

Мислінчук Володимир Олександрович – доцент кафедри методики фізики і хімії, кандидат педагогічних наук Рівненського державного гуманітарного університету.

Коло наукових інтересів: фізика, астрономія.

Левчун Ірина Миколаївна – студентка II курсу кафедри методики фізики і хімії, Рівненський державний гуманітарний університет.

ДОСВІД ВИКОРИСТАННЯ PYTHON ЯК ПЕРШОЇ МОВИ ПРОГРАМУВАННЯ ДЛЯ СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ «ІНФОРМАТИКА»

Павло МЕРЗЛИКІН

На основі досвіду провідних світових ВНЗ обґрунтовано добір мови програмування для вступних курсів програмування. Описано досвід впровадження Python як першої мови програмування на першому курсі спеціальності «Інформатика» на фізико-математичному факультеті Криворізького педагогічного інституту ДВНЗ «Криворізький національний університет».

On the basis of leading world universities experience the reasonable choice of programming language for introductory programming courses is grounded. The experience of Python implementation as a first programming language for first-year students of "Computer Science" specialty of Physics and Mathematics department of Kryvyi Rih Pedagogical Institute SIHE "Kryvyi Rih National University".

Постановка проблеми. В проєкті Концепції розвитку освіти України на період 2015-2025 років передбачено сприяння міжнародній акредитації освітніх програм українських університетів [1]. У підготовці майбутніх інженерів-програмістів велике значення мають вступні курси, що закладають базові поняття й закладають фундамент стилю мислення фахівця. Тому актуальною є проблема розробки вступних курсів з урахуванням міжнародної практики.

Аналіз актуальних досліджень. Одним з найбільш визнаних міжнародних стандартів підготовки спеціалістів у сфері інформаційних технологій є Computer Science Curricula (CS2013) [2], що розробляється Асоціацією обчислювальної техніки (Association for Computing Machinery) та Комп'ютерним товариством IEEE (IEEE Computer Society). У розділі 5 цього документу приділяється увага проблемі вибору мови програмування для організації вступних курсів (introductory courses). Автори CS2013 проаналізували вступні курси для спеціалістів у галузі інформаційних технологій, що викладалися в провідних світових вишах починаючи з 2001 року та дійшли висновку, що всі їх можна умовно поділити на шість видів: (імперативні (imperative-first), об'єктно-орієнтовані (objects-first), функціонально-орієнтовані (functional-first), оглядові (breadth-first), алгоритмічні (algorithms-first), апаратно-орієнтовані (hardware-first)). Протягом останнього десятиріччя спостерігалася тенденція до розширення спектру мов програмування, що застосовуються в таких вступних курсах. Спостерігався також поступовий перехід до більш «безпечних» мов програмування, або до так званого керованого коду (managed code). Наприклад, перехід від C до Java. Також зростає частка динамічних мов програмування (Python, JavaScript) і візуальних мов програмування (Alice, Scratch). Використання «безпечних» мов програмування дає хороші результати на початкових етапах вивчення програмування, проте забезпечує надто високий рівень абстракції для розуміння особливостей виконання

програми на машинному рівні. Тож автори CS2013 рекомендують використовувати більш низькорівневі мови для поглиблених курсів програмування.

Розглянемо детальніше деякі вступні курси з програмування.

В Університеті Каліфорнії (Берклі) вступний курс програмування носить назву “Структура та інтерпретація комп’ютерних програм” (Structure and Interpretation of Computer Programs) та має шифр 61A. У 2011 році цей курс розділювався на два: 61A з новим навчальним планом і мовою Python та класичний курс 61AS з використанням мови програмування Scheme.

Аналізуючи причини переходу на Python, Брайан Харві [3] відзначає, що великою мірою це пов’язано з даниною моді й «модернізацією» курсу. Аргументами на користь використання Python були його популярність, велика кількість прикладних бібліотек та чимала відкрита спільнота, до якої можуть долучитися студенти.

З іншого боку, він наводить ряд аргументів на користь Scheme. Перш за все, на думку Б. Харві, побудова курсу не повинна починатися з вибору мови програмування, адже це не курс із вивчення конкретної мови програмування. З цього випливає, що краща мова програмування для вступного курсу – не обов’язково краща для написання реальних програм. Кращою для навчання буде та мова, яка не приховує за синтаксичними конструкціями та бібліотеками основні ідеї програмування.

