

DOI: 10.32517/0234-0453-2023-38-5-78-86

ПРЕПОДАВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON В ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ

Н. А. Отаханов¹ ✉¹ Наманганский государственный университет, г. Наманган, Узбекистан

✉ nurillo_otaxanov@list.ru

Аннотация

В современном цифровом обществе расширяется спектр профессий, связанных с задачами обработки большого объема данных. Для быстрого, эффективного и качественного решения таких задач требуются соответствующие программные средства, разработанные профессиональными программистами. Язык программирования Python позволяет быстро, просто и экономично разрабатывать новые программные средства в рамках профессиональных обязанностей сотрудниками, которые не обладают глубокими знаниями в области программирования, но хорошо осведомлены о специфике задач. По этой причине внедрение языка Python в учебный процесс в вузах с учетом особенностей образовательных направлений может стать хорошим инструментом повышения уровня готовности выпускников к профессиональной деятельности.

В статье рассматривается проблема интеграции языка программирования Python в процесс преподавания информатики в вузах. Предлагается определить содержание учебных материалов в зависимости от специфики направления обучения. Рекомендуется организовать процесс изучения языка программирования Python на основе классификации модулей. В исследовании используются методы анализа и синтеза существующих источников по языку Python, обучающих программ по различным образовательным направлениям, стандартных программ для преподавания информатики и информационных технологий. Предложен примерный перечень тем и учебной литературы по языку Python с учетом специфики образовательных направлений бакалавриата.

Выпускники университетов, владеющие языком Python в пределах, рекомендованных в этой статье, в дополнение к профессиональным навыкам также будут обладать компетенциями программирования и будут способны реализовать навыки программирования в рамках своих профессиональных обязанностей.

Ключевые слова: язык программирования Python, образовательное направление, классификация, класс, спецификация обучения, модуль.

Для цитирования:

Отханов Н. А. Преподавание языка программирования Python в высших учебных заведениях. *Информатика и образование*. 2023;38(5):78–86. DOI: 10.32517/0234-0453-2023-38-5-78-86.

TEACHING THE PYTHON PROGRAMMING LANGUAGE IN HIGHER EDUCATION INSTITUTIONS

N. A. Otahanov¹ ✉¹ Namangan State University, Namangan, Uzbekistan

✉ nurillo_otaxanov@list.ru

Abstract

In the modern digital society, the range of professions whose activities are related to the tasks of processing a large amount of data is expanding. For a quick, efficient, and high-quality solution of such tasks, appropriate software tools developed by professional programmers are required. The Python programming language allows fast, simple and economic development of new software tools within the framework of professional duties by employees who do not have deep knowledge in the field of programming but are well aware of the subject specifics of tasks. For this reason, the introduction of Python into the educational process in universities taking into account the peculiarities of educational directions can be a good tool for increasing the level of readiness of graduates for professional activity.

This article discusses the problem of integrating the Python programming language into the informatics course at universities. It is proposed to determine the content of educational materials depending on the specifics of the field of study. It is recommended to organize the process of learning the Python programming language based on the classification of modules. The research uses methods of analysis and synthesis of existing sources on the Python language, training programs in various educational directions, standard programs for teaching informatics and information technology. An approximate list of topics and educational literature on the Python language is proposed, taking into account the specifics of educational directions of bachelor's degree programs.

University graduates with Python language skills within the limits recommended in this article, in addition to professional skills, will also have programming competencies and will be able to implement programming skills within their professional duties.

Keywords: Python programming language, educational direction, classification, class, learning specialty, module.

For citation:

Otahanov N. A. Teaching the Python programming language in higher education institutions. *Informatics and Education*. 2023;38(5):78–86. (In Russian.) DOI: 10.32517/0234-0453-2023-38-5-78-86.

© Отханов Н. А., 2023

1. Введение

Особенности жизни в современном мире таковы, что уметь работать с цифровыми данными должны не только сотрудники ИТ-предприятий, но и рядовые члены цифрового общества. Поскольку объемы информации постоянно растут, проблемы создания, хранения, сортировки, обработки и передачи информации невозможно решить быстрыми, экономичными и качественными методами без знания информатики. Во всех современных вузах студенты практически всех специальностей изучают предмет «Информатика и ИКТ», глубина и объем освоения которого зависят от специфики выбранной ими специальности.

