały, 2 – mały, 3 – średni, 4 – duży, 5 – bardzo duży). Z tabeli 2 możemy odczytać, że respondenci najczęściej (moda $MO = 3$) wskazywali na średni wpływ takich czynników, jak: nauczany język programowania (47,5%), środowisko programistyczne (37,5%) oraz trudności z nabywaniem umiejętności programistycznych (35,0%). Z tym że w przypadku ostatniego z tych czynników dominują dwa wskazania – jednakowy odsetek badanych (35,0%) zdecydował, że jest to duży wpływ ($MO = 4$). Taką samą wartość mody uzyskały metody nauczania – ten stopień wpływu wybrało 31,3% badanych. Wyróżniającą się kategorią ($MO = 5$) było indywidualne nastawienie/motywacja studenta – 40,0% respondentów wskazuje na bardzo duży wpływ tego czynnika na niepowodzenia w procesie nauki programowania.

Tabela 2. Czynniki wpływające na niepowodzenia w procesie nauki programowania – rozkład procentowy

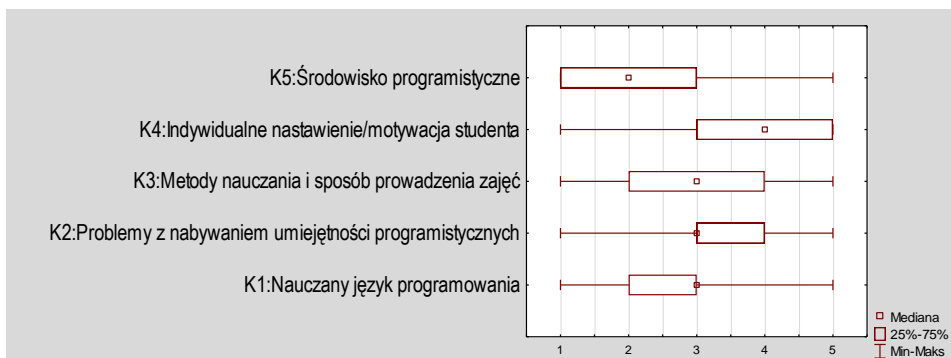
| Czynnik | 1 | 2 | 3 | 4 | 5 |
|--|------|------|------|------|------|
| K1: Nauczany język programowania | 18,8 | 23,8 | 47,5 | 8,8 | 1,3 |
| K2: Problemy z nabywaniem umiejętności programistycznych | 2,5 | 15,0 | 35,0 | 35,0 | 12,5 |
| K3: Metody nauczania | 11,3 | 17,5 | 27,5 | 31,3 | 12,5 |
| K4: Indywidualne nastawienie/motywacja studenta | 1,3 | 3,8 | 21,3 | 33,8 | 40,0 |
| K5: Środowisko programistyczne | 30,0 | 22,5 | 37,5 | 5,0 | 5,0 |

wpływ: 1 – bardzo mały, 2 – mały, 3 – średni, 4 – duży, 5 – bardzo duży

Źródło: opracowanie własne.

Uzyskane wyniki zilustrowano również na rysunku 1, na którym zestawiono informację o wartościach mediany (MD) oraz zakresie pierwszego (Q25) i trzeciego kwartyła (Q75) dla wszystkich rozważanych kategorii. W oparciu o wykres można stwierdzić, że najmniejsze zróżnicowanie odpowiedzi występuje w przypadku opinii o nauczonym języku programowania (K1: MD = 3, Q25 = 2, Q75 = 3) – obserwujemy przesunięcie odpowiedzi w kierunku małego wpływu, oraz problemach z nabywaniem umiejętności programistycznych (K2: MD = 3, Q25 = 3, Q75 = 4) – tu dominują opinie o średnim i dużym wpływie. Uzyskane wartości miar rozkładu pozwalają na stwierdzenie, że w obu przypadkach są to opinie zdecydowanej większości badanych.

W ocenie pozostałych trzech kategorii respondenci wykazali się większą zmiennością. Czynnikiem, który według badanych ma najmniejszy wpływ na rozważany problem, jest środowisko programistyczne (K5: MD = 2, Q25 = 1, Q75 = 3). Czynnikiem neutralnym okazały się metody nauczania i sposób prowadzenia zajęć (K3: MD = 3, Q25 = 2, Q75 = 4), a za najistotniejszą uznano motywację studentów (K4: MD = 4, Q25 = 3, Q75 = 5).



Rysunek 1. Ocena wpływu poszczególnych czynników na niepowodzenia w procesie nauki programowania

Źródło: opracowanie własne.

W związku z tym, że grupa studentów nie była jednorodna pod względem posiadanych umiejętności programistycznych, uznano, że fakt ten może różnicować opinię badanych w analizowanych kwestiach. Przeprowadzony nieparametryczny test U Manna-Whitneya dla każdej z kategorii nie wykazał jednak istotnych statystycznie różnic pomiędzy grupą studentów, którzy rozpoczęli naukę programowania na studiach (grupa G_NDO), a tymi, którzy programowali już na wcześniejszych etapach edukacji (grupa G_DO). Szczegółowe wyniki testu zestawiono w tabeli 3.

