

ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON В ПЕДАГОГИЧЕСКОМ ВУЗЕ: ОТ ОСНОВ ДО ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

К. В. Розов¹, А. В. Подсадников¹

¹ *Новосибирский государственный педагогический университет*
630126, Россия, г. Новосибирск, ул. Виллюйская, д. 28

Аннотация

В статье актуализируется выбор языка программирования Python для профессиональной подготовки будущих учителей информатики в педагогическом вузе. Python, являясь одним из наиболее актуальных языков программирования, внедряется в образование, и будущий учитель информатики должен быть готов эффективно применять его для обучения основам программирования. Кроме того, Python является одним из ведущих средств реализации перспективных на сегодняшний день технологий искусственного интеллекта, в связи с чем владеющий данным языком программирования педагог может стать частью механизма внедрения искусственного интеллекта в сферу образования.

Цель статьи — рассмотреть особенности организации курса обучения программированию на языке Python для профессиональной подготовки будущих учителей информатики.

Методология. В процессе разработки авторского курса обучения программированию на языке Python была проанализирована учебно-методическая литература по программированию на данном языке, а также статьи практикующих преподавателей и исследователей, описывающих свой опыт преподавания языка Python.

Результаты и заключение. Для обеспечения качественной подготовки будущих учителей информатики в Новосибирском государственном педагогическом университете был разработан курс обучения программированию на языке Python и апробирован в рамках дисциплины «Программирование». В ходе разработки курса были учтены типичные проблемы преподавания языка Python в школе и вузе, выявленные как авторами статьи, так и другими практикующими преподавателями программирования. Приведена структура разработанного курса с описанием составляющих его разделов. Представлены результаты опроса обучающихся третьего и четвертого курсов бакалавриата, завершивших изучение представленного курса, по ключевым вопросам, освещенным в данной статье. Респонденты отметили актуальность изучения языка Python и эффективность отдельных организационных особенностей курса. Раскрыта преемственность курса для дальнейшего обучения будущих учителей информатики актуальным в настоящее время технологиям искусственного интеллекта.

Ключевые слова: программирование, язык программирования, Python, искусственный интеллект, профессиональная подготовка, проблема обучения, обучающий курс.

DOI: 10.32517/0234-0453-2019-34-6-26-33

Для цитирования:

Розов К. В., Подсадников А. В. Язык программирования Python в педагогическом вузе: от основ до искусственного интеллекта // Информатика и образование. 2019. № 6. С. 26–33.

Статья поступила в редакцию: 20 июня 2019 года.

Статья принята к печати: 6 августа 2019 года.

Сведения об авторах

Розов Константин Владимирович, аспирант кафедры педагогики и психологии профессионального образования, ассистент кафедры информатики и дискретной математики. Новосибирский государственный педагогический университет, Россия; konstantin_dubrava@mail.ru; ORCID: 0000-0001-5231-8035

Подсадников Алексей Владимирович, старший преподаватель кафедры информатики и дискретной математики, Новосибирский государственный педагогический университет, Россия; cite2006@mail.ru

1. Особенности языка Python

История языка программирования Python начинается с 1980-х годов. Наиболее актуальная на сегодняшний день третья версия языка выпущена в декабре 2008 года [1]. В силу различных особенностей, которые выделяют Python среди других языков программирования, применяющихся в образовательных целях, язык используется на различных уровнях обучения как за рубежом [2], так и в России.

Особенностью языка программирования Python является то, что, в отличие от многих используемых в учебных заведениях языков программирования (Pascal, C, C++ и др.), в нем используется полностью динамическая типизация. Такой подход к типизации

избавляет от проблемы подбора типа данных при решении задач.

Также отличительной чертой Python являются его синтаксические особенности.

Во-первых, принятое в большинстве языков программирования требование заканчивать инструкции символом «точка с запятой» в Python отсутствует. Точка с запятой необходима только для разделения двух инструкций, записанных в одной строке. Данная особенность позволяет уменьшить количество ошибок, не связанных с правильностью решения задачи. Часто с точки зрения понимания программного кода человеком использование точки с запятой избыточно, так как в основном каждая инструкция записывается с новой строки и нет видимой необ-

Pascal	C++	Python
<pre>if a mod 2 = 0 then writeln('Четное') else writeln('Нечетное');</pre>	<pre>if (a % 2 == 0) cout << "Четное"; else cout << "Нечетное";</pre>	<pre>if a % 2 == 0: print('Четное') else: print('Нечетное')</pre>

Рис. 1. Решение одной и той же задачи на трех языках программирования

ходимости разделения инструкций специальным символом. Вместе с тем можно назвать избыточными и некоторые правила расстановки таких символов, тем более что в разных языках эти правила могут отличаться. В качестве примера приведем фрагмент кода решения одной задачи на трех популярных для обучения языках программирования (рис. 1).

В языке Pascal точку с запятой перед *else* ставить запрещено, в то время как в языке C++ она необходима. В языке Python в данном случае добавление точки с запятой перед *else* не обязательно, так как она не выполняет своей функции разделения инструкций.

