

Оценка языка программирования Python в обучении как образовательный инструмент

Курбанкулова Махабат Салабатовна

Иссык-Кульский государственный университет им. Касыма Тыныстанова

Преподаватель

Приамурский государственный университет им. Шолом-Алейхема

Магистрант

Научный руководитель:

Баженов Руслан Иванович

Приамурский государственный университет им. Шолом-Алейхема

к.п.н., доцент, зав.кафедрой информационных систем, математики и правовой информатики

Аннотация

Целью данного исследования является анализ возможности внедрения языка программирования Python в учебные планы средних специальных учебных заведений. Проведен обзор данных о применении данного языка программирования, изучены достоинства и недостатки, распространенность использования. Результаты исследования приводят к положительной оценке внедрения изучения.

Ключевые слова: компьютерные языки, парадигма программирования, формальные языки, императивное программирование.

Evaluation of the Python programming language in learning as an educational tool

Kurbankulova Mahabat Calabatovna

Kasym Tynystanov Issyk-Kul State University

Lector

Sholom-Aleichem Priamursky State University

Master student

Scientific supervisor

Bazhenov Ruslan Ivanovich

Sholom-Aleichem Priamursky State University

Ph.D, Associate Professor, Head of the Department of Information Systems, Mathematics and Legal Informatics

Abstract

The purpose of this study is to analyze the possibility of introducing the Python programming language into the curricula of secondary specialized educational

institutions. A review of the data on the use of this programming language was carried out, the advantages and disadvantages, and the prevalence of use were studied. The results of the study lead to a positive assessment of the implementation of the study.

Keywords: computer languages, programming paradigm, formal languages, imperative programming.

1 Введение

1.1 Актуальность

Программирование становится необходимым компонентом современной учебной программы, а приобретенные с его помощью навыки будут необходимы гражданам технологического общества. Согласно учебному плану средней школы, поскольку старшеклассники изучают естественные науки, чтобы понять мир природы, им необходимо изучать информатику, чтобы понять социальную, экономическую и культурную среду информационного века. Независимо от того, чем ученики занимаются в своей будущей карьере, чем раньше они познакомятся с тем, что такое программирование, тем лучше для них. Еще один аргумент в пользу обучения программированию — поддержка существующей программы по математике в школах.

1.2 Обзор исследований

Скриптовые языки легче изучать и легче писать. Джон К. Оустерхаут, автор Tcl, комментирует в статье «Scripting: Higher Level Programming for the 21st Century» в журнале IEEE Computer [4]. Во-первых, уместно определение: скриптовый язык программирования — это язык программирования, предназначенный для быстрого решения проблем в тот момент, когда возникает необходимость, не требуя фазы проектирования «сверху вниз». Время выполнения в скриптовом языке обычно вдвое больше, чем в скомпилированном, а это означает, что для завершения программы требуется в два раза больше времени. С другой стороны, на реализацию той же идеи уходит примерно в два раза меньше времени даже у профессионалов, цитируя Л. Прехельта в отчете, сравнивающем семь языков программирования в определенном сценарии решения проблем [5].

В каждом случае версия сценария требовала меньше кода и времени на разработку, чем версия системного программирования; разница варьировалась от 2 до 60 раз. Разработка и написание программы на Perl, Python или Tcl занимает примерно вдвое меньше времени, чем ее написание на C, C++ или Java, а результирующая программа вдвое короче.

Ниже приводится мнение учителя Дж. Элкнера, основанное на его опыте работы с Python [6]: «Я учитель средней школы, который использует Python для преподавания нашего вводного курса уже два года. Я никогда не видел лучшего языка для этой цели. [...] Python хорош как язык обучения по тем же причинам, по которым вы сейчас используете его для программирования: он позволяет вам сосредоточиться на стоящей перед вами проблеме и не мешает вам. Эти особенности делают его выгодным для

школьных компьютерных центров и лабораторий». Тенденция к использованию таких сред в программировании также была отмечена Грином [7].

1.2 Цель исследования

Цель исследования — определить, можно ли применять Python в среднем специальном образовании в Кыргызстане и, если да, то при каких условиях и обстоятельствах.