Tabela 3. Wyniki testu U Manna-Whitneya

| | Mediana | | Test UMW | | |
|--|-----------------|----------------|----------|-------|-------|
| | G_NDO N = 40 | G_DO N = 40 | U | Z | P |
| K1: Nauczany język programowania | 3 | 3 | 715,5 | 0,81 | 0,419 |
| K2: Problemy z nabywaniem umiejętności programistycznych | 3 | 3 | 799,0 | 0,00 | 0,996 |
| K3: Metody nauczania | 3 | 3 | 761,5 | -0,37 | 0,715 |
| K4: Indywidualne nastawienie/motywacja studenta | 4 | 4 | 689,5 | -1,06 | 0,290 |
| K5: Środowisko programistyczne | 2 | 3 | 624,0 | 1,69 | 0,091 |

Źródło: opracowanie własne.

Respondenci mogli też w pytaniu otwartym opisać, jakie ich zdaniem były główne przyczyny niepowodzeń edukacyjnych – braku zaliczenia lub niezdanego egzaminu. Analiza jakościowa uzyskanych odpowiedzi pozwoliła na ich skategoryzowanie i ujęcie ilościowe. Największy odsetek odpowiedzi (50%) ponownie dotyczył problemu indywidualnego podejścia studentów do przedmiotu i ich zaangażowania w proces nabywania wiedzy i umiejętności. Przykładowe wypowiedzi badanych będące ilustracją tego stanowiska to: *zbyt mało czasu*

poświęconego na naukę; lekceważąca postawa studentów i brak regularnej, systematycznej nauki; brak motywacji do nauki przedmiotu. 18% badanych wskazywało na problemy związane z procesem oceny, czasem i formą zaliczenia lub egzaminu, 12% zgłaszało uwagi dotyczące metod nauczania, a 20% stanowiły wypowiedzi różne.

Podsumowanie

Jak wynika z analizy uzyskanych wyników, czynnikiem, który według respondentów ma największy wpływ na wysoki wskaźnik niepowodzeń na początkowym etapie nauczania i uczenia się programowania, jest indywidualna postawa studentów. Drugą istotną przyczyną są trudności związane z nabywaniem umiejętności programistycznych. Opinie te nie zależą od posiadanego przez ankietowanych doświadczenia w programowaniu. Kluczem do osiągnięcia sukcesu przez osoby rozpoczynające naukę programowania jest zatem ich silna motywacja, szczególnie wewnętrzna, związana z prawdziwym zainteresowaniem przedmiotem, oraz koncentracja procesów i metod dydaktycznych na pomocy studentom w zrozumieniu głównych koncepcji programowania. Przykłady takich skutecznych działań, które pomogły zredukować odsetek niepowodzeń studentów o ponad 77%, zawiera raport Yadina (2011). Po pierwsze, autor proponuje, aby wstępne kursy prowadzić w oparciu o język Python w celu zmniejszenia złożoności składni języka, dzięki czemu studenci będą mogli skupić się na nabywaniu umiejętności rozwiązywania problemów i konstruowania algorytmów. Yadin zaleca również korzystanie z narzędzi do wizualizacji abstrakcyjnych i złożonych koncepcji programistycznych lub algorytmów. A ponadto za efektywne uznaje wprowadzenie systemu zadań indywidualnych – dostosowywanych do poziomu wiedzy i umiejętności każdego ucznia.

Literatura

- Ala-Mutka, K. (2004). *Problem in Learning and Teaching Programming- a Literature Study for Developing Visualizations in the Codewitz-Minerva Project*. Pobrane z: http://www.cs.tut.fi/~edge/literature_study.pdf (15.05.2018).
- Bennedsen, J., Caspersen, M.E. (2007). Failure Rates in Introductory Programming. *ACM SIGCSE Bulletin*, 39(2), 32–36.
- Bergin, S., Reilly, R. (2005). The Influence of Motivation and Comfort-level on Learning to Program. W: P. Romero, J. Good, E. Acosta Chaparro, S. Bryant (Eds). *Proceedings of the 17th Annual Workshop on the Psychology of Programming Interest Group*, Sussex University, (s. 293–304).
- Lahtinen, E., Mutka, K.A., Jarvinen, H.M. (2005). A Study of the Difficulties of Novice Programmers. W: *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005)*, New York: ACM Press. (s. 14–18).
- Mendes, A.J., Paquete, L., Cardoso, A., Gomes, A. (2012). Increasing Student Commitment in Introductory Programming Learning. *Frontiers in Education Conference (FIE) 2012 IEEE*, 1–6.
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., Paterson, J. (2007). A Survey of Literature on the Teaching of Introductory Programming. *SIGCSE Bulletin*, 39(4), 204–223.

- Robins, A., Rountree, J., Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13, 137–172.
- Xinogalos, S., Pitner, T., Ivanović, M., Savić, M. (2018). Students' Perspective on the First Programming Language: C-like or Pascal-like Languages? *Education and Information Technologies*, 23(1), 287–302. DOI: 10.1007/s10639-017-9601-6.
- Yadin, A. (2011). Reducing the Dropout Rate in an Introductory Programming Course. *ACM Inroads*, 2(4), 71–76.