Во-вторых, в языке отсутствуют операторные скобки для объединения операторов в единый блок, их роль выполняет отступ, одинаковый для каждой инструкции внутри блока. Такая особенность делает код более компактным, не снижая его читаемость. Кроме того, требование добавления отступов обеспечивает необходимость выравнивания кода. Практика преподавания программирования в школе и вузе показывает, что многие учащиеся пренебрегают выравниванием кода в своих программах, что негативно сказывается на его читаемости. Язык Python фактически вынуждает выравнивать код, тем самым дисциплинируя учащегося, развивая у него соответствующий элемент культуры программирования.

Третья особенность касается построения составных логических выражений. Синтаксис языка программирования Python поддерживает записи условий типа $a < b < c < \dots < n$ в их явном виде, в то время как во многих других языках программирования нужно разбивать подобные логические выражения на простые, бинарные условия и объединять их при помощи логического оператора «И». На рисунке 2 приведен пример записи одного условия на трех языках программирования.

Вышеперечисленные и некоторые другие особенности языка Python позволили ему войти в ряд общепринятых для обучения языков программирования. Дальнейший рост популярности языка в образовательных учреждениях обеспечил его включение в контрольно-измерительные материалы для проведения государственной итоговой аттестации, а также в список допустимых языков программирования на олимпиадах. «Включение Python в языки выбора для решения задач государственной итоговой аттестации де-факто потребовало разработки соответствующих

курсов для подготовки будущих учителей информатики» [3, с. 282]. Авторы учебников по информатике для средней школы выпускают учебный материал по основам программирования на языке Python [4].

2. Проблемы преподавания языка Python

Но, несмотря на многочисленные упрощения, особенности языка Python создают и некоторые проблемы в обучении программированию. Существует мнение, что Python нежелательно изучать в качестве первого языка программирования [5, 6].

Опыт преподавания авторами языка Python в Новосибирском государственном педагогическом университете и в гимназии № 4 г. Новосибирска позволил выявить следующие проблемы.

Одна из проблем связана непосредственно с динамической типизацией. Д. А. Васильев также указывает на данную проблему [7]. Часто обучающиеся забывают о том, что функция ввода данных *input* в Python всегда возвращает строковое значение, в то время как, например, в Pascal процедура ввода данных *readln* является универсальной, т. е. передает данные, соответствующие типу переменной, переданной в качестве параметра. Это приводит к ошибке типа данных (*TypeError*) или логической ошибке. Если в первом случае ошибку выявить достаточно просто, о ее наличии сообщит интерпретатор, то логическую ошибку выявить гораздо сложнее, так как программа будет выполнена, но результат ее работы будет неверным. Приведем пример программы, содержащей логическую ошибку, связанную с динамической типизацией:

```
gradebook = {input('Фамилия: '):
  input('Оценка: ') for i in range(10)}
grades = {}.fromkeys([1,2,3,4,5], 0)

for key in gradebook:
  if gradebook[key] in grades.keys():
    grades[gradebook[key]] += 1

print(grades)
```

В данном примере приведен код программы решения следующей задачи из авторского курса программирования на языке Python: «Дана таблица оценок в пятибалльной системе по информатике в классе из 10 учеников. Вывести количество каждой оценки. Введенные оценки, не входящие в пятибалльную систему, учитываться не должны».

Pascal	C++	Python
<pre>if (a>0) and (a<5) then</pre>	<pre>if (a > 0 && a < 5)</pre>	<pre>if 0 < a < 5:</pre>

Рис. 2. Запись одного и того же условия на трех языках программирования

Ошибка в коде программы связана с тем, что в словарь *gradebook* оценки вводятся в строковом формате, а ключи в словаре *grades*, соответствующие оценкам, в числовом. При попытке поиска числового значения оценки в списке ключей словаря *grades* оценка не может быть найдена из-за несоответствия типов данных.

Другая проблема возникает при изучении цикла *for*. В отличие от многих других языков в языке Python цикл *for* выполняет функцию перебора элементов некоторой последовательности, а не изменения значения параметра на каждой итерации. С одной стороны, цикл позволяет обращаться к элементу последовательности не только по его индексу, но и напрямую, что делает его более гибким в использовании. С другой стороны, обучающиеся часто не понимают разницы между этими двумя способами обращения к элементу либо не понимают, в каком случае какой из способов нужно применить, вследствие чего, даже имея алгоритм решения задачи, они не могут написать реализующий его код программы.

Тем не менее приведенные выше проблемы при обучении языку не снижают его достоинств как языка для обучения программированию. Педагогам следует лишь обращать особое внимание обучающихся на указанные нюансы.

3. Курс программирования для педагогического вуза

Актуальность дисциплин в педагогическом вузе определяется их соответствием образовательным тенденциям. В связи с этим при написании рабочей программы и подготовке материала для проведения занятий по дисциплине «Программирование» в качестве изучаемого языка было принято решение использовать Python.