2 Материалы и методы

Проведено обобщение педагогического опыта, анализ методологических основ профессиональной деятельности преподавателей вузов, рабочих программ и методических пособий.

3 Результаты и обсуждение

В реальном мире наиболее часто используемой моделью программирования является императивная. По этой причине именно императивное программирование преподается до сих пор в большинстве учебных программ средней школы. С другой стороны, объектно-ориентированное и функциональное программирование приобретают все большее значение в течение последних нескольких лет. Возникают опасения по поводу обучения другим парадигмам, кроме императивной, но проблема заключается в том, с чего начать. С этим связано и то, что учителя не являются учеными-компьютерщиками и не должны изучать несколько языков, чтобы преподавать один курс программирования. В идеале существует единый язык, который можно изучать постепенно и выборочно обучать определенной модели. Известно, что большинство существующих языков подходят для одной модели программирования. Python хорошо подходит для этого мышления, поскольку он объектно-ориентированный, что является супер императивом, с функциональными расширениями. Это означает, что кто-то может использовать все три парадигмы, используя один и тот же язык. Язык Python не навязывает уникальную парадигму [2].

Мы можем думать о парадигме как о методе моделирования, специально приспособленном для решения проблем в вычислительной среде. Существует несколько основных парадигм программирования, которые включают в себя, в порядке важности [1]:

- Императив, если программа представляет собой упорядоченное выполнение команд (операторов).
- Объектно-ориентированный, когда программа представляет собой набор объектов, которые взаимодействуют друг с другом.
- Функциональные, где программа представляет собой набор функций;
- Логический, где программа представляет собой набор «логических» объявлений.

Это общие семейства, на которые мы также классифицируем компьютерные языки. Вкратце, большинство компьютерных языков принадлежат императивной парадигме. Его эволюция - объектно-

ориентированная парадигма. Функциональная парадигма ближе к математическому мышлению и подходит для формальной спецификации программы.

Python — компьютерный язык, который обсуждается в последнее время как простой для изучения и отличный для освоения даже, по мнению ученых-компьютерщиков.

Утверждается, что логическая парадигма ближе к человеческой логике. Также, существуют языки, которые не совсем соответствуют этим парадигмам, например, специальные языки для параллельных вычислений или обратная польская запись. Последний адаптирован к внутренней структуре компьютера: он предполагает наличие специальной структуры данных, стека, в который поступают данные и выполняемые над ними операции.

Программирование часто рассматривают как решение проблем, а не как лингвистическую деятельность, часто игнорируя тот факт, что языки программирования являются случаем формальных языков. Формальные языки имеют характеристику, что их интерпретация - их значение - уникальны.

Математика, например, является формальным языком: в ней есть ряд символов, ряд определений, фиксированных понятий и отношений между ними. В некоторых случаях можно использовать основные определения. Формальные языки позволяют постуляционной модели развиваться детерминированным образом; и это верно независимо от области их применения.

Вы должны ожидать, что программа, написанная на определенном языке программирования, будет «работать» точно так же в любой компьютерной системе, а также будет понята каждым человеком, читающим ее, точно так же. Это свойство языков программирования делает их устойчивой основой для педагогических целей, поскольку в них остается мало места для двусмысленности. Практическое преимущество этого состоит в том, что для разрешения различий в способах понимания определенного «предложения» не требуется дополнительной коммуникации; вы можете сравнить это с тем, что происходит в естественных языках.

Мы не можем подтвердить то же свойство для естественных «разговорных» языков. Предложения многих, а может быть, и большинства естественных языков оставляют место для разнообразной интерпретации и могут означать разные вещи для разных людей. В 16-веке Макиавелли написал «Государь», учитывая это наблюдение.

Хотя существует скептицизм в отношении того факта, что обучение школьников программированию сопряжено с дополнительными накладными расходами для достижения минимального и необходимого уровня владения языком, признано, что использование ИТ для изучения математики, полезно, т.к. дает:

- новый способ просмотра или выражения многих математических понятий и отношений, например, коэффициенты и кратные, простота,

решение уравнений, преобразование функции в новую функцию, создание значений для параметров и переменных, оценка или графическое отображение функций или отношений, трансформационная геометрия, и так далее.

