



Полтавський національний технічний  
університет імені Юрія Кондратюка

# СИСТЕМИ УПРАВЛІННЯ, НАВІГАЦІЇ ТА ЗВ'ЯЗКУ

4(50) ' 2018

Заснований  
у 2007 році

Наукове періодичне видання,  
в якому відображені результати  
наукових досліджень з розробки та  
удосконалення систем управління,  
навігації та зв'язку у різних  
проблемних галузях.

**Засновник:**

Полтавський національний технічний  
університет імені Юрія Кондратюка

**Адреса редакційної колегії:**

Україна, 36011, м. Полтава,  
Першотравневий проспект, 24

**Телефон:** +38 (066) 706-18-30  
(консультації, прийом статей).

**E-mail:**

kozelnkova@ukr.net

**Інформаційний сайт:**

<http://www.pntu.edu.ua>

**Реферативна інформація**

зберігається: у загальнодержавній  
реферативній базі даних  
„Україніка наукова” та публікується  
у відповідних тематичних серіях  
УРЖ „Джерело”.

**РЕДАКЦІЙНА КОЛЕГІЯ:**

**Голова:**

КОЗЕЛКОВ Сергій Вікторович (д-р техн. наук, проф., Україна)

**Заступники голови:**

ШЕФЕР Олександр Віталійович (канд. техн. наук, доц., Україна)

ШУЛЬГА Олександр Васильович (д-р техн. наук, доц., Україна)

**Члени:**

БЛАУНШТЕЙН Натан Олександрович (д-р техн. наук, проф., Ізраїль)

ВЕСОЛОВСЬКИЙ Кшиштоф (д-р техн. наук, проф., Польща)

ІЛЬІН Олег Юрійович (д-р техн. наук, проф., Україна)

КОРОБКО Богдан Олегович (д-р техн. наук, доц., Україна)

КОШОВИЙ Микола Дмитрович (д-р техн. наук, проф., Україна)

КРАСНОБАЄВ Віктор Анатолійович (д-р техн. наук, проф., Україна)

КУЧУК Георгій Анатолійович (д-р техн. наук, проф., Україна)

ЛАДАНЮК Анатолій Петрович (д-р техн. наук, проф., Україна)

ЛУНТОВСЬКИЙ Андрій Олегович (д-р техн. наук, проф., Німеччина)

МАШКОВ Віктор Альбертович (д-р техн. наук, проф. Чехія)

МАШКОВ Олег Альбертович (д-р техн. наук, проф., Україна)

МОРГУН Олександр Андрійович (д-р техн. наук, проф., Україна)

ПОПОВ Валентин Іванович (д-р фіз.мат. наук, проф., Латвія)

СТАНКУНАС Йонас (д-р техн. наук, проф., Литва)

СТАСЄВ Юрій Володимирович (д-р техн. наук, проф., Україна)

ФРОЛОВ Євгеній Андрійович (д-р техн. наук, проф., Україна)

ХОРОШКО Володимир Олександрович (д-р техн. наук, проф., Україна)

ЧОРНИЙ Олександр Петрович (д-р техн. наук, проф., Україна)

ШЛОМЧАК Георгій Григорович (д-р техн. наук, проф., Україна)

**Відповідальний секретар:**

КОЗЕЛКОВА Катерина Сергіївна (д-р техн. наук, проф., Україна)

*За достовірність викладених фактів, цитат та інших відомостей відповідальність несе автор*

Журнал індексується міжнародними бібліометричними та наукометричними базами даних: Index Copernicus (Польща), General Impact Factor (ЄС), Google Scholar (США), Academic Resource Index (ЄС), Scientific Indexed Service (США).

*Затверджений до друку вченою радою Полтавського національного технічного університету імені Юрія Кондратюка (протокол № 1 від 12 вересня 2018 року)*

*Занесений до "Переліку наукових фахових видань України, в яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора і кандидата наук" (технічні науки) строком на п'ять років (від дати включення наказом Міністерства освіти і науки України) від 24.10.2017 № 1413 (додаток 7, п. 31)*

*Свідоцтво про державну реєстрацію КВ № 19512-93/2ПР від 16.11.2012 р.*

## З М І С Т

### НАВИГАЦІЯ ТА ДИСТАНЦІЙНЕ ЗОНДУВАННЯ ЗЕМЛІ

<i>Барсов В. И., Плахотный А. В.</i> Определение расстояния до объекта и его геометрических параметров для навигации робота .....	3
<i>Ивашко А. В., Зуев А. А.</i> Разработка методов удаления дефектов на изображениях энергообъектов, полученных в процессе дистанционного мониторинга с помощью БПЛА .....	8