Информатику в зависимости от области применения можно рассматривать как науку, как технологию и как индустрию. Информатика как наука объединяет собой группу дисциплин, занимающихся изучением различных свойств информации в информационных процессах. Информатика как технология — это система процедур компьютерного преобразования информации с целью ее формирования, хранения, обработки, передачи и применения в практической деятельности. Информатика как индустрия обеспечивает все другие отрасли народного хозяйства необходимыми информационно-программными ресурсами и сервисными услугами. Именно в последней сфере заметен острый дефицит. Предметно-ориентированные языки программирования могут стать одним из ключевых средств решения этой проблемы.

2. Python как предметно-ориентированный язык программирования

Предметно-ориентированные языки программирования (ПОЯП) — это языки программирования с высокой степенью модульности. Они являются удобными инструментами для разработки новых программ и упрощают процесс создания программных средств для различных предметных областей. Это позволяет разрабатывать программные средства специалистам, которые не обладают высокими навыками программирования, но глубоко понимают специфические особенности предметно-профессиональных проблем конкретной области. ПОЯП используют специфические термины и правила соответствующей отрасли для сбора, хранения, обработки и передачи данных. Это упрощает процесс разработки программ и работу программистов по многим смежным направлениям. Поэтому такие языки программирования становятся все более популярными в мировой практике программирования.

В ПОЯП термин «*модуль*» интерпретируется как библиотека методов и функций, каждая из которых независима и предназначена для полного решения небольшой конкретной задачи. Каждый модуль может содержать большой набор типичных методов и функций конкретной области, предназначенный для использования в процессе разработки

программных средств. Теоретически количество методов и функций в модуле не ограничено.

Языки программирования, разработанные для конкретной области и обогащенные парадигмами программирования, являются эффективными инструментами для разработки программного обеспечения и используются сегодня для широкого класса задач.

2.1. Особенности использования языка Python

Python является наиболее популярным ПОЯП и представляет собой надежную систему, состоящую из большого набора инструментов и возможностей. Python не так сложен, как языки программирования общего назначения — Java, Pascal или C++. Он предлагает программистам удобный интерфейс, широкий диапазон данных и огромное количество готовых шаблонов для типичных задач. В новых версиях языка расширяются и области применения программных средств. Именно поэтому в 2022 году Python стал самым востребованным языком программирования среди профессиональных и непрофессиональных программистов*.

Благодаря модульности высокого уровня разработка новых приложений на языке программирования Python не требует особых навыков: достаточно обладать необходимыми знаниями по предметной области, для которой создается программное обеспечение, и иметь базовые навыки программирования.

Python, как и другие ПОЯП, в процессе разработки новых программных средств обеспечивает программистам следующие преимущества:

- оптимизирует работу за счет использования готовых шаблонов и методов;
- избавляет от необходимости создания менее эффективных кодов, требующих повторного выполнения одинаковых операций;
- делает наглядным процесс разработки ПО за счет возможности построения алгоритма решения задачи в виде логического проекта;
- обеспечивает эффективную и качественную организацию этапов разработки;
- улучшает качество программного продукта;
- допускает использование общепринятых условий и правил в задачах конкретной предметной области;
- обеспечивает удобство редактирования, отладки, чтения, понимания и анализа программного кода;
- позволяет снизить затраты, связанные с процессом разработки приложений.

2.2. Классификация модулей языка Python

Язык программирования Python включает в себя около 400 модулей: 250 стандартных и 150 внешних. Общее количество классов, методов, функций и констант языка Python составляет примерно 8000. Они

* Сменился самый популярный в мире язык программирования. *CNews*. 15.08.2022. https://www.cnews.ru/news/top/2022-08-15_si_nizvergnutsmenilsya

позволяют программистам разрабатывать новые приложения простыми, быстрыми, высококачественными и экономичными способами. Поскольку каждый модуль Python предназначен для решения некоторых типичных задач в определенной области и использует термины, концепции и законы, относящиеся к этой области, было бы логично рассматривать его как язык программирования, специфический для конкретных предметных областей.

Анализ открытых источников показал, что существует большое количество научных исследований по языку программирования Python [1–13], но ни в одном из них внешние и внутренние модули Python не классифицированы полностью. Основная часть этих исследований посвящены описанию отдельно взятых модулей языка Python или применению языка программирования в конкретной узкой предметной области.