Курс ориентирован на бакалавров направлений подготовки 44.03.01 «Педагогическое образование» и 44.03.05 «Педагогическое образование (с двумя профилями подготовки)» специальностей, связанных с подготовкой будущих учителей информатики: информатика и ИКТ; информатика и ИКТ и экономическое образование; математика и информатика.

Курс разработан на основе открытой системы управления обучением Moodle, популярной при организации дистанционного обучения в вузах [3, 8]. Данная система позволяет обеспечить доступ обучающихся к курсу в любое удобное для них время.

Разработанный курс состоит из 14 разделов, каждый из которых включает в себя три компонента: теоретическая часть, примеры решения задач, блок задач для практической работы.

В теоретической части описываются особенности синтаксиса, команды, языковые конструкции, способы их применения. Описание сопровождается иллюстрирующими учебный материал фрагментами программного кода.

Примеры решения задач дополняют теоретическую часть с целью демонстрации тех приемов или

особенностей языка, которые эффективнее продемонстрировать на практике, чем словесно описать.

Практическая часть содержит задачи различного уровня сложности. Для каждой задачи явно сформулированы входные и выходные данные, что способствует устранению неоднозначности в понимании ее формулировки.

Прежде чем подробно рассматривать разделы курса, приведем их полный список в порядке следования:

1. Синтаксис языка Python, структура программы, типы данных, ввод и вывод.
2. Числа в Python. Арифметические операции, математические функции, модуль *Math*, функции случайных чисел.
3. Алгоритм ветвления. Управляющие конструкции в Python.
4. Циклический алгоритм в Python.
5. Строки в Python.
6. Списки и кортежи в Python.
7. Словари в Python.
8. Множества в Python.
9. Функции в Python. Создание и подключение модулей в Python.
10. Работа с файлами в Python.
11. Основы объектно-ориентированного программирования в Python.
12. Графика в Python. Модуль *Turtle*.
13. Обработка исключений в Python.
14. Математические вычисления в Python. Библиотека символьных вычислений *SymPy*.

Первый раздел «Синтаксис языка Python, структура программы, типы данных, ввод и вывод данных» посвящен изучению отличительных синтаксических особенностей языка Python. Внимание уделяется таким особенностям, как система отступов при написании кода, разбирается общая структура программы. Так как Python — язык программирования с полностью динамической типизацией, во избежание дальнейших проблем с типами данных данный вопрос рассматривается достаточно подробно. Также по сравнению, например, с языком Pascal, в Python более гибко реализована система ввода и вывода данных. Чтобы при изучении последующих разделов не отвлекаться на основные синтаксические особенности, вводу и выводу данных уделяется достаточно много внимания.

Ввиду того что курс в большей степени разрабатывался для специальностей, имеющих непосредственное отношение к математике, принято решение **второй раздел «Числа в Python. Арифметические операции, математические функции, модуль *Math*, функции случайных чисел»** посвятить числам и основным операциям над ними. В данном блоке изучается модуль *Math*, так как именно в нем описаны основные математические функции для работы с действительными числами. Также в этом блоке уделяется внимание такой особенности языка Python, как работа с комплексными числами. При решении различных математических задач зачастую необходимо тестировать задачу на большом количестве

различных данных, поэтому именно в этот блок мы добавили работу со случайными числами — модуль *Random*.

Наш опыт преподавания программирования на других языках показал, что понятие проверки условий гораздо быстрее усваивается учащимися именно на примере математических выражений. На уроках математики в школе достаточно много времени уделяется решению неравенств. Поэтому **изучение разветвляющегося алгоритма** решено проводить сразу же после математического блока. Здесь рассматривается отличительная особенность построения условий в Python, возможность записи условий вида $a < b < c < \dots < n$ без разделения их на простые выражения. Рассматривается тип данных *bool* и тот важный факт, что в Python значение *True* возвращает любой непустой и ненулевой объект, а значение *False* — любой пустой либо нулевой объект.

Следующий блок направлен на **изучение циклических алгоритмов**. Нет единого мнения по поводу того, какой из операторов цикла изучать первым — *for* или *while*. Например, в [4, 9] предлагается сначала изучать оператор *for*, а в учебном пособии [10] и факультативном курсе для будущих учителей информатики [11] — оператор *while*. Так как функционирование цикла со счетчиком в Python заметно отличается от аналогичных циклов, например, в популярных при обучении программированию языках Pascal и C++, а функционирование циклов *while* практически идентично, авторами принято решение сначала изучать цикл с предусловием *while*. Затем рассматривается функция генерации последовательностей *range()*. И только познакомившись с функцией *range()*, приступаем к рассмотрению цикла со счетчиком *for*. Такой порядок выбран, исходя из специфики построения цикла *for*. В отличие от многих других языков программирования, в Python в цикле *for* задается не счетчик и его шаг, а последовательность, элементы которой будет перебирать счетчик. Также в этом блоке рассматриваются операторы *break* и *continue*. Здесь же обращается внимание на отличающую Python возможность применения в цикле инструкции *else*.