### КОНТРОЛЬ КОСМІЧНОГО ТА ПОВІТРЯНОГО ПРОСТОРУ

<i>Гришманов Д. С., Несмиян О. Ю., Барилюк Ф. А., Толкаченко С. А.</i> Аналіз форм представлення інформації про моделі діяльності чергової зміни районного диспетчерського центру .....	13
<i>Павленко М. А., Шило С. Г., Дмитрієв О. М.</i> Модель функціональної діяльності оператора автоматизованої системи управління повітряним рухом .....	17
<i>Ситник Ю. Б., Стадник В. В.</i> Формування інформаційного поля в кабіні бойового літака в умовах застосування потужних вибухів .....	22

### ПИТАННЯ УПРАВЛІННЯ В СКЛАДНИХ СИСТЕМАХ

<i>Boriak B.</i> Relationships between filter-extrapolator parameters and quality of filtration and forecast .....	27
<i>Волосюк В. К., Жила С. С., Помазан А. И.</i> Синтез когерентных изображений поверхностей пространственно-распределенными многоканальными системами в зоне Фраунгофера .....	33
<i>Запаловський М. Й., Скородєлов В. В., Мезенцев М. В.</i> Синтез управління для оптимізації динамічних процесів електроприводу змінного струму .....	38
<i>Кириченко Ю. В., Карлов В. Д., Кириченко И. Ю., Артеменко А. Н.</i> Плазменная антенна на основе стратифицированного плазменного столба .....	42

### ОПТИМІЗАЦІЯ ТРАНСПОРТНИХ СИСТЕМ

<i>Горбачов П. Ф., Макарічев О. В., Немна Т. В., Свічинський С. В.</i> Експериментальне дослідження прибутковості міжнародних автомобільних перевезень вантажів за разовими заявками .....	50
--	----

### МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ

<i>Ахизер Е. Б., Дунаевская О. И., Сердюк И. В., Стрельникова А. Ю., Гармаш Д. В.</i> Методы решения многоиндексных транспортных задач высокой размерности .....	57
<i>Гадецька С. В., Гороховатський В. О.</i> Застосування статистичних мір релевантності для векторних структурних описів об'єктів у задачі класифікації зображень .....	62
<i>Дєдов О. П.</i> Визначення впливу середовища на динаміку машини на основі спектрального аналізу .....	69
<i>Дьяченко О. С.</i> Теоретичні дослідження просторових коливань віброустановки з пневматичними вібробудувачами коливань .....	73
<i>Самохвалов Ю. Я., Бурба О. И.</i> Оценка эффективности научных и научно-технических проектов на основе обобщенной функции Харрингтона .....	77
<i>Чала О. В.</i> Метод ієрархічного виведення в базі знань інформаційно-управляючої системи в парадигмі «Enterprise 2.0» ....	86
<i>Шуклін Г. В., Барабаши О. В.</i> Математичне моделювання керування процесами інформаційної безпеки в системі державного регулювання кібернетичною безпекою фондового ринку .....	91

### ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

<i>Alyoshin S., Borodina E., Kikot A., Zabran I.</i> Python programming features and new possibilities .....	95
<i>Бульба С. С., Давидов В. В., Кучук Г. А.</i> Метод розподілу ресурсів між композитними застосунками .....	99
<i>Ichanska N., Gritsenko A., Shefer V.</i> Development of software service for sale of vehicles .....	105
<i>Кальченко В. В.</i> Огляд методів проведення тестування на проникнення для оцінки захищеності комп'ютерних систем .....	109
<i>Коваленко О. В.</i> Розробка інформаційної технології передтестової компіляції та розподілу доступу .....	115
<i>Мелешко С. В.</i> Проблеми сучасних рекомендаційних систем та методи їх рішення .....	120
<i>Мерлак В. Ю., Зиков І. С., Молчанов Г. І.</i> Ситуаційно-орієнтований підхід при проектуванні віджетів .....	125
<i>Москаленко А. О., Сокол Г. В., Рвачова Н. В., Глуховець Ю. В., Варич В. В., Бурак Т. В.</i> Інформаційно-довідкова система сервісного центру з ремонту обчислювальної техніки засобами C++/QT .....	129
<i>Нгуєн А.В., Сидоров Я. Е.</i> Рекуррентная нейронная сеть для обработки больших текстовых данных .....	135
<i>Певнев В. Я., Фролов А. В., Фролов В. В.</i> Результаты исследований генераторов псевдослучайных последовательностей с динамическими параметрами .....	139
<i>Рысований А. Н.</i> Метод генерирования нелинейной псевдослучайной последовательности без использования обратных связей .....	144
<i>Улічев О. С.</i> Дослідження моделей розповсюдження інформації та інформаційних впливів в соціальних мережах .....	147
<i>Хох В. Д.</i> Розробка скриптової мови запису правил в експертній системі на основі нечіткої логіки .....	152