D. Hellmann в книге «The Python 3 standard library by example» [8] классифицировал модули ПОЯП Python: описал около 120 и сгруппировал их

в 17 классов. Мы уточнили и дополнили эту классификацию, для чего проанализировали большой объем существующей литературы, каталог программного обеспечения PyPI* и системы поддержки интерпретатора Python.

В результате мы обнаружили еще 130 внутренних модулей языка и расширили диапазон классов на шесть единиц. Полученная классификация была представлена в статье «Classification of Python language modules» [14].

Классификация модулей показывает, что известные на сегодняшний день внутренние модули языка Python охватывают типичные задачи по 23 областям профессиональной деятельности. Обновленная классификация может быть не самой совершенной, но на данный момент является наиболее полной. Учитывая важность классификации в организации учебного процесса с учетом специфики образовательных направлений, представим ее в этой статье (табл. 1).

* Python Package Index (PyPI). <https://pypi.org>

Таблица 1 / Table 1

Классификация внутренних модулей языка Python

Classification of Python internal modules

№ п/п	Классы	Список модулей
1	Работа с текстами	difflib, grep, re, string, stringprep, textwrap, textview
2	Структуры данных	array, bisect, collections, collection.abc, copy, enum, heapq, pprint, queue, struct, weakref
3	Алгоритмы	contextlib, functools, itertools, operator
4	Время и дата	calendar, datetime, time
5	Математика	cmath, decimal, fractions, math, numeric, random, statistics
6	Файловая система	codecs, code, filecmp, filelist, fnmatch, genericpath, glob, io, linecache, mmap, mimetypes, ntpath, os.path, outwin, pathlib, plistlib, posixpath, quopri, setuptools, shutil, tempfile
7	Постоянное хранение данных и обмен данными	aifc, binascii, binhex, chunk, copyreg, csv, dbm, dbm.dumb, dbm.gnu, dbm.ndbm, Elementtree, grp (Unix), msilib, netrc, pickle, pickletools, query, shelve, sqlite3, spwd (unix), unicodedata, uu, xml.etree, xdrlib
8	Сжатие и архивирование файлов	bz2, gzip, lzma, tarfile, zipapp, zipfile, zipimport, zlib
9	Криптография	crypt, encoding, hashlib, hmac, h1, secrets
10	Параллельные вычисления: процессы, потоки, сопрограммы	asyncio, concurrent.Futures, multiprocessing, signal, threading, ubprocess
11	Обмен данными в сети	asynchat, asyncore, ipaddress, mb1, nntplib, poplib, prc, selectors, select, socket, socketserver, telnetlib
12	Интернет	base64, browser, email, ftplib, html, html.entities, html.parser, h2, http.client, http.cookiejar, http.cookies, http.server, imaplib, json, mailbox, mailcap, nturlpath, smtpd, smtplib, squeezer, ssl, urllib.parse, urllib.request, useurllib.Robotparser, uuid, webbrowser, xml.prscient, xml.prserver, xml.client, xml.server
13	Блоки разработки приложений	argparse, atexit, cmd, configparser, fcntl (Unix), fileinput, getopt, getpass, logging, MacOSx, pipes (unix), pty (Unix), py_compile, pynshell, readline, shlex, shed, sidebar, token, wsgiref
14	Интернационализация и локализация приложений	gettext, locale, zoneinfo

Окончание таблицы 1 /
End of the table 1

№ п/п	Классы	Список модулей
15	Инструменты для разработки программного обеспечения	bdb, builtins, cgitb, compileall, contextvars, configdialog, ctypes, curses, curses.ascii, curses.panel, curses.textpad, cProfile, dataclasses, debugger, delegator, distutils, doctest, dynoption, e1, ensurepip, errno, faulthandler, formatter, gyp, gyp_parser, help_about, h3, help, hotshot, idle, idlelib, idle_test, iomenu, keyword, lib2to3, marshal, ParenMatch, pstats, pydoc, pdb, profile, pycbr, tabnanny, venv, rlcompleter, pydoc, pyexpat, PyParsing, sched, search, sre_parse, sre_compile, sre_constants, stat, symtable, test, timeit, trace, traceback, tty (UNIX), unittest, xxsubtype, zoomheight, zdummy
16	Инструменты среды выполнения программы	cgi, codeop, gc, msvcrt, nt, os, platform, resource, replib, runscript, site, sys, sysconfig, tooltip, tracemalloc, types, typing, undo, winreg
17	Языковые инструменты	abc, dis, inspect, tokenize, warnings
18	Работа с модулями и пакетами	importlib, pip, imp, modulefinder, multical, nis (unix), opcode, pkgutil, pkg_resource, redirector, runpy, syslog, zipimport
19	Работа с изображениями	antigravity, colorsys, colormath, config_key, graphlib, scrolledlist, tkinter, turtle, turtledemo
20	Парсинг параметров командной строки	ast, optparse
21	Работа со звуком	audioop, autocomplete, sndhdr, sunau, wave, winsound
22	Распознавание образов	imghdr

