

НЯКОИ РЕЗУЛТАТИ ОТ ТЕСТОВЕ ЗА РАЗБИРАНЕ НА КОД В ОБУЧЕНИЕТО ПО ПРОГРАМИРАНЕ

Владимир Сълов¹

¹ Икономически университет – Варна, катедра „Информатика“,
Варна, България, vsulov@ue-varna.bg

РЕЗЮМЕ

Дисциплините, които запознават студентите с основите по програмиране, присъстват в учебните планове на почти всички специалности в областта на информатиката и компютърните науки. Докладът представя и коментира някои резултати от проведени от автора тестове пред различни видове студенти и на различни езици за програмиране в сходни въвеждащи в програмирането дисциплини. Резултатите могат да послужат за по-добро разбиране на типичните затруднения, които изпитват студентите в подобен тип курсове.

КЛЮЧОВИ ДУМИ: обучение по програмиране, тестове, успеваемост на студентите

SOME RESULTS FROM CODE COMPREHENSION TESTS IN COMPUTER PROGRAMMING CLASSES

Vladimir Sulov¹

¹ University of Economics – Varna, Department of Informatics, Varna, Bulgaria, vsulov@ue-varna.bg

ABSTRACT

The classes that introduce university students to computer programming are present in the curricula of almost every graduate program in the sphere of informatics and computer science. This paper presents and discusses some results from author's tests of different students studying different programming languages in similar introduction to programming classes. The results could facilitate better understanding of the typical difficulties that students have in such classes.

KEYWORDS: computer programming education, tests, students' success rate

ВЪВЕДЕНИЕ

Дисциплините, които запознават студентите с основите по програмиране, присъстват в учебните планове на почти всички специалности в областта на информатиката и компютърните науки (Aleksić and Ivanović, 2016, Kumar, 2024). Типични наименования на подобни дисциплини на български език са „Въведение в програмирането“, „Увод в програмирането“, „Основи на програмирането“, а на английски език – „Introduction to Programming“, „Programming Fundamentals“, „Foundations of Programming Languages“ и др.

Конкретните езици за програмиране, които се използват в различните университети и дисциплини, са различни, най-често C, C++, C#, Java, Python (Aleksić and Ivanović, 2016, Duffany, 2014, Ivanović, 2015). Въпреки разликите в синтаксиса и донякъде в някои от концепциите и парадигмите на тези езици за програмиране, те имат общи идеи, конструкции и черти, разчитат на логическо и алгоритмично мислене, а дисциплините имат сходни изисквания към нивото и възможностите на обучаемите.

Цел на настоящия доклад е да представи някои резултати от проведени тестове пред студенти от Икономически университет – Варна (ИУ – Варна) във въвеждащи в програмирането дисциплини. Резултатите могат да послужат за по-добро разбиране на типичните затруднения, които изпитват студентите и съответно за подобряване на обучението им с акцентирание в правилната посока.

1. ИЗСЛЕДВАНА СЪВКУПНОСТ И МЕТОДИКА

За нуждите на изследването са използвани отговорите на тестовете за текущ контрол и сесийно оценяване на 167 студенти в ИУ – Варна от 1¹, 3² и 5³ курс в специалности „Информатика и компютърни науки“, „Мобилни и уеб технологии“, „Бизнес информационни системи“, „Дигитални технологии в бизнеса“.

Дисциплините, от които са тестовете, са сходни и въвеждащи в програмирането, макар и с различни наименования и различни езици: „Въведение в програмирането“ пред 1 курс (на C/C++), „Алгоритмизация и програмиране“ пред 3 курс (на C#), „Въведение в програмирането“ пред 5 курс (на C#). Изучават се еднакви базови възможности и конструкции като типове данни, управляващи оператори, потребителски функции и др.

Използвани са 2730 отговора на 60 въпроса тип „затворен“. Данните са извлечени от системата за електронно и дистанционно обучение на университета – eLearn⁴, като са експортирани към Excel, където са обработени.