### ЗАПОБІГАННЯ ТА ЛІКВІДАЦІЯ НАДЗВИЧАЙНИХ СИТУАЦІЙ

<i>Глива В. А., Левченко Л. О., Тихенко О. М.</i> Методи визначення концентрацій аероіонів у приміщеннях та моделювання їх змін .....	157
<i>Макогон О. А., Акінішин О. Г., Щокін В. М., Пономаренко А. П., Кумпан О. О., Шпінда Є. М.</i> Застосування методів перевірки статичних гіпотез для підвищення ефективності засобів пожежогасіння зразка ОБТ .....	161
<i>Тютюник В. В., Калугін В. Д., Писклакова О. О.</i> Основоположні принципи створення у єдиній державній системі цивільного захисту інформаційно-аналітичної підсистеми управління процесами попередження й локалізації наслідків надзвичайних ситуацій .....	168

<b>АЛФАВІТНИЙ ПОКАЖЧИК</b> .....	178
----------------------------------	-----

S. Alyoshin, E. Borodina, A. Kikot, I. Zaban

Poltava National Technical Yuriy Kondratyuk University, Poltava, Ukraine

## PYTHON PROGRAMMING FEATURES AND NEW POSSIBILITIES

**Purpose.** The purpose of the article is to demonstrate the profitability of new versions of the Python programming language, and also to demonstrate the volumes of this language together with new software. The content of the article is an overview that allows you to understand the features and new features of the Python language. **Results.** It was found that to create programs for different purposes, use a powerful tool Python. **Originality.** Programs written in Python work in exactly the same way, regardless of which operating system they are running on, and also Python adds features that make it widely used. **Practical value.** The cost-effectiveness of new versions of Python and the volumes of this language together with the new provision provides a detailed analysis of the areas where Python is used. The conclusion is that Python has some noteworthy features that make it widely used. **Conclusions.** The arguments we have presented prove that the program written in Python will function exactly the same regardless of which operating system it is running in. Differences arise only in rare cases, and they are easy to anticipate due to the availability of detailed documentation.

**Keywords:** Python, professional language, features, possibilities of Python.

### Introduction

YouTube, DropBox, Google, Quora, Instagram, BitTorrent, Spotify, Reddit, Yahoo Maps, Hipmunk. The thing that unites them is Python. [1] “General purposeness” of the language resulted into extraordinary large field of its application.

**A relevance of this article** is that for the time being Python is the fifth most used as a professional language in Ukraine and the eighth in the list of languages a person wants to learn next according to Dou [2].

**A new element** is that there is no comprehensive and complex article that would cover all below-mentioned questions from different views and both for prepared and non-prepared readers.

**The purpose of article** is to demonstrate profitability of new versions of Python and show the scope of this language along with new soft.

*The tasks are:*

1. To consider new versions of Python and their features.
2. To describe a scope of Python and new soft for its usage.

### Main part

Let us start with first item and examine advantages and specificities of 3.5 version of Python that was released in 2008. Despite the mentioned, most people still prefer to use v.2.7 [3].

Below is the list of minor and not so minor features supposed to change their preferences [3]:

#### 1. *Advanced unpacking*

You can already do this:

```
>>> a, b = range(2)
>>> a
0
>>> b
1
```

Now you can do this:

```
>>> a, b, *rest = range(10)
>>> a
0
>>> b
1
>>> rest
[2, 3, 4, 5, 6, 7, 8, 9]
```

#### 2. *Keyword only arguments*

```
def f(a, b, *args, option=True):
```

...

- option comes after \*args.
- The only way to access it is to explicitly call `f(a, b, option=True)`
- You can write just a \* if you do not want to collect \*args.

```
def f(a, b, *, option=True):
```

#### 3. *Chained exceptions*

Python 3 shows you the whole chain of exceptions.