В связи с тем что строки в Python являются базовым типом вводимых данных и при этом итерируемой последовательностью, **изучение раздела «Строки в Python»** следует за изучением циклического алгоритма и операторов цикла. При изучении данной темы уделено внимание рассмотрению функций и методов работы со строками, а также циклической обработке строк. Делается акцент на то, что строки в Python являются неизменяемым типом данных.

В следующем **разделе «Списки и кортежи в Python»** авторами было принято решение объединить изучение двух структур данных: *list* (списки) и *tuple* (кортежи). Связано это с тем, что эти типы данных схожи между собой и представляют упорядоченные последовательности. Различие между списком и кортежем состоит в том, что список — это изменяемая последовательность, а кортеж — неизменяемая последовательность. Основной упор

делается на изучение методов работы со списками. Для кортежей применяются методы работы со списками, которые не изменяют сам список. В Python списки используются как альтернатива стандартным массивам в языках программирования Pascal и C++. Так как обычно в рамках курсов программирования рассматриваются одномерные и двумерные массивы, в разработанном курсе изучаются вложенные списки как альтернатива двумерным массивам.

Следующей изучается **структура данных *dict* (словарь)**. По сути, словарь представляет собой ассоциативный массив, т. е. массив, в котором данные хранятся в виде пар ключ—значение. Обращение к значению в словаре происходит не по номеру ячейки, а по ключу (имени). В разделе разобраны способы создания словарей, функции и методы работы со словарями. Отдельное внимание уделяется рассмотрению ситуаций, в которых выгоднее использовать список вместо словаря и наоборот.

Далее в курсе изучается **тема «Множества»**. Множество представляет собой неупорядоченный набор уникальных объектов. Для организации работы с множествами в Python существуют структуры данных *set* (изменяемое множество) и *frozenset* (неизменяемое множество). В рамках данной темы рассматриваются встроенные в Python операторы и методы для работы с множествами. Поясняется, что методам соответствуют определенные операторы, позволяющие сократить запись инструкций.

Следующей после изучения основных типов и структур данных является **тема «Функции»**. Функция — это блок организованного, многократно используемого кода, который используется для выполнения конкретной задачи. Функции обеспечивают модульность приложения и значительно повышают эффективность повторного использования кода. Элементами содержания темы являются: функции, возвращающие (с ключевым словом *return*) и не возвращающие значения, типы аргументов (обязательные аргументы, аргументы — ключевые слова, аргументы по умолчанию, аргументы произвольной длины), области видимости переменных, рекурсия, правила подключения внешних и создание собственных модулей. Дополнительными элементами являются лямбда-функции и декораторы.

Так как описываемый курс ориентирован на применение в педагогическом вузе, изучение следующей **темы «Работа с файлами»** ограничивается лишь текстовыми файлами. Умения работать с текстовыми файлами достаточно для решения основных учебных и олимпиадных задач.

Объектно-ориентированное программирование в настоящее время является ведущей парадигмой программирования на императивных языках. Поэтому в нашем курсе уделяется внимание основам объектно-ориентированного программирования. Рассматриваются такие понятия, как класс, экземпляр класса (объект), поле, метод, конструктор. Углубленное изучение объектно-ориентированного программирования на Python подразумевает отдельную дисциплину.

В связи с тем что данный курс рассчитан на подготовку будущих учителей информатики, **графические возможности Python** ограничены рассмотрением только модуля *Turtle*. Опыт отечественных [12] и зарубежных [13] преподавателей показывает эффективность применения данного модуля для формирования базовых навыков текстового программирования. В данном модуле реализован учебный алгоритмический исполнитель Черепаша, основанный на языке Logo [14], предназначенный для обучения детей младшего школьного возраста основам алгоритмизации и программирования. Умение работать с модулем *Turtle* расширит инструментарий будущих учителей информатики для работы с обучающимися в рамках пропедевтического курса информатики в школе.

Как и во многих других языках программирования, в Python есть специальный **инструмент для обработки исключений**: *try*. По нашему мнению, данный инструмент должен быть рассмотрен в рамках курса для формирования у обучающихся полноценного представления о возможностях современных языков программирования. Было бы полезно рассмотреть данный инструмент в начале курса, так как уже на этапе изучения ввода данных могут возникнуть ошибки, которые можно обработать как исключения. Но, в связи с тем что курс в первую очередь направлен не на непосредственное изучение языка программирования Python как профессионального инструмента, а на обучение алгоритмизации и программированию, принято решение рассмотреть обработку исключений после изучения алгоритмов, операторов, типов и структур данных.

Когда основы алгоритмизации и программирования на языке Python уже изучены обучающимися, логично показать им **возможности практического применения языка в различных сферах деятельности с использованием специализированных средств**.