Въпросите представляват кратки фрагменти програмен код, които студентите следва да разчетат, разберат правилно и да дадат отговор какво се извършва, какъв е резултатът, изведен на екрана, или каква е стойността на определена променлива или променливи. Всеки въпрос акцентира на особеностите на определена конструкция и/или оператор/ключова дума от езика за програмиране. Всеки въпрос има само един верен отговор от 4 до 6 възможни. Подобни тестове типично се използват в университетите в сходни дисциплини (Киров, 2019, Simon, 2012, Kuechler and Simkin, 2003).

Методиката на изследването включва най-общо следните задачи/последователност:

1. Да се пресметне средната успеваемост на цялата съвкупност.
2. Да се провери дали резултатите по курсове/специалности са съпоставими и могат да се разглеждат като една съвкупност.
3. Да се групират въпросите по теми (и евентуално подтеми, ако е възможно).
4. Да се пресметнат резултатите по теми и подтеми и да се сравнят със средната успеваемост.
5. На база на горното да се направят някои изводи.

¹ Студенти в ПН 4.6 „Информатика и компютърни науки“, ОКС „Бакалавър“ започват изучаването на програмиране от 1 курс.

² Студенти в ПН 3.8 „Икономика“, ОКС „Бакалавър“ започват изучаването на програмиране от 3 курс.

³ Студенти в ПН 4.6 „Информатика и компютърни науки“, ОКС „Магистър“, които имат приравнителна дисциплина, въвеждаща в програмирането.

⁴ <https://e-learn.ue-varna.bg>

2. СРЕДНА УСПЕВАЕМОСТ И ГРУПИРАНЕ ПО ТЕМИ/ПОДТЕМИ

Изчислената средна успеваемост на всички въпроси на всички видове тестове и студенти е 58%⁵. В сходни изследвания за други университети (Tie at al, 2012, Watson and Li, 2014, Ivanović, 2015, Kuechler and Simkin, 2003), вкл. в различни държави и използване на различни езици за програмиране, се установява, че резултатите и успеваемостта варират в много широки граници (приблизително от 45% до 75%), като изводите са, че не могат да бъдат направени директни сравнения по методика, език за програмиране и други фактори, тъй като в голяма степен резултатите зависят от способностите на съответните изследвани студентски съвкупности.

Тъй като в ИУ – Варна съществува стандартизирана точкова система за оценяване на студентите от 0 до 100 точки (Икономически университет – Варна, 2021), а резултатите от тестовете се включват 1:1 в тези точки при формирането на оценката на разглежданата съвкупност студенти, можем да приравним процентните стойности директно към точки на въпросната точкова система за оценяване.

В университета съществува и скала за преобразуване на крайния брой точки в оценка за семестриалните изпити. 58 точки отговарят на оценка „Добър 4“, но оценките от семестриалните изпити могат да бъдат само цели числа от 2 до 6. В по-подробно разписаната скала (Икономически университет – Варна, 2021), макар и само по отношение на държавните изпити, където това е възможно, 58 точки отговарят на оценка „Добър 3,50“, с 1 точка по-малко от следващата оценка „Добър 3,75“.

Разглеждайки резултатите по различни дисциплини (които са и в различни курсове и специалности), се оказва, че резултатите се движат в интервала 57-59%, което ни дава основание да ги третираме като една съвкупност. Това беше проверено и по отделни групи въпроси.

Съобразно изучавания материал в дисциплините, въпросите бяха групирани в следните основни теми и подтеми:

- управляващи оператори:
 - условни (if, switch, ?);
 - цикли (for, while, break, continue);
- организация на данните:
 - масиви, структури, стрингове;
 - указатели, референции (само за някои езици за програмиране);
- потребителски функции.

3. РЕЗУЛТАТИ ПО ТЕМИ/ПОДТЕМИ, НАБЛЮДЕНИЯ И ИЗВОДИ

Резултатите в проценти верни отговори по отделните теми и подтеми са представени в таблица 1.

⁵ Навсякъде данните са закръглени до цял процент, поради липса на необходимост от още по-висока прецизност.