Схожі зміни відбулися в Массачусетському технологічному інституті. Замість класичного вступного курсу 6.001 “Структура та інтерпретація комп’ютерних програм” (Structure and Interpretation of Computer Programs) [4] впроваджено курс 6.01 на базі мови програмування Python. Але не слід розглядати це лише як зміну мови програмування. Курс 6.001 є більш фундаментальним, тоді як 6.01 – більш практично орієнтованим, що дозволяє зацікавити більш широку аудиторію студентів[5].

Метою статті є добір першої мови програмування для навчання студентів за напрямом підготовки «Інформатика» та висвітлення досвіду використання Python у вступному курсі “Програмування” для навчання майбутніх інженерів-програмістів на базі ДВНЗ «КНУ».

Викладення основного матеріалу. Анкетування [6], проведене серед студентів-першокурсників безпосередньо перед вивченням курсу «Програмування» показало, що близько двох третин опитаних вивчали у школі мову програмування Pascal, як основну. Деякі студенти вивчали кілька мов програмування (переважно самотійно, а не в рамках шкільного курсу). Разом з тим, з іншим завданням анкети – написанням будь-якою відомою мовою фрагменту коду для обміну значеннями двох змінних – впоралися лише 3 з 17 студентів. Тобто понад 80% першокурсників не володіють жодною мовою програмування навіть на початковому рівні. За таких обставин мова програмування, яка буде використовуватися в рамках даного курсу, по суті, стане першою для більшої частини групи. При використанні ж мови, що не вивчалась жодним з опитаних, студенти з більш високим рівнем підготовки не будуть втрачати мотивацію до вивчення курсу завдяки елементу новизни.

Тим паче, що першочерговим завданням курсу «Програмування» для майбутніх спеціалістів за напрямом підготовки «Інформатика» є закладення основ професійної

діяльності (розвиток алгоритмічного мислення, формування стилю програмування, здатності знаходити ефективні розв'язки поставлених задач тощо).

Мова програмування Python надає можливість “швидкого старту”. Написання перших програм потребує мінімальних знань синтаксису мови програмування. Ці знання поглиблюватимуться по мірі ускладнення навчальних завдань. Проста задача, як правило, матиме простий розв'язок на Python. Найперші завдання для більшої наочності доцільно виконувати в режимі командного інтерпретатора.

Ще однією безсумнівною перевагою Python є те, що це сучасний інструмент, який має широку сферу застосування і великий набір прикладних бібліотек. Python доступний на багатьох платформах, існують також online-інтерпретатори, що не потребують попереднього встановлення. Тобто використання Python для навчання студентів напряму підготовки “Інформатика” розв'язує не лише навчальні, а й професійні задачі, що в майбутньому постануть перед випускником.

Нарешті, форматування коду як вимога синтаксису Python одразу привчає студентів структурувати код, що є важливою задачею саме на перших етапах навчання програмування.

Крім того знання основ мови програмування Python може знадобитись при майбутньому вивченні курсу «Методи обчислень» з використанням системи комп'ютерної алгебри Sage, при вивченні курсу веб-програмування, моделювання та інших.

Використання Python як першої мови програмування має й свої недоліки, серед яких суттєвим є відсутність статичної типізації. Тому слід приділити особливу увагу введенню поняття типу даних.

Для підтримки навчального курсу було розроблено методичні матеріали, що доступні на сайті kdrp.edu.ua [7]. Логіка побудови лабораторних робіт є цілком типовою для курсу програмування, хоча деякі завдання побудовані таким чином, що спонукають використовувати специфічні можливості Python (як то множення рядку на число). Частина завдань було розроблено на основі популярних збірників задач [8; 9], інші ж є авторськими розробками. Кожна лабораторна робота складається з загальної частини (завдання, що мають бути виконані усіма студентами) та варіативної частини (завдання за варіантами). Кожна лабораторна робота містить 20 варіантів завдань.

Крім лабораторних робіт курс передбачає виконання двох контрольних робіт. Перша – аудиторна – проводиться в середині семестру і містить кілька варіантів завдань для перевірки рівня засвоєння матеріалу. Друга контрольна робота являє собою невеликий навчальний проект, який студенти виконують протягом семестру. Проекти передбачають більш детальне ознайомлення з котроюсь із бібліотек Python на вибір студента (графічні розширення, математичні розширення тощо).