Обучающиеся по специальностям, предполагающим профильное изучение математических дисциплин, ввиду специфики их учебной деятельности регулярно сталкиваются с необходимостью проведения математических расчетов, в том числе **символьных вычислений**. Используя Python, можно упростить процесс выполнения расчетов, не прибегая к специализированным математическим пакетам. Существует несколько модулей компьютерной математики для Python. В нашем курсе рассматривается **модуль SymPy**, который позволяет решать как учебные, так и прикладные исследовательские задачи [15]. Для демонстрации возможностей *SymPy* были подобраны задачи следующих типов: решение алгебраических уравнений и систем уравнений, дифференцирование, интегрирование, вычисление предела функции, вычисление суммы ряда, операции над матрицами.

4. Искусственный интеллект для будущих учителей

В настоящее время перспективным направлением развития информационных технологий является искусственный интеллект (ИИ). В работах как от-

ечественных [16–18], так и зарубежных [19–20] исследователей отмечается возможность применения ИИ в различных сферах человеческой деятельности, в том числе в образовании. В педагогических вузах России для подготовки будущих учителей информатики традиционно выделяется отдельная дисциплина для знакомства обучающихся с основами ИИ, так как, безусловно, современный учитель информатики должен ориентироваться в современных информационных технологиях и тенденциях их развития.

Язык Python широко применяется профессиональными программистами в области ИИ. Знание основ языка Python позволит будущим учителям информатики изучать технологии ИИ через использование специализированных библиотек, упрощающих применение таких технологий. Компьютерное зрение, обработка естественного языка, машинное обучение могут быть доступны для людей, не обладающих специфическими знаниями в области ИИ. В то время как роль технологий ИИ в жизни людей возрастает, будущие учителя информатики могут быть включены в процесс внедрения ИИ в образование, находя ему применение в данной сфере, используя технологии ИИ в своих проектах, вызывая интерес к перспективным технологиям у детей.

5. Исследование мнения студентов

Мы провели опрос обучающихся третьего и четвертого курсов бакалавриата, прошедших обучение по рассматриваемому курсу программирования на языке Python. Опрос был проведен в форме анкетирования. Анкета содержала девять вопросов, предполагающих ответы «да» или «нет»:

1. Считаете ли вы актуальным изучение Python в школе?
2. Можно ли изучать Python в качестве первого языка программирования?
3. Может ли язык Python полностью заменить Pascal в школе?
4. Облегчает ли, по вашему мнению, динамическая типизация написание программы на Python?
5. Облегчает ли понимание условия задачи явное указание в нем вводимых и выводимых данных?
6. Целесообразно ли, по вашему мнению, изучение оператора *while* перед оператором *for* при изучении циклических алгоритмов, как это было реализовано в преподаваемом вам курсе программирования на Python?
7. Считаете ли вы эффективным изучение работы со строками в Python раньше, чем с другими структурами данных?
8. Считаете ли вы полезным изучение технологий искусственного интеллекта в педагогическом вузе?
9. Известно ли вам, что существуют библиотеки для языка Python, реализующие технологии искусственного интеллекта?

Вопросы 2 и 3 предполагали возможность прокомментировать свой выбор варианта ответа.

Результаты проведения опроса отображены на диаграмме (рис. 3).



Рис. 3. Соотношение ответов на вопросы анкеты

Изучение Python в школе считают актуальным 100 % респондентов.

Большинство респондентов (87,5 %) считают, что «да», язык Python можно изучать в качестве первого языка программирования. Наиболее часто, аргументируя свой ответ в пользу данного выбора, респонденты отмечали, что язык Python «легче», чем язык Pascal, обладает более простым синтаксисом и он «простой в понимании». Респонденты, ответившие «нет», указали на возможную проблему перехода на другие языки программирования после изучения Python. На наш взгляд, большинство обучающихся, отвечая на вопрос, формулировали свою точку зрения с позиции программиста, а не учителя, уделив недостаточно внимания проблеме обучения программированию.

При ответе на вопрос 3 мнения респондентов разделились практически поровну. 54,17 % считают, что Python может полностью заменить Pascal при изучении программирования в школе, ссылаясь на актуальность Python и устаревание Pascal. 45,83 % сомневаются в возможности полной замены языка Pascal языком Python, указывая на необходимость изучения двух этих языков. Мы полагаем, что под необходимостью изучения двух языков студенты понимали необходимость формирования у учеников представления о языках разной степени «строгости».

Динамическая типизация облегчает написание программы по мнению 91,67 % респондентов.

Явное указание вводимых и выводимых данных в условиях заданий курса оправдало себя, так как 100 % респондентов отметили, что такой способ представления условия задачи облегчает ее понимание.

Целесообразность изучения оператора *while* перед оператором *for* подтвердила большая часть респондентов (66,67 %).

Эффективность изучения работы со строками в Python раньше, чем с другими структурами данных, в рамках курса также подтвердили большинство респондентов (83,33 %).

87,5 % респондентов считают полезным изучение технологий искусственного интеллекта в педагогическом вузе, что подтверждает актуальность наличия соответствующего курса в учебном плане. В то же время о существовании библиотек для языка Python, реализующих технологии искусственного интеллекта, известно лишь 29,17 % респондентов.