#### 4. *Everything is an iterator*

In Python 3, `range`, `zip`, `map`, `dict.values`, etc. all return memory-efficient iterables. If you want a list, just wrap the result with `list`.

This way it is harder to write code that accidentally uses too much memory, because the input was bigger than you expected.

#### 5. *No more comparison of everything to everything*

In Python 2, you can do:

```
>>> max(['one', 2]) # One “is” the loneliest
number
```

```
'one'
```

It is because in Python 2, you can < compare anything to anything. In Python 3, you cannot do this:

```
>>> 'one' > 2
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: unorderable types: str() > int()
```

6. *Yield from*

Instead of accumulating a list, just yield or yield from.

- 1) Bad
 

```
def dup(n):
    A = []
    for i in range(n):
        A.extend([i, i])
    return A
```
- 2) Good
 

```
def dup(n):
    for i in range(n):
        yield i
        yield i
```
- 3) Better
 

```
def dup(n):
    for i in range(n):
        yield from [i, i]
```

7. *Asyncio*

Uses new coroutine features and saved state of generators to do asynchronous IO.

8. *FaultHandler*

Shows traceback even if Python has crashed in a bad way.

9. *Unicode variable names*

```
>>> résumé = "knows Python"
>>> π = math.pi
```

10. *Pathlib*

In Python 2, path handling is verbose.

```
import os
directory = "/etc"
filepath = os.path.join(directory, "test_file.txt")
if os.path.exists(filepath):
    stuff
```

In Python 3, it is much simpler.

```
from pathlib import Path
directory = Path("/etc")
filepath = directory / "test_file.txt"
if filepath.exists():
    stuff
```

In the second part of this article, we will take a closer look at both evident and little-known or specific fields of Python application.

Testing automation. As a programmer, you need to write tests all the time and Python code fits all requirements for test code: it is plain, simple and needs not so much time to be written. In addition, it has a lot of tooling which comes in handy as well.

Web sites. One can use Python to create dynamic web sites, a program that uses CGI (still it is not recommended), a test for such a program, let alone usage of Django, Flask, Tornado and other frameworks.

Mixing of Python and HTML is possible and provides with libraries, templates and template engines (Mako, Genshi, Jinja) that help a developer to maintain code (fig. 1) [5].

```
>>> template = "<html><body><h1>Hello $s!</h1></body></html>"
>>> print template % "Reader"
<html><body><h1>Hello Reader!</h1></body></html>

>>> from string import Template
>>> template = Template("<html><body><h1>Hello $(name)</h1></body></html>")
>>> print template.substitute(dict(name='Dinsdale'))
<html><body><h1>Hello Dinsdale!</h1></body></html>
```

Fig. 1. Mixed Python and HTML code template

As a Data analyst, you would use Python for getting data from databases, from the web, text files etc., for analysis itself and especially visualization for Python is accompanied by a vast number of great visualization libraries and tools. Here are some examples (fig. 2, 3).

The most common tools for scientific and numeric computing are SciPy (collection of packages for mathematics, science, and engineering), Pandas (data analyzing and modeling library), iPython (interactive shell supporting visualizations and parallel computing) [7].

Another chance to apply this language is teaching or learning programming at any level up to advanced. There is a lot of books and online courses, both free and paid services.

Now about future scope of Python. For a Linux admin, Python is crucial, nowadays almost 1/5 of Linux

job require Python code to be hired [6]. Python programming skills come in handy for Database developers and boost their productivity.

In view of the above and given fact of Python is used in such an amount of fields, the fact that this language in addition has its own GUIs sounds unbelievably.

However, here they are:

- Widgets
- Kivy, for writing multitouch applications.
- Qt via pyqt or pyside [7]

Specialists say that Python has a bright future because it has been voted as most favorite language beating C, C++ and Java; it is robust, comprehensive and versatile.