Резултати на студентите от тестовете по теми/подтеми

Тема / подтема	Резултат (верни отговори)	Оценка по скалата на ИУ
Управляващи оператори, в т.ч.:	57%	3,50
условни (if, switch, ?)	56%	3,50
цикли (for, while, break, continue)	58%	3,50
Организация на данните, в т.ч.:	62%	3,75
масиви, структури, стрингове	66%	4,00
указатели, референции	43%	3,00
Потребителски функции	52%	3,25

По отношение на темата „управляващи оператори“ може да се нарече в известен смисъл неочаквано еднаквото представяне в участващите подтеми, тъй като традиционно студентите се затрудняват повече при формулирането на цикли, когато трябва сами да решат задача, т.е. да напишат собствен програмен код. Като цяло резултатът на темата почти съвпада с общата средна успеваемост.

Студентите показват по-добро представяне в темата „организация на данните“ и особено класически възможности като масиви, структури, стрингове. В същото време, обаче, те значително се затрудняват при разбирането на по-сложните концепции, касаещи разполагането и организацията на данните в паметта, съответно указатели и референции, като резултатът в тази подтема е най-слабият изобщо.

На по-слабо от средното равнище са също и резултатите от темата „потребителски функции“. Въпреки че подобна по-детайлна разбивка не е представена в настоящия доклад, беше установено, че ниските резултати донякъде са свързани по-тясно с разбирането на предаването на параметри, вкл. и при използване на указатели и/или референции, което беше коментирано по-горе.

С оглед на представените резултати могат да се направят няколко основни извода.

Разбирането на програмен код от страна на студентите, които се обучават в въвеждащите в програмирането дисциплини в ИУ – Варна, е на формално добро ниво. Студентите изпитват затруднения в някои конкретни области, на които би следвало да се наблегне повече при преподаването. Реално резултатите могат да се разглеждат за не толкова добри, но отчасти това се дължи и на фактори, които допълнително следва да бъдат изследвани, както посочват и други автори (Tie at al, 2012, Ivanović, 2015), като липса на мотивация, постоянство, отделено време, възможности точно в това отношение у някои от тях и др.

ЗАКЛЮЧЕНИЕ

Направеното изследване показва нивото на представяне на студентите, някои слабости в тяхната подготовка и възможности и очертава насоки за подобрението на успеваемостта им. За още по-добро разбиране на представянето на обучаемите, резултатите от тестовете могат

да се сравнят и анализират спрямо резултатите от практическите задания и контролни, а също да се потърсят други фактори, които влияят на успеваемостта на студентите.

REFERENCES / ИЗПОЛЗВАНА ЛИТЕРАТУРА

1. ИКОНОМИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА. (2021) Правилник за оценяване на знанията, уменията и компетентностите на студентите в Икономически университет – Варна. [Online] Available from: https://ue-varna.bg/uploads/filemanager/303/regulations/Pravilnik_ocenqvanе_znaniq_umeniq_kompetentnost.pdf [Accessed 10/11/2024].
2. КИРОВ, Н. (2019) *Тестове за оценка на знанията по програмиране*. [Online] Available from: <https://eprints.nbu.bg/id/eprint/4091/1/tests.pdf> [Accessed 10/11/2024].
3. ALEKSIĆ, V. and IVANOVIĆ, M. (2016) Introductory programming subject in European higher education. *Informatics in Education*, 15(2), pp.163-182.
4. DUFFANY, J. (2014). Choice of language for an introduction to programming course. *Excellence in Engineering To Enhance a Country's Productivity, Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014)*, pp. 1-9.
5. IVANOVIĆ, M. et al. (2015) Does the choice of the first programming language influence students' grades? *Proceedings of the 16th International Conference on Computer Systems and Technologies*, pp. 305-312.
6. KUECHLER, W. and SIMKIN, M. (2003) How well do multiple choice tests evaluate student understanding in computer programming classes? *Journal of Information Systems Education*, 14(4), pp. 389-399.
7. KUMAR, A. et al. (2024) *Computer Science Curricula 2023*. ACM.
8. SIMON, J. et al. (2012) Introductory programming: examining the exams. *Proceedings of the Fourteenth Australasian Computing Education Conference*, Volume 123, pp. 61-70.
9. TIE, Z. et al. (2012) Analysis on the relationship between student grades and computer programming time in learning the C programming language. *2012 7th International Conference on Computer Science & Education (ICCSE) IEEE*. pp. 1584-1589.
10. WATSON, C., and LI, F. (2014) Failure rates in introductory programming revisited. *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pp. 39-44.