6. Выводы

Обучающиеся осознают актуальность изучения языка Python, однако существует проблема «преобладания моды над методикой». Вопрос эффективности использования Python в школе как первого и, возможно, единственного языка для обучения основам программирования остается открытым, и его однозначно стоит обсуждать с будущими учителями информатики в ходе освоения ими курса программирования на языке Python для формирования у них критического отношения к выбору языка с точки зрения методики преподавания информатики. Большинство обучающихся были приняты особенности построения описанного в статье курса программирования. Разработанный нами курс обучения программированию на языке Python для педагогического вуза ввиду подробного рассмотрения основ языка имеет преимущество для изучения информационных технологий с интеграцией языка Python, в том числе актуальных на сегодняшний день технологий ИИ.

Список использованных источников

1. Python 3.0 Release. <https://www.python.org/download/releases/3.0/>
2. Hromkovic J., Kohn T., Komm D., Serafini G. Combining the power of Python with the simplicity of Logo for a sustainable computer science education // Informatics in Schools: Improvement of Informatics Knowledge and Perception. 2016. P. 155–166. DOI: 10.1007/978-3-319-46747-4_13
3. Горбатов С. В., Добудько А. В., Добудько Т. В., Пугач О. И. Организационно-методические вопросы разработки и эксплуатации электронных курсов в педагогическом вузе (на примере электронного курса «Программирование на Python») // Балтийский гуманитарный журнал. 2017. Т. 6. № 4. С. 281–284.
4. Поляков К. Ю. Программирование. Python. C++. Ч. 1: учебное пособие. М.: БИНОМ. Лаборатория знаний, 2019. 144 с.
5. Иванов А. А., Рожкова Л. Искусственный интеллект как основа инновационных преобразований в технике, экономике, бизнесе // Известия СПбГЭУ. 2018. № 3. С. 112–115. https://unecon.ru/sites/default/files/izvestiya_3-2018.pdf
6. Поляков К. Ю. Язык Python глазами учителя // Информатика. 2014. № 9. С. 4–16.
7. Васильев Д. А. Методические особенности изучения языка Python школьниками // Символ науки. 2017. Т. 1. № 1. С. 170–172. <https://os-russia.com/SBORNIKI/SN-17-01-1.pdf>
8. Барабанова П. С., Лямкин Р. А., Макарова И. С. Разработка электронного курса по программированию для самоподготовки студентов технического вуза // Бюллетень лаборатории математического, естественнонаучного образования и информатизации. Материалы Международной научно-практической конференции «Математическое, естественнонаучное образование и информатизация». М.: МГПУ, 2015. С. 62–66.

9. Шелудько В. М. Основы программирования на языке высокого уровня Python. Ростов-на-Дону; Таганрог: ЮФУ, 2017. 146 с.

10. Сысоева М. В., Сысоев И. В. Программирование для нормальных с нуля на языке Python: В 2 ч. Ч. 1. М.: Базальт СПО; МАКС Пресс, 2018. 176 с.

11. Пименова А. Н. Обучение будущих учителей информатики основам работы с языком программирования Python // Преподавание информационных технологий в Российской Федерации. Материалы Шестнадцатой открытой Всероссийской конференции. М.: МГТУ им. Н. Э. Баумана, 2018. С. 163–164. <http://www.it-education.ru/conf2017/thesis/2712/>

12. Сорокина Т. Е. Использование графической библиотеки Turtle Graphics языка Python для плавного перехода от блочного программирования к текстовому // Информатика в школе. 2018. № 3. С. 6–10.

13. Fagan B. J. M. Ed., Payne B. Learning to program in Python — by teaching it! // Proc. Interdisciplinary STEM Teaching and Learning Conf. 2017. Vol. 1. Article 9. P. 99–107. DOI: 10.20429/stem.2017.010109

14. Пейперт С. Переворот в сознании: Дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989. 224 с.

15. Якимов И. М., Кирпичников А. П., Конюхов А. А., Седов И. А. Имитационное моделирование вероятностных

объектов в библиотеке SimPy языка программирования Python // Вестник технологического университета. 2019. Т. 22. № 3. С. 134–137.

16. Васильев Д. А. К вопросу изучения языка программирования Python в школе // Актуальные исследования в области математики, информатики, физики и методики их изучения в современном образовательном пространстве. Курск: КГУ, 2016. С. 80–83.

17. Самсонович О. О., Фокина Е. А. Искусственный интеллект — новые реалии // Международный журнал прикладных и фундаментальных исследований. 2018. № 5-1. С. 257–263. <https://applied-research.ru/ru/article/view?id=12253>

18. Уваров А. Ю. Технологии искусственного интеллекта в образовании // Информатика и образование. 2018. № 4. С. 14–22.

19. Edwards C., Edwards A., Spence P. R., Lin X. I, teacher: Using artificial intelligence (AI) and social robots in communication and instruction // Communication Education. 2018. Vol. 67. Is. 4. P. 473–480. DOI: 10.1080/03634523.2018.1502459

20. Goksel N., Bozkurt A. Artificial intelligence in education: Current insights and future perspectives // Handbook of Research on Learning in the Age of Transhumanism. Hershey, PA: IGI Global, 2019. P. 224–236.