Возможности языка Python расширены дополнительными модулями. Эти модули не являются частью стандартного языка, и поэтому их называют внешними. На сегодняшний день разработано большое количество внешних модулей для различных сфер,

которые широко используются в практике программирования. Более 100 таких модулей обнаружены и классифицированы нами с предметно-ориентированной точки зрения в указанной выше статье. Классификация приведена в таблице 2.

Таблица 2 / Table 2

Классификация внешних модулей языка Python

Classification of external Python modules

№ п/п	Классы	Список модулей
1	Работа с графическими интерфейсами	Pmw, PyGObject, Tix, tree, WCK, wxPython
2	Работа с базами данных	Gadfly, KinterbasDB, MySQLdb, PyGreSQL, SQLAlchemy
3	Работа с веб-страницами	Beautiful Soup, Flask, Google, jabberpy, Libgmail, Maps, Mechanize, pyquery, Requests, Scrape, Selenium, Sentry, Twisted, WhatWaf
4	Изображения и видео	Gdmodule, MoviePy, PIL, pyscreenshot, Video Capture, VPython
5	Данные и математические операции	Chainer, Matplotlib, NumPy, Pandas, SciPy, Sympy, Teano
6	Разработка игр	Pygame, Pyglet, pyOpenGL
7	Работа со звуком	Mutagen, PMIDI, pySonic, pyMedia
8	Для Mac OS	py2app, PyObjC
9	Для USB и последовательных портов	PyUSB, PySerial, USPP
10	Нейронные сети	Detectron, Keras, Nilearn, PyTorch, Pylearn2, sikit-learn, Som-tsp
11	Для процесса разработки программного обеспечения	Cookiecutter, Multidiff, Pipenv, pyExpect, Python-fire
12	Криптовалюта	SimpleCoin
13	Работа с файлами	Gensim

Следует отметить, что требуется много физических усилий и времени, чтобы собрать, проанализировать и подробно описать, какие типичные задачи включены в каждый из модулей языка программирования Python. Можно сказать, что методы и функции каждого модуля Python требуют отдельного исследования. Например, модуль *itertools* класса «алгоритмы» включает в себя более 20 методов, каждый из которых предназначен для решения одной из типичных задач итерационных процессов и может быть использован в процессе разработки программных кодов для обработки данных в различных сферах человеческой деятельности.

Знание модулей языка Python и умение их эффективно применять повышают уровень готовности выпускников к профессиональной деятельности. **Основная цель статьи** — показать способы создания курса, посвященного языку программирования Python, выбора подходящих модулей для конкретных образовательных направлений и организации учебного процесса.

3. Примерное содержание курса изучения языка Python: темы и модули

Обучение языкам программирования в высших учебных заведениях формирует у студентов следующие компетенции:

- теоретические знания о программировании, его положении в мире и месте в системе наук;
- знание основ предметно-ориентированных и других языков программирования;
- умение интерпретировать с точки зрения программирования информацию различных форм и разного содержания;
- умение классифицировать прикладные задачи по содержанию, масштабу или практической значимости;
- умение находить эффективные методы разработки программных кодов;
- навыки выбора и целенаправленного применения парадигм и средств программирования;
- умение использовать основные механизмы обработки графической, текстовой, табличной и аудиовизуальной информации с использованием комплекса современных прикладных программ;
- навыки разработки и внедрения новых программных продуктов для обработки данных различной формы и содержания.

Указанный комплекс компетенций формируется и развивается еще на уровне бакалавриата независимо от специализации, но только при условии, что процесс обучения студентов языкам программирования организован с учетом особенностей выбранной предметной области. Следует отметить, что правильное определение содержания курса является одной из важнейших задач при обучении языкам программирования.