- новые средства для решения проблем, например, математическое моделирование и симуляция становятся средствами для понимания «реальных» проблем.

- выбранные среды программирования, например, компьютерные «микромиры» или динамические графические среды, могут рассматриваться как математические области сами по себе, т. е. компьютер становится средством выражения в том смысле, что «мысленные эксперименты» математиков могут быть превращены в вычислительный опыт для неспециалистов.

Мы можем добавить к этому видению, что до сих пор преподавание программирования не смогло проникнуть в образовательную систему (системы) в больших и системных масштабах. Это можно объяснить утверждением, что время является дефицитным ресурсом в современных школах. Школьные планы на самом деле посвящены очень конкретным целям, и учителям предоставляется небольшая свобода в выборе альтернативных путей обучения математике, не говоря уже о экспериментах с инструментами для этого. В настоящее время это меняется, поскольку в большинстве школ установлены компьютеры. Новые пути проложены, так почему бы не воспользоваться этой возможностью?

Прежде всего, мы можем рассматривать компьютерные языки, отделенные от их технологической основы, как средства коммуникации, точно так же, как и естественные языки.

Мы по-прежнему слишком легко думаем о программах как о предназначенных только для компиляции. Мы должны думать о них также как о способах общения с другими и о средствах для выражения наших собственных мыслей самим себе. Поэтому нам следует больше думать о чтении, а не о письме, о фиксации идей, а не о демонстрации идей.

Как студенты становятся экспертами в компьютерном языке? В принципе, кто-то должен освоиться с синтаксисом и семантикой языка в дополнение к среде, в которой он применяется.

Результаты использования определенных алгоритмов и известных структур программирования:

- Будет ли это подходящей идеей?
- Будет ли это эффективно?
- Достаточно быстро?
- Достаточно просто?
- Получу ли я ответ, который хочу?

Фундаментальные вопросы, касающиеся овладения естественными языками, также возникают при овладении компьютерными языками. Эти вопросы, как описано в книге «Знание языка» Н. Хомского [3], являются:

- Что представляет собой знание языка?

- Как приобретается знание языка?
- Как можно использовать знание языка?

Похоже, что принципы, лежащие в основе того, чтобы стать экспертом в компьютерном языке, - это принципы естественного языка плюс компонент решения проблем. Доказательство этого и ответы на эти же самые вопросы, касающиеся компьютерных языков, вполне могут быть предметом более серьезного исследования, чем этот исследовательский проект. Однако можно предположить, что существуют убедительные признаки того, что концепция универсальной грамматики также применима для понимания освоения компьютерных языков. Это суждение основано на наблюдении, что современные компьютерные языки задокументированы контекстно-свободными грамматиками в специальном метаязыке.

4 Выводы

По итогам исследования можно сделать следующие значимые выводы:

Python — это простой для изучения язык с понятным синтаксисом. Это приводит к небольшой длине кода, что означает небольшую вероятность ошибок. Заметны постоянные улучшения, по сравнению с существующими используемыми языками.

Самым ценным приобретением в научном или техническом образовании являются универсальные умственные инструменты, которые остаются полезными на всю жизнь. Можно отнести естественный язык и математику к наиболее важным из этих инструментов, а информатику — к третьему. Совместное изучение математики и информатики имеет педагогические преимущества, поскольку основные понятия каждого из них подкрепляют изучение другого.

Библиографический список

1. Кнут Д. Искусство программирования. Основные алгоритмы. М.: Вильямс, 2006. 720 с.
2. Downey A. Think python. O'Reilly Media, Inc., 2012.
3. Chomsky N. Knowledge of language: Its nature, origin, and use. Greenwood Publishing Group, 1986.
4. Ousterhout J. K. Scripting: Higher level programming for the 21st century //Computer. 1998. Т. 31. №. 3. С. 23-30.
5. Prechelt L. An empirical comparison of c, c++, java, perl, python, rexx and tcl //IEEE Computer. 2000. Т. 33. №. 10. С. 23-29.
6. Elkner J. Yorktown high school, Arlington, VA. USENET posting. URL: <http://mail.python.org/pipermail/edu-sig/2001-March/001045.html>
7. Green T. R. G. The nature of programming //Psychology of programming. Academic Press, 1990. С. 21-44.