Складовими оцінки за лабораторну роботу є: виконання роботи, захист роботи, звіт про виконання, якість виконання, дотримання графіку. Оскільки даний курс фактично є вступним до спеціальності, то велика увага приділяється стилю програмування. Складова “звіт про виконання” передбачає штрафи (у відсотках від балів, що нараховуються за лабораторну роботу) за погане коментування коду або відсутність коротких пояснень у звіті про виконання лабораторної роботи. Складова “виконання роботи” допускає зниження оцінки за неінформативне виведення результатів роботи програми, відсутність

фільтрування вхідних даних. Складова “якість виконання” передбачає бонуси за реалізацію зручного інтерфейсу користувача, використання програмних конструкцій, які оптимізують код за критеріями швидкодії або обсягу використовуваної пам'яті.

Для вивчення теоретичного матеріалу як необхідний мінімум було рекомендовано посібник [10] через його компактність та доступність (матеріал адаптовано для загальноосвітньої школи). Недоліком даного посібника є те, що всі приклади в ньому написано на Python 2.x. Для більш детального вивчення рекомендувалися класичні підручники (за авторством Гвідо Ван Россума та Марка Лутца).

Висновки. Розроблений практикум використовувався для навчання студентів групи І-13 фізико-математичного факультету ДВНЗ «Криворізький національний університет». За результатами впровадження курсу можна сказати, що він загалом виконує поставлені перед ним завдання. Курс не лише формує базові навички програмування, а й озброює майбутніх фахівців сучасним потужним інструментом, володіння яким підвищує попит на спеціаліста на ринку праці.

Напрямки подальших досліджень. В майбутньому планується доповнити практикум роботами, орієнтованими на використання колекцій об'єктів Python, включити в роботи стислий теоретичний матеріал з прикладами на Python 3.x та підвищити інтеграцію з іншими курсами.

БІБЛІОГРАФІЯ

1. Проект Концепції розвитку освіти України на період 2015–2025 років [Електронний ресурс] / Міністерство освіти і науки України. – К. – Режим доступу: <http://old.mon.gov.ua/ua/pr-viddil/1312/1390288033/1414672797>.
2. Computer Science Curriculum 2013 [Electronic resource] / The Joint Task Force on Computing Curricula, Association for Computing Machinery, IEEE-Computer Society. – 2013 – Access mode : <http://www.acm.org/education/CS2013-final-report.pdf>.
3. Harvey B. What's Going On with 61A? [Electronic resource] / Brian Harvey – 2013 – Access mode : <https://www.cs.berkeley.edu/~bh/61a.html>.
4. Abelson H. Structure and Interpretation of Computer Programs / Harold Abelson and Gerald Jay Sussman with Julie Sussman ; The MIT Press Cambridge – 1996 – 588 p.
5. Emerick C. Why MIT now uses python instead of scheme for its undergraduate CS program [Electronic resource] / Chas Emerick – 2009 – Access mode: <http://cemerick.com/2009/03/24/why-mit-now-uses-python-instead-of-scheme-for-its-undergraduate-cs-program>.
6. Мерзликін П. Анкета першокурсника [Електронний ресурс] / [Павло Володимирович Мерзликін] – [Кривий Ріг], [2014]. – Режим доступу : <http://goo.gl/WZmj08>.
7. Мерзликін П. Матеріали з курсу “Програмування” [Електронний ресурс] / Мерзликін П. В. – Кривий Ріг, 2014. – Режим доступу : http://kdpu.edu.ua/index.php?option=com_content&task=view&id=5482&Itemid=14.
8. Златопольский Д. Сборник задач по программированию / Д. М. Златопольский – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2007. – 240с.
9. Юркин А. Задачник по программированию / А. Г. Юркин – СПб. : Питер, 2002. – 182с.
10. Хахаев И. Практикум по алгоритмизации и программированию на Python / И. А. Хахаев – М. : Альт Линукс, 2010. – 126 с.

ВІДОМОСТІ ПРО АВТОРІВ

Мерзликін Павло Володимирович – к. ф.-м. н., доцент кафедри інформатики та прикладної математики Криворізького педагогічного інституту ДВНЗ «Криворізький національний університет».

Коло наукових інтересів: нові матеріали в сучасній електроніці, методи комп'ютерного моделювання в фізиці твердого тіла, методика викладання інформатики у вищій школі.