Образовательный процесс должен быть организован по принципу «от простого к сложному» и «от

известного к неизвестному». Если при такой организации обучения студентов в вузах преподаватели будут опираться на структуры языка Python, известные из школьного курса информатики, то эффективность учебного процесса значительно возрастет.

Для подготовки специалистов высокого уровня, обладающих вышеуказанными компетенциями, необходимо проанализировать, пересмотреть, переопределить содержание образования и внести соответствующие коррективы в процесс преподавания языка программирования Python в вузах.

Созданная классификация модулей позволяет преподавать популярнейший язык программирования Python в зависимости от специфических особенностей и общих квалификационных требований к уровню подготовки выпускников вузов. **В процессе обучения языку Python важно определить:**

- типы модулей для конкретного образовательного направления или предметной отрасли;
- методы, входящие в каждый модуль;
- назначение и практическое применение методов.

На первый вопрос позволяет ответить классификация модулей, а на два следующих — непосредственно процесс обучения языку программирования Python. Таблицы 1 и 2 показывают, что с помощью языка программирования Python можно ускорить разработку программ для решения большого количества специфических задач различных форм и содержания, часто встречающихся в профессиональной деятельности.

3.1. Темы для изучения языка Python

Обычно темы, рекомендованные для преподавания языков программирования, указываются в типовых программах, разработанных на основе квалификационных требований соответствующих специальностей. На наш взгляд, было бы целесообразно включить базовые структуры и модули языка программирования Python в типовую программу соответствующих предметов (например, «Применение цифровых технологий», «Основы программирования и алгоритмизации» и т. д.) по всем образовательным направлениям, которые связаны с обработкой большого объема данных. Основываясь на результатах исследований и рекомендациях ведущих ученых в области преподавания языка программирования Python* [1–13, 15, 16], а также достижениях современной мировой практики программирования, предлагаем примерный перечень общих тем, рекомендуемых для преподавания ПОЯП (табл. 3).

Мы предлагаем сделать темы 1–7 обязательными для всех образовательных направлений. Содержание учебных материалов для теоретических, практических и самостоятельных занятий по этим темам должно соответствовать специфике направ-

* Sharma R. Top 5 Python modules you should know in 2023. *upGrad.com*. 06.11.2022. <https://www.upgrad.com/blog/python-modules-you-should-know/>

Рекомендуемые темы для преподавания языка Python**Recommended themes for teaching Python**

№ п/п	Темы
1	Алфавит языка программирования Python. Запись переменных и констант. Правила записи арифметических операций. Типы данных, преобразование данных одного типа в другой. Команды ввода и вывода данных. Линейные коды. Библиотека math
2	Способы кодирования процессов разветвления и повторения
3	Работа со списками
4	Организация работы со словарями и коллекциями
5	Создание, запись, чтение и обработка файлов
6	Объявление и использование новых функций. Лямбда-функция. Декораторы. Работа с модулями. Создание и применение модулей
7	Создание и использование классов и объектов
8	Работа с модулем Matplotlib
9	Основы работы с модулями Keras и PyTorch
10	Методы и классы модуля SciPy
11	Библиотека itertools
12	Модули time, datetime и calendar

лений бакалавриата (т. е. выбранной специальности) и учитывать типичные задачи, часто встречающиеся в практической деятельности будущих специалистов. В противном случае предмет будет непонятен и скучен для студентов, и такая ситуация приведет к постепенному угасанию интереса у обучающихся и потере стимулов к изучению языков программирования.

На практических и самостоятельных занятиях должна быть использована таблица значений. Создание таблицы значений для несложных задач вручную поможет визуализировать процесс выполнения разрабатываемых кодов. Преподаватель может потребовать от студентов включить таблицу значений в отчет о проделанной работе. В результате уменьшается количество ошибок в тексте кодов, упрощается процесс редактирования, улучшается наглядность разработки программных средств. Это, в свою очередь, является одним из важных и положительных факторов формирования и развития профессиональных навыков будущих программистов.

3.2. Модули для изучения языка Python

Анализ существующей учебной литературы показывает, что модули, представленные в темах 8–12, могут быть признаны обязательными для всех специальностей. Эти темы помогут студентам научиться составлять программные коды для большого количества задач.

Модуль Matplotlib — это библиотека методов и функций, предназначенных для визуализации данных в процессе работы с двумерными (2D) и трехмер-

ными (3D) изображениями. Графические элементы, созданные с помощью средств этого модуля, могут быть использованы в качестве иллюстраций.