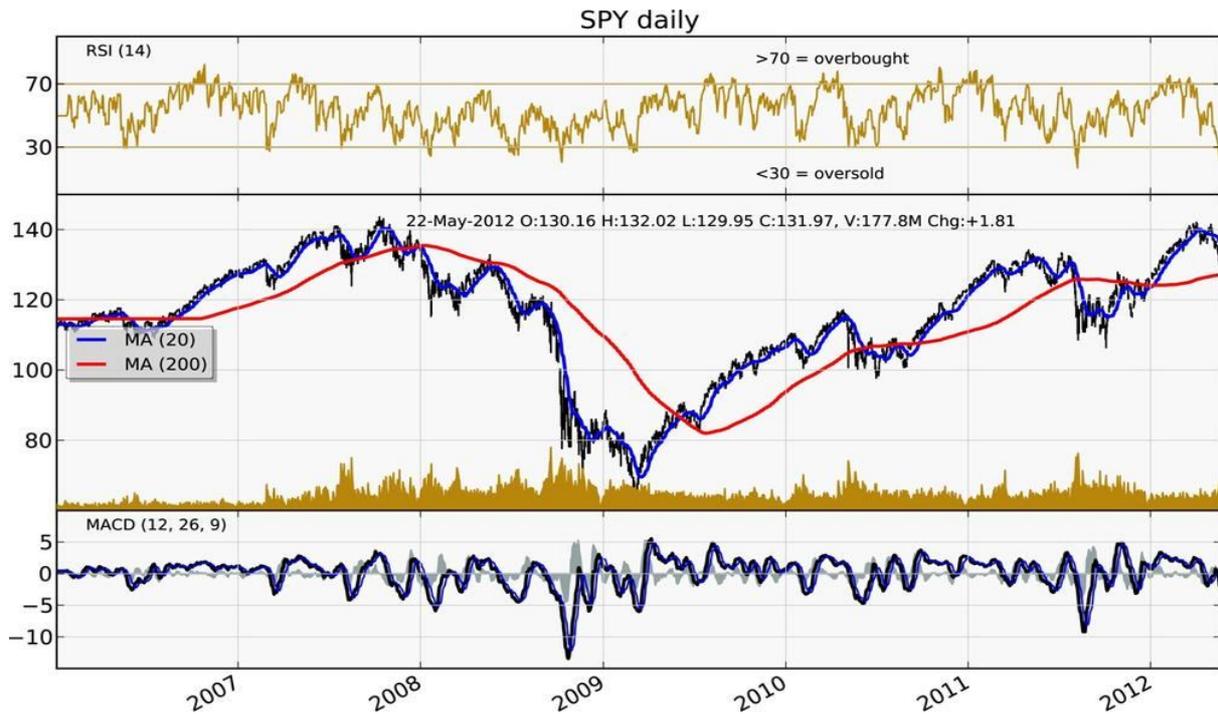


Fig. 2. Matplotlib financial plot

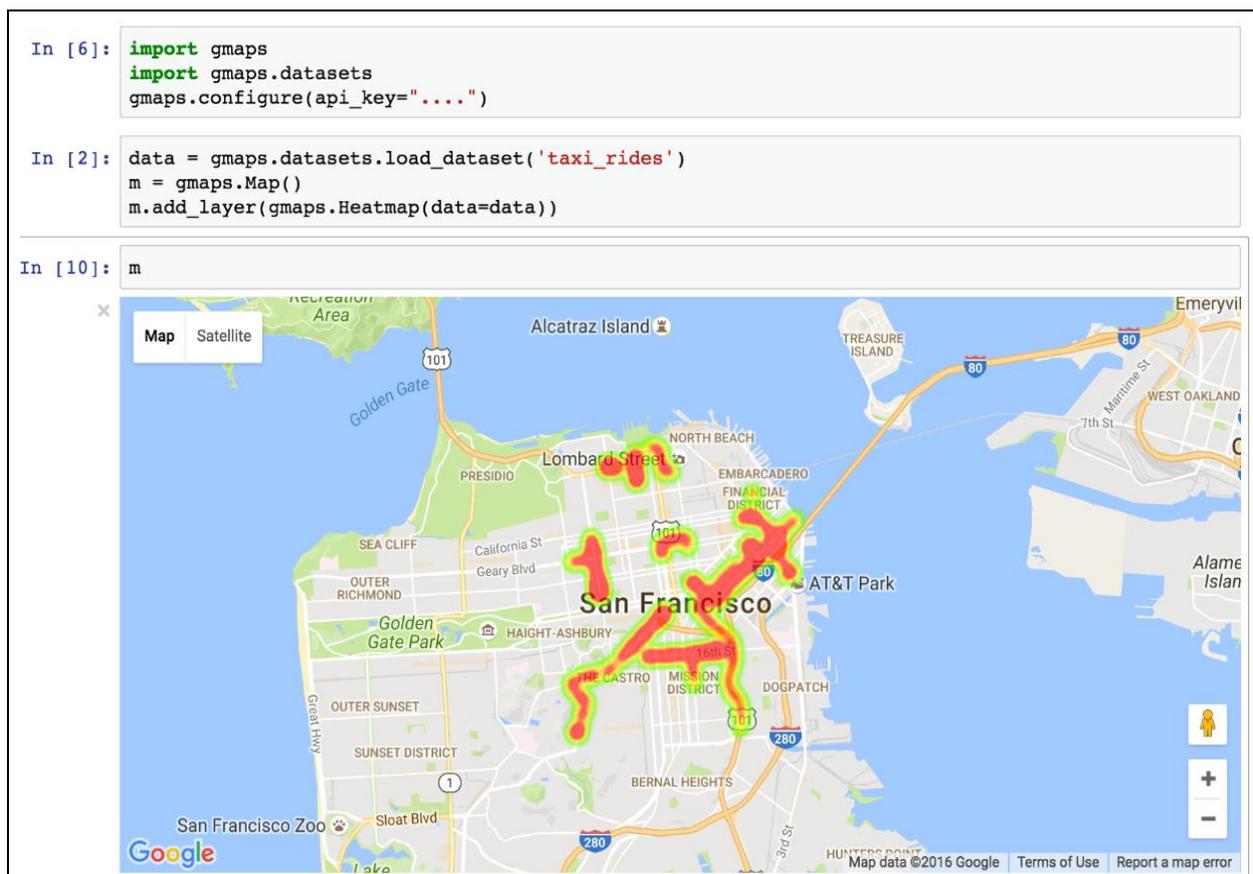


Fig. 3. Heatmap in Python

### Conclusions

The conclusion is that Python has some noteworthy features that make it widely used. The arguments we have presented prove that the program

written in Python will function exactly the same regardless of which operating system it is running in. Differences arise only in rare cases, and they are easy to anticipate due to the availability of detailed documentation.

## REFERENCES

1. WHAT ARE THE 10 MOST FAMOUS SOFTWARE PROGRAMS WRITTEN IN PYTHON? Available at: [http://www.hartmannsoftware.com/Blog/Articles\\_from\\_Software\\_Fans/Most-Famous-Software-Programs-Written-in-Python](http://www.hartmannsoftware.com/Blog/Articles_from_Software_Fans/Most-Famous-Software-Programs-Written-in-Python) (accessed 13 May 2018).
2. Рейтинг языков программирования 2018: Go и TypeScript вошли в высшую лигу, Kotlin стоит воспринимать серьезно Available at: <https://dou.ua/lenta/articles/language-rating-jan-2018/> (accessed 13 May 2018).
3. 10 awesome features of Python that you can't use because you refuse to upgrade to Python 3 Available at: <https://www.asmeurer.com/python3-presentation/slides.html#1> (accessed 13 May 2018).
4. Testing with Python: Just Makes Sense Available at: <https://blog.smartbear.com/testing/testing-with-python-and-why-it-makes-sense/> (accessed 13 May 2018).
5. HOWTO Use Python in the web Available at: <https://docs.python.org/2/howto/webrowsers.html> (accessed 13 May 2018).