PYTHON PROGRAMMING LANGUAGE IN PEDAGOGICAL UNIVERSITY: FROM THE BASES TO ARTIFICIAL INTELLIGENCE

K. V. Rozov¹, A. V. Podsadnikov¹

¹ Novosibirsk State Pedagogical University
630126, Russia, Novosibirsk, Vilyuiskaya ul., 28

Abstract

The article actualizes the choice of Python programming language for professional training of future teachers of informatics in pedagogical university. Python, being one of the most relevant programming languages, is being introduced into education, and the future informatics teacher should be ready to effectively use it to teach the basics of programming. In addition, the Python language is one of the leading tools for the implementation of advanced technologies of artificial intelligence, and therefore, owning this programming language teacher can be part of the mechanism of implementation of artificial intelligence in education.

The objective of the article is to consider the features of the organization of the course of programming in Python for the training of future teachers of informatics.

Methodology. During the development of the author's course of programming in Python educational literature on programming in this language was analyzed, as well as articles by practicing teachers and researchers describing their experience of teaching Python.

Results and conclusion. To ensure quality training of future teachers of informatics at Novosibirsk State Pedagogical University a course of programming in Python was developed and tested in the discipline "Programming". During the development of the course, typical problems of teaching Python at school and university were taken into account, identified by both the authors of the article and other practicing programming teachers. The structure of the developed course with the description of its constituent sections is given. The article presents the results of a survey of students of 3d and 4th undergraduate courses, who completed the presented course, on the key issues covered in the article. Respondents noted the relevance of learning Python and the effectiveness of certain organizational features of the course. The continuity of the course for further training of future teachers of informatics to the current technologies of artificial intelligence is revealed.

Keywords: programming, programming language, Python, artificial intelligence, training, problem of teaching, training course.

DOI: 10.32517/0234-0453-2019-34-6-26-33

For citation:

Rozov K. V., Podsadnikov A. V. Yazyk programmirovaniya Python v pedagogicheskom vuze: ot osnov do iskusstvennogo intellekta [Python programming language in pedagogical university: From the bases to artificial intelligence]. *Informatika i obrazovanie — Informatics and Education*, 2019, no. 6, p. 26–33. (In Russian.)

Received: June 20, 2019.

Accepted: August 6, 2019.

About the authors

Konstantin V. Rozov, Postgraduate Student at the Department of Pedagogy and Psychology of Vocational Education, Assistant at the Department of Computer Science and Discrete Mathematics, Novosibirsk State Pedagogical University, Russia; konstantin_dubrava@mail.ru; ORCID: 0000-0001-5231-8035

Alexey V. Podsadnikov, Senior Lecturer at the Department of Computer Science and Discrete Mathematics, Novosibirsk State Pedagogical University, Russia; cite2006@mail.ru