Модули Keras и PyTorch включают в себя множество инструментов и методов, которые могут быть широко использованы в работе с искусственным интеллектом, в компьютерном обучении на основе нейронных сетей и при разработке умных технологий. Изучение этих модулей полезно, поскольку умение работать с искусственным интеллектом становится одним из квалификационных требований специалиста широкого ряда направлений.

Модуль itertools состоит из набора методов, которые позволяют организовать процесс повторения операций в заданном диапазоне для каждого элемента заданной последовательности. Обычно программисты сталкиваются с подобными операциями в процессе обработки данных больших объемов.

Модули time, datetime и calendar состоят из библиотеки классов и методов, предназначенных для работы с данными времени и даты. Эти модули мы рекомендуем как обязательные, поскольку все специалисты по обработке информации должны уметь с ними работать.

Модуль SciPy разработан на основе библиотеки NumPy и поможет легко выполнять широко используемые научные вычисления по математике, численным методам, теории вероятности и т. д.

В дополнение к вышеуказанным темам в учебном плане должны быть предусмотрены занятия для изучения новых модулей Python. Отбор модулей для

Список рекомендуемых внутренних и внешних модулей для некоторых курсов бакалавриата и смежных дисциплин**List of recommended internal and external modules for some undergraduate courses and related disciplines**

№ п/п	Образовательные направления	Рекомендуемые модули для преподавания	Учебная литература
1	Экономика, финансы	NumPy, SciPy, Pandas, matplotlib, itertools, SimpleCoin, statistics, decimal, Sqlite3, MySQLdb, zipfile, locale, zoneinfo, PyTables, Chainer	[2, 11]
2	Геология, геодезия, география, геоинформационные системы	Gdal/OGR, Seaborn, Shapely, Mapnik, geoDjango, Matplotlib, Dolerean, NumPy, pyproj, PyGreSQL, locale, SciPy, zoneinfo, Folium, GmPlot, BaseMap	[15]
3	Психология	array, NumPy, random, matplotlib, psychopy, os, os.path	[3]
4	Журналистика, лингвистика	abc, string, PyQt5, PyTorch, NLTK, Scikitlearn, textblob, multiprocessing, Tweepy, Bokeh, Flask, matplotlib, Keras, tensorflow, textwrap, textview	[6, 7]
5	Биология, биоинформатика	Pymol, PyTorch, Keras, NumPy, SumPy, pandas, decimal, fractions, random, statistics, pylj, iPython, Matplotlib	[1, 16]
6	Химия, медицина	Keras, NumPy, SymPy, pandas, decimal, fractions, random, array, statistics, ChemPy, pylj, Chemistry, iPython, Matplotlib	[9, 16]
7	Физика	Mathplotlib, NumPy, array, decimal, itertools, kahansum, chargearray, legendry, psis, triang, jacobi, ludec, grdec, bisection, scipy, tkinter, vpython, sympy	[5, 13]
8	Математика, прикладная математика, информатика, инженерия (вместе со смежными направлениями)	binarytree, NumPy, SymPy, SciPy, array, Pandas, matplotlib, decimal, operator, itertools, dolerean, MySQLdb, fractions, SQLAlchemy, cmath, random, zipfile, Flask, Selenium, Sqlite3, Django, Keras, chainer, statistics, teano, MoviePy, VPython, Plotly, PyQt5, PyGame, os.path, shutil and other	[4, 8, 10, 12]

преподавания осуществляется с учетом специфики образовательных направлений. Для этого рекомендуется использовать таблицу 4.

Научить студентов всему потенциалу Python — практически невыполнимая задача. Поэтому считаем целесообразным организовать учебный процесс по изучению ПОЯП с учетом типичных задач, соответствующих специфике будущей профессиональной деятельности студентов, обучающихся по конкретным образовательным направлениям. Иначе говоря, учебные материалы, предлагаемые на занятиях, должны опираться на профессиональные задачи, с которыми студенты могут столкнуться в своей будущей деятельности.

Поскольку практически *всем специалистам, чья деятельность связана с обработкой информации, понадобятся умения и навыки по работе с математической и строковой информацией, мы считаем необходимым обучение методам и классам математических и строковых модулей*. Эти модули можно выбрать по мере необходимости из таблиц 1 и 2. Кроме того, в таблице 4 перечислены некоторые рекомендуемые для преподавания модули с учетом специфики направлений бакалавриатов.