6. Future Scope of Python Programming Language Available at: <https://legacy.gitbook.com/book/sarahjohan/what-is-the-scope-of-python-programming-language/details> (accessed 13 May 2018).
7. Applications for Python Available at: <https://www.python.org/about/apps/> (accessed 13 May 2018).

## СПИСОК ЛІТЕРАТУРИ

1. ЧТО ТАКОЕ 10 САМЫХ ИЗВЕСТНЫХ ПРОГРАММНЫХ ПРОГРАММ, НАПИСАННЫХ В ПИТОНЕ? [Электронный ресурс] - Режим доступа: [http://www.hartmannsoftware.com/Blog/Articles\\_from\\_Software\\_Fans/Most-Famous-Software-Programs-Written-in-Python](http://www.hartmannsoftware.com/Blog/Articles_from_Software_Fans/Most-Famous-Software-Programs-Written-in-Python).
2. Рейтинг языков программирования 2018: Go и TypeScript вошли в высшую лигу, Kotlin следует воспринимать всерьез. [Электронный ресурс] - Режим доступа: <https://dou.ua/lenta/articles/language-rating-jan-2018/>
3. 10 удивительных особенностей Python. Python 3 [Электронный ресурс] - Режим доступа: <https://www.asmeurer.com/python3-presentation/slides.html#1>.
4. Тестирование с помощью Python: просто имеет смысл. [Электронный ресурс] - Режим доступа: <https://blog.smartbear.com/testing/testing-with-python-and-why-it-make-sense/>
5. Как использование Python в Интернете [Электронный ресурс] - Режим доступа: <https://docs.python.org/2/howto/webrowsers.html>
6. Будущий масштаб языка программирования Python [Электронный ресурс] - Режим доступа: <https://legacy.gitbook.com/book/sarahjohan/what-is-the-scope-of-python-programming-language/details> (доступно 13 мая 2018 года).
7. Приложения для Python [Электронный ресурс] - Режим доступа: <https://www.python.org/about/apps/>