References

1. Python 3.0 Release. Available at: <https://www.python.org/download/releases/3.0/>
2. Hromkovic J., Kohn T., Komm D., Serafini G. Combining the power of Python with the simplicity of Logo for a sustainable computer science education. *Informatics in Schools: Improvement of Informatics Knowledge and Perception*, 2016, p. 155–166. DOI: 10.1007/978-3-319-46747-4_13
3. Gorbatov S. V., Dobudko A. V., Dobudko T. V., Pugach O. I. Organizatsionno-metodicheskie voprosy razrabotki i ehkspluatatsii ehlektronnykh kursov v pedagogicheskom vuze (na primere ehlektronnogo kursa “Programmirovaniye na Python”) [Organizational-methodological issues of development and exploitation of e-learning in a pedagogical university (on the example of the e-learning course “Programming in Python”)]. *Baltiyskiy gumanitarnyy zhurnal — Baltic Humanitarian Journal*, 2017, vol. 6, no. 4, p. 281–284. (In Russian.)
4. Polyakov K. Yu. Programmirovaniye. Python. C++. Chast’ 1: Uchebnoe posobie [Programming. Python. C++. Part 1: Study Guide]. Moscow, BINOM. Laboratoriya znaniy, 2019. 144 p. (In Russian.)
5. Ivanov A. A., Rozhkova L. Iskusstvennyy intellekt kak osnova innovatsionnykh preobrazovaniy v tekhnike, ehkonomie, biznese [Artificial intelligence as the basis of innovative transformations in technology, economics, business]. *Izvestiya SPbGEHU — SPSUE news*, 2018, no. 3, p. 112–115. (In Russian.) Available at: https://unecon.ru/sites/default/files/izvestiya_3-2018.pdf
6. Polyakov K. Yu. Yazyk Python glazami uchitelya [Python language through the eyes of a teacher]. *Informatika — Informatics*, 2014, no. 9, p. 4–16. (In Russian.)
7. Vasiliev D. A. Metodicheskie osobennosti izucheniya yazyka Python shkol’nikami [Methodological features of learning Python language by schoolchildren]. *Simvol nauki — Symbol of Science*, 2017, vol. 1, no. 1, p. 170–172. (In Russian.) Available at: <https://os-russia.com/SBORNIKI/SN-17-01-1.pdf>
8. Barabanova P. S., Lyamkin R. A., Makarova I. S. Razrabotka ehlektronnogo kursa po programmirovaniyu dlya samopodgotovki studentov tekhnicheskogo vuza [The development of the electronic course on programming for independent learning of technical university students]. *Byulleten’ laboratorii matematicheskogo, estestvennonauchnogo obrazovaniya i informatizatsii. Materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii “Matematicheskoe, estestvennonauchnoe obrazovanie i informatizatsiya” [Bulletin of the Laboratory for Mathematical, Natural Science Education and Informatization. Proc. Int. Scientific-Practical Conf. “Mathematical, Natural Science Education and Informatization”]*. Moscow, MGPU, 2015, p. 62–66. (In Russian.)
9. Sheludko V. M. Osnovy programmirovaniya na yazyke vysokogo urovnya Python [Python high level programming basics]. Rostov-on-Don; Taganrog, SFedU, 2017. 146 p. (In Russian.)
10. Sysoeva M. V., Sysoev I. V. Programmirovaniye dlya normal’nykh s nulya na yazyke Python: V dvukh chastyakh. Chast’ 1 [Programming for normal from scratch in Python: In two parts. Part 1]. M.: Bazal’t SPO; MAKSS Press, 2018. 176 p. (In Russian.)
11. Pimenova A. N. Obuchenie budushchikh uchitelej informatiki osnovam raboty s yazykom programmirovaniya Python [Teaching basics of programming language Python to future IT teachers]. *Prepodavaniye informatsionnykh tekhnologiy v Rossiyskoy Federatsii. Materialy Shestnadsatoj otkrytoj Vserossiyskoy konferentsii [Teaching information technology in the Russian Federation. Proc. Sixteenth Open All-Russ. Conf.]*. Moscow, BMSTU, 2018, p. 163–164. (In Russian.) Available at: <http://www.it-education.ru/conf2017/thesis/2712/>
12. Sorokina T. E. Ispol’zovaniye graficheskoy biblioteki Turtle Graphics yazyka Python dlya plavnogo perekhoda ot blochnogo programmirovaniya k tekstovomu [Using the Turtle Graphics Python library to seamlessly transition from block programming to text programming]. *Informatika v shkole — Informatics in School*, 2018, no. 3, p. 6–10. (In Russian.)
13. Fagan B. J. M. Ed., Payne B. Learning to program in Python — by teaching it! *Proc. Interdisciplinary STEM Teaching and Learning Conf*, 2017, vol. 1, article 9, p. 99–107. DOI: 10.20429/stem.2017.010109
14. Pejbert S. Perevorot v soznanii: Deti, komp’yutery i plodotvornyye idei [Revolution in mind: children, computers and fruitful ideas]. Moscow, Pedagogika, 1989. 224 p. (In Russian.)
15. Yakimov I. M., Kirpichnikov A. P., Konyukhov A. A., Sedov I. A. Imitatsionnoye modelirovaniye veroyatnostnykh ob’ektov v biblioteke SimPy yazyka programmirovaniya Python [Simulation of probabilistic objects in the SimPy library of the Python programming language]. *Vestnik tekhnologicheskogo universiteta — University of Technology Herald*, 2019, vol. 22, no. 3, p. 134–137. (In Russian.)
16. Vasiliev D. A. K voprosu izucheniya yazyka programmirovaniya Python v shkole [To the question of learning Python programming language at school]. *Aktual’nye issledovaniya v oblasti matematiki, informatiki, fiziki i metodiki ikh izucheniya v sovremennom obrazovatel’nom prostranstve [Actual research in the field of mathematics, computer science, physics and methods of their study in the modern educational space]*. Kursk, KSU, 2016, p. 80–83. (In Russian.)
17. Samsonovich O. O., Fokina E. A. Iskusstvennyy intellekt — novyye realii [Artificial intelligence — new scope]. *Mezhdunarodnyy zhurnal prikladnykh i fundamental’nykh issledovaniy — International Journal of Applied and Fundamental Research*, 2018, no. 5-1, p. 257–263. (In Russian.) Available at: <https://applied-research.ru/ru/article/view?id=12253>
18. Uvarov A. Yu. Tekhnologii iskusstvennogo intellekta v obrazovanii [Artificial intelligence technologies in education]. *Informatika i obrazovanie — Informatics and Education*, 2018, no. 4, p. 14–22. (In Russian.)
19. Edwards C., Edwards A., Spence P. R., Lin X. I, teacher: using artificial intelligence (AI) and social robots in communication and instruction. *Communication Education*, 2018, vol. 67, is. 4, p. 473–480. DOI: 10.1080/03634523.2018.1502459
20. Goksel N., Bozkurt A. Artificial intelligence in education: Current insights and future perspectives. *Handbook of Research on Learning in the Age of Transhumanism*. Hershey, PA, IGI Global, 2019, p. 224–236.