В рамках своих профессиональных обязанностей инженеры сталкиваются со многими проблемами,

связанными с выполнением расчетов, созданием чертежей и диаграмм. Модули языка программирования Python, рекомендуемые для математиков (в том числе для работы с большими данными, базами данных, веб-приложениями, файлами и т. д.), могут быть полезными для подготовки будущих инженеров к профессиональной деятельности. Поэтому им могут быть рекомендованы те же модули и литература, что и математикам.

4. Планируемые результаты

В современном цифровом обществе увеличивается объем информации, поэтому проблемы ее сбора, сохранения, сортировки, обработки и передачи данных должны решаться своевременно, быстро, качественно и экономно. Известно, что успех в этой сфере зависит от наличия соответствующего программного обеспечения и выполнения условий его использования.

Для решения большого круга задач уже существует и активно применяется на практике соответствующее программное обеспечение. Но в различных профессиональных и других сферах человеческой деятельности остается огромное количество широко распространенных задач, которые ожидают своей очереди в цифровизации. Обычно создание

новых или модернизация существующих программ профессионалами-программистами обходится дорого. Выпускники вузов, знакомые с типичными проблемами в сфере своих профессиональных обязанностей и обладающие умениями и навыками программирования, станут ключевыми специалистами в области решения вышеуказанных проблем. Поэтому преподавание языка программирования Python с учетом профессиональных особенностей бакалавров будет способствовать успешному выполнению квалификационных требований и повышению профессионального уровня выпускников. Студенты различных образовательных направлений (математика, теория вероятностей, статистика, прикладная математика, информатика, программная инженерия, программное обеспечение ЭВМ, цифровая экономика, большие данные, облачные технологии, искусственный интеллект, информационные технологии, сетевые технологии и т. д.), обладающие компетенциями программирования на языке Python, закончив вуз, будут вести активную деятельность в условиях цифровизации общества и экономики.

Преподавание ПОЯП в вузах позволяет подготовить:

- активных членов цифрового общества, умеющих работать с информацией;
- выпускников, готовых к самостоятельной исследовательской жизни в цифровом обществе;
- специалистов, способных создавать программные средства в рамках профессиональных обязанностей;
- сотрудников, умеющих решать проблемы, связанные с обработкой информации оптимальными, быстрыми, экономными и качественными способами.

Кроме этого, изучение языка Python будет способствовать формированию у будущих специалистов следующих профессионально-личностных качеств:

- владение всеми формами и механизмами получения и применения новых знаний;
- способность к удовлетворению современных информационных и культурно-духовных потребностей;
- владение навыками решения задач, встречающихся в профессиональных и других формах деятельности, с помощью цифровых технологий;
- широта интересов, как профессиональных, так и личностных;
- положительное отношение к профессиональным обязанностям;
- устойчивый интерес к исследовательской деятельности и поиску оптимальных способов решения проблем.

5. Заключение

В любых организациях высоко ценятся сотрудники, обладающие навыками и умениями программирования для выполнения типичных задач по своим

профессиональным обязанностям. На сегодняшний день Python является наиболее подходящим языком программирования для подготовки таких специалистов. Поэтому требуется внедрить изучение языка Python в содержание предмета «Информатики и ИКТ».

С появлением новых версий языка состав модулей модернизируется (удаляются устаревшие и добавляются новые), что расширяет возможности ПОЯП Python за счет упрощения и совершенствования процедуры разработки программного обеспечения и увеличивает число его активных пользователей. Эти изменения, безусловно, должны учитываться при подготовке специалистов, чья деятельность связана с обработкой больших объемов информации.

Как мы отметили выше, наиболее важным преимуществом Python является то, что он снижает затраты (память, время, деньги, человеческие и другие ресурсы), связанные с разработкой программного обеспечения. Кроме того, он позволяет программистам значительно снизить сложность задач, а значит, даже те, кто обладает поверхностными знаниями и навыками в программировании на Python, могут легко разрабатывать новые программные приложения или модернизировать существующие.

На занятиях по курсу «Информатика и ИКТ» имеет смысл преподавать только те модули языка Python, которые будут необходимы в дальнейшей профессиональной деятельности студентов, т. е. учитывать специфику образовательных направлений.

Сотрудник, умеющий пользоваться широким спектром программного обеспечения, способный вносить соответствующие изменения в программные коды или разрабатывать новые программные продукты по мере необходимости для своей профессиональной деятельности, ценится выше.

Благодарности

Я хотел бы поблагодарить моего покойного отца Абдумалика Хаджи, маму Лутфияхон и жену Асилахон за всевозможную поддержку, а также выразить особую благодарность и признательность моему научному руководителю, доктору физико-математических наук, профессору Мирсаиду Арипову за ценные замечания и идеи при проведении исследования и подготовке данной статьи.

Acknowledgments

I would like to thank my deceased father Abdumalik Haji, mother Lutfiyahon and wife Asilahon for all kinds of support and also express special thanks and appreciation to my scientific supervisor, Doctor of Sciences (Physics and Mathematics), Professor Mirsaid Aripov for valuable comments and ideas in conducting the research and preparing this article.

Список источников / References

1. Bassi S. Python for bioinformatics. 2nd ed. New York, Chapman and Hall/CRC Press; 2017. 452 p. DOI: 10.1201/9781315268743.
2. Хрипунова М. В., Губернаторов А. М. Экономика на Python. М.: Прометей; 2021. 316 с.
[Khripunova M. V., Gubernatorov A. M. Economics in Python. Moscow, Prometey; 2021. 316 p. (In Russian.)]
3. Dalmaijer E. S. Python for experimental psychologists. Abingdon, Oxon; New York, NY; Routledge; 2017. 228 p.

4. *Dangeti P. et al.* Numerical computing with Python. Birmingham, UK, Packt Publishing Ltd; 2018. 682 p.
5. *Gezerlis A.* Numerical methods in physics with Python. London, Cambridge University Press; 2020. 504 p.
6. *Горожанов А. И.* PyQt5 для лингвистов. Электронное учебное пособие, версия в формате PDF. 2014. 202 с. [*Goroganov A. I.* PyQt5 for linguistics. Electronic training manual, PDF version. 2014. 202 p. (In Russian.)]
7. *Hammond M.* Python for Linguists. Cambridge, UK, Cambridge University Press; 2020. 310 p.
8. *Hellmann D.* The Python 3 standard library by example. Boston, Addison Westley, USA, 2017; 1464 p.
9. *Heys J. J.* Chemical and biomedical engineering calculations using Python. USA, John Wiley & Sons, 2016; 263 p.
10. *Hill K.* Learning scientific programming with Python. London, Cambridge Press; 2020. 647 p.
11. *Hilpisch I.* Python for finance. Mastering data-driven finance. Sebastopol, CA, USA, O'Reilly Media Inc., 2016; 693 p.
12. *Криволапов С. Я., Хрипунова М. Б.* Математика на Python. Учебник. М.: Кнорус; 2021. 456 с. [*Krivolapov S. Ya., Khripunova M. B.* Mathematics in Python. Tutorial. Moscow, Knorus; 2021. 456 p. (In Russian.)]
13. *Moruzzi G.* Essential Python for the physicist. Springer Nature Switzerland, 2021; 312 p.
14. *Otahanov N. A.* Classification of Python language modules. *Indonesian Journal of Innovation Studies*. 2022;19:1–12. DOI: 10.21070/ijins.v19i.672.
15. *Вестра Э.* Разработка геоприложений на языке Python. М.: ДМК Пресс; 446 с. [*Westra E.* Python geospatial development. Moscow, ДМК Press; 446 p. (In Russian.)]
16. *Yones-Clark K.* Mastering Python for bioinformatics. Sebastopol, CA, USA, O'Reilly Media Inc., 2021; 454 p.

Информация об авторе

Отаханов Нурилло Абдумаликович, канд. пед. наук, доцент, профессор кафедры информатики, факультет математики, Наманганский государственный университет, г. Наманган, Узбекистан; *ORCID*: <https://orcid.org/0000-0002-3155-9347>; *e-mail*: nurillo_otahanov@list.ru

Information about the author

Nurillo A. Otahanov, Candidate of Sciences (Education), Docent, Professor at the Department of Informatics, Faculty of Mathematics, Namangan State University, Namangan, Uzbekistan; *ORCID*: <https://orcid.org/0000-0002-3155-9347>; *e-mail*: nurillo_otahanov@list.ru

Поступила в редакцию / Received: 25.04.23.

Поступила после рецензирования / Revised: 03.07.23.

Принята к печати / Accepted: 04.07.23.