Рецензент: д-р техн. наук, проф. І. В. Шостак,  
 Національний аерокосмічний університет імені М. С. Жуковського «ХАІ», Київ  
 Received (Надійшла) 7.06.2018  
 Accepted for publication (Прийнята до друку) 22.08.2018

## Особенности програмування та нові можливості мови програмування Python

С. П. Альшин, О. О. Бородіна, А. С. Кікоть, І. Б. Жабран

**Мета.** Метою статті є необхідність продемонструвати рентабельність нових версій мови програмування Python, а також продемонструвати обсяги цієї мови разом з новим програмним забезпеченням. Зміст статті представляє оглядовий характер, що дозволяє зрозуміти особливості та нові можливості мови Python. **Результати.** Було встановлено, що для створення програм для різних цілей застосовують потужної інструмент програмування Python. **Наукова новизна.** Програми, написані на мові програмування Python працюють так само та незалежать від того, на якій операційній системі працює, а також в Python додані особливості, які роблять його широко застосовуваним. **Практична значимість.** Рентабельність нових версій Python та обсяги цієї мови разом з новим забезпеченням дає детальний аналіз областей, де застосовується Python. **Висновки.** Висновок полягає в тому, що у Python є деякі функції, які роблять його широко використовуваними. Наведені аргументи доводять, що програма, написана на Python, буде працювати однаково, незалежно від того, в якій операційній системі вона працює. Різниця виникає лише в рідкісних випадках, і їх легко передбачити через доступність докладної документації.

**Ключові слова:** Python, професійна мова, особливості, можливості Python.

## Особенности программирования и новые возможности языка программирования python

С. П. Алёшин, Е. А. Бородина, А. С. Кикоть, И. Б. Жабран

**Цель.** Целью статьи является необходимость продемонстрировать рентабельность новых версий языка программирования Python, а также продемонстрировать объемы этого языка вместе с новым программным обеспечением. Содержание статьи представляет собой обзорный характер, которая позволяет понять особенности и новые возможности языка Python. **Результаты.** Было установлено, что для создания программ для разных целей применяют мощный инструмент Python. **Научная новизна.** Программы, написанные на Python работают точно так же, независимо от того, на какой операционной системе работает, а также в Python добавлены особенности, которые делают его широко применяемым. **Практическая значимость.** Рентабельность новых версий Python и объемы этого языка вместе с новым обеспечением дает детальный анализ областей, где применяется Python. **Выводы.** Вывод заключается в том, что у Python есть некоторые функции, которые делают его широко используемым. Представленные аргументы доказывают, что программа, написанная на Python, будет работать одинаково, независимо от того, в какой операционной системе она работает. Различия возникают только в редких случаях, и их легко предвидеть из-за доступности подробной документации.

**Ключевые слова:** Python, профессиональный язык, особенности, возможности Python.

С. С. Бульба, В. В. Давидов, Г. А. Кучук

Національний технічний університет "Харківський політехнічний інститут", Харків, Україна

## МЕТОД РОЗПОДІЛУ РЕСУРСІВ МІЖ КОМПОЗИТНИМИ ЗАСТОСУНКАМИ

**Предмет** розгляду – методи оптимального розподілу ресурсів. **Метою статті** є розробка методу розподілу ресурсів між композитними застосунками, орієнтованому на динамічний вибір алгоритму оптимізації. методи, що використовуються, – жадібні та мурашині алгоритми, кластеризаційний підхід. **Результати роботи.** Розглянуто узагальнення існуючих евристик динамічного планування, які відносяться до класу жадібних алгоритмів та знаходять на кожному кроці локально оптимальне рішення. Проведено укрупнену класифікацію методів розподілу пакету композитних застосунків. Проаналізовано два типи кластеризації - вертикальну та горизонтальну. Розглянута можливість планування наборів композитних застосунків на базі кластеризаційного підходу. Наведено спосіб розподілу наборів композитних застосунків як з використанням мурашиного алгоритму, так і на базі кластеризаційного підходу з використанням мурашиного алгоритму. Проаналізовано переваги та недоліки кожного із розглянутих підходів, що дало змогу визначити межі застосування кожного із підходів. **Висновки.** Запропоновано чотири різних підходи до розподілу ресурсів між композитними застосунками з використанням таких методів: жадібні алгоритми, мурашині алгоритми, вертикальна та горизонтальна кластеризація, сумісне використання мурашиного алгоритму та кластеризації. Розроблений метод дозволяє динамічно провести найкращий вибір. Подальші дослідження будуть направлені на розробку відповідного алгоритму.

**Ключові слова:** композитний застосунок, жадібний алгоритм, мурашиний алгоритм, кластеризація.

### Вступ

Безперервно зростаючі вимоги в області складних обчислювальних систем призводять до необхідності розвитку нових методів реалізації обчислювальних ресурсів і сервісів, функціонування яких можуть задовольнити дані вимоги. Один з напрямків вирішення даної проблеми засноване на подальшому розвитку технологій розробки платформ розподілених обчислень для виконання композитних застосунків (КЗ). Під композитними застосунками в розподілених середовищах маються на увазі програми, які дозволяють користувачеві мати доступ до сервісів, розташованих на різних пристроях, не пов'язаних між собою. Даний підхід дає можливість в деякій мірі забезпечити процес обробки інформації завдяки тому, що вихід з робочого стану одного з сервісів не призводить до зриву процесу.

У процесі роботи в даному обчислювальному середовищі користувач надсилає запит на виконання необхідних обчислень, після чого дана система розподіляє отримане навантаження між наявними сервісами. Далі сервіси повертають оброблену інформацію назад, компонується відповідь і надається користувачеві. При цьому при паралельному виконанні декількох КЗ виникає задача розподілу ресурсів системи між КЗ.

**Аналіз літератури.** В [1 – 7] розглядаються різні підходи для розподілу ресурсів у інфокомунікаційних системах. При цьому застосовується цілий ряд оптимізаційних алгоритмів, зокрема, кожен із підходів, що пропонується, зазвичай є орієнтованим тільки на один із методів. Однак варто приділити увагу питанням, пов'язаним з динамічним вибором оптимізаційного алгоритму, при цьому вибір повинен бути орієнтовним на якість розподілу. Це і визначило завдання досліджень, тобто метою даної статті є розробка методу розподілу ресурсів між композитними застосунками, орієнтованому на динамічний вибір алгоритму оптимізації.

### 1. Жадібні алгоритми

Жадібний алгоритм — евристичний алгоритм, який приймає найкраще рішення, виходячи з наявних на поточному етапі даних [8]. Сутність жадібного алгоритму полягає локально оптимальному виборі на кожному кроці. Для реалізації алгоритму необхідно визначити такі елементи задачі:

- набір можливих варіантів, з яких робиться вибір;
- функція вибору, за допомогою якої знаходиться найкращий варіант;
- функція придатності, яка визначає придатність отриманого набору;
- функція цілі, яка оцінює цінність рішення та зазвичай не виражена явно;
- функція розв'язку, яка вказує на те, що знайдене кінцеве рішення.

Основний недолік жадібних алгоритмів – велика ймовірність того, що знайдені рішення не буде оптимальним. Основна перевага – швидкість знаходження рішення. Якщо постановка задачі потребує знаходження оптимального, а не раціонального рішення, то необхідно проведення аналізу можливостей його застосування.

До оптимізаційної задачі можна застосувати принцип жадібного вибору, якщо послідовність локально оптимальних виборів дає глобальний оптимальний розв'язок. Доведення оптимальності можна проводити за такою схемою: довести, що жадібний вибір на першому етапі не унеможливило шляху до оптимального розв'язку: для всякого розв'язку є інше, узгоджене із жадібним і не гірше першого, подалі довести, що підзадача, яка виникла після жадібного вибору на першому етапі, аналогічна початкової. Доведення проводиться методом математичної індукції [6].

Ідею, що лежить у використанні жадібних алгоритмів до розподілу ресурсів КЗ, можна описати таким чином: