

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”
Факультет інформатики та обчислювальної техніки

Сучасні технології створення програмних систем

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ПРАКТИЧНИХ РОБІТ
для студентів 1-го курсу магістратури
напряму “126 Інформаційні системи та технології”
(заочна форма навчання)**

КИЇВ 2018

Сучасні технології створення програмних систем (заочна форма навчання).

Практична робота № 1 Розробки алгоритмів мовою PYTHON

Мета. Розробити програму з використанням скриптової мови Python
Напишіть код за наступним алгоритмом:

ВАРІАНТИ 1

1. Є коробка зі сторонами: $A \times B \times C$. Визначити, чи пройде вона в двері з розмірами $M \times K$.
2. Чи можна з колоди, що має діаметр поперечного перерізу D , випилити квадратний брус шириною A ?
3. Чи можна в квадратному залі площею S помістити круглу сцену радіусом R так, щоб від стіни до сцени був прохід не менше K ?
4. Відомо грошова сума. Розміняти її купюрами 500, 100, 10 і монетою 2 руб., Якщо це можливо
5. Дано номер місця в плацкартному вагоні. Визначити, яке це місце: верхнє або нижнє, в купе або бічне.
6. Дано позитивні числа a, b, c, d . З'ясуйте, чи може прямокутник зі сторонами a, b вміститися всередині прямокутника зі сторонами c, d так, щоб кожна сторона внутрішнього прямокутника була паралельна або перпендикулярна стороні зовнішнього прямокутника.

Теоретичні відомості :

9) Мусін Д. Самоучитель Python 2015 Розділи 3,4, 5, 6, 7, 8,9 [с.1..с.28]

5) У.Сэнд Hello word! Занимательное программирование 2016, Глава 1..Глава 5 [с.6..с.69]

Практична робота № 2 Використання віконного менеджера TkInter для побудови графіка функцій

МЕТА : Розробити програму для побудови табулярами функції та графічного її відображення у вигляді графіку згідно даних табл. 1

В програмі необхідно визначити особливі точки, задати інтервал та крок табуляції.

Для відображення результатів роботи програми використати віконий менеджер PyQt або TkInter.

Теоретичні відомості :

5)У.Сэнд Hello word!Занимательное программирование 2016, Глава 1..Глава 6,7,8[с.6..с.105], Глава 11 [с.115..128],Глава 13 [с.155..169],Глава 16 [с.201..227],Глава 20 [с.284..305]

10)Н. Прохоренок Python и 3 PyQt. Разработка приложений. 2012 , Глава 11 [с.197..215],Глава 19.20 [с.359..425]

3)Федоров Д. Основы программирования на примере языка программирования Python 2018, Глава 20 [с.133..143]

1)Программирование на языке Python 2017

ВАРІАНТИ 2

Функція

1) $Y = \ln(x)/x$

2) $Y = \text{tg}(x)/x$

3) $Y = x/\cos(x)$

4) $Y = x/\text{tg}(x)$

5) $Y = x/\sin(x)$

6) $Y = x^2/\sin(x^2)$

Практична робота № 3 Регулярні вирази в обробці текстових документів

Тема : Використання регулярних виразів та технологія обробки текстових документів

Розробити програму мовою Python для обробки текстових елементів з використанням регулярних виразів

Розробити програму :

- 1) для вирішення завдань обробки шаблоних текстів
- 2) Використати віконний менеджер TkInter.

Теоретичні відомості :

3) Федоров Д. Основы программирования на примере языка программирования Python 2018, Глава 20 [с.133..143]

9) Мусін Д. Самоучитель Python 2015 Розділи 10..16 [с.28..с.46]

10) Н. Прохоренко Python и 3 PyQt. Разработка приложений. 2012 , Глава 6 [с.76..113], Глава 7 [с.114..131]

ВАРІАНТИ 3

- 1 Розпізнати такі рядки: "bat", "Bit ", "but", "hat", "hi t" або "hut".
- 2 Забезпечити зіставлення з будь-якою парою слів, розділених одним пробілом, такими як ім'я та прізвище.
- 3 Забезпечити зіставлення з будь-яким словом і однією літерою, розділеними комою і одним пропуском, як прийнято в США в написанні прізвища та першого ініціалу.
- 4 Забезпечити зіставлення з усіма допустимими ідентифікаторами Python.
- 5 Забезпечити зіставлення з частиною поштової адреси, що позначає вулицю, згідно з форматом, прийнятим в конкретній країні (регулярний Вираз має бути достатньо загальним, щоб забезпечити зіставлення з назвами вулиць, що складаються з будь-якої кількості рядків, включаючи тип вулиці: проспект, провулок, бульвар і т.д.).

6 Забезпечити зіставлення з простими доменними іменами Інтернету, які починаються з підрядка "www." і закінчуються суфіксом ". Com"; наприклад www. yahoo. com. Додаткова можливість більш успішно виконати завдання полягає в наступному: якщо складений регулярний вираз буде також підтримувати доменні імена більш високих рівнів, такі як. edu,. net і т.д. (Наприклад, www. Foothill.edu)

Додаток 1. Приклади програм

Приклад 1. Метод Ейлера (з використанням віконного менеджера PyQt)

```
import sys
from PyQt4 import QtGui, QtCore
from PyQt4.QtCore import pyqtSlot
from PyQt4.QtGui import *

class PenStyles(QtGui.QWidget):
    def __init__(self, parent=None):
        QtGui.QWidget.__init__(self, parent)

        self.setGeometry(28, 28, 500, 500)
        self.setWindowTitle('penstyles')

        btn = QPushButton('Рішення задачі', self)
        self.text = QLineEdit('(x*y/10+y/3+1)/100', self)
        self.text.move(100, 2)
        self.function = "(x*y/10+y/3+1)/100";

        QtCore.QObject.connect(btn,QtCore.SIGNAL("clicked()"),
self.process)

    def process(self):
```

```

self.function = self.text.text()
self.repaint()

def paintEvent(self, event):
    paint = QtGui.QPainter()

    paint.begin(self)

    pen = QtGui.QPen(QtCore.Qt.black, 2, QtCore.Qt.SolidLine)
    paint.setPen(pen)
    paint.drawLine(250, 500, 250, 0)
    paint.drawLine(0, 250, 500, 250)

    pen = QtGui.QPen(QtCore.Qt.black, 1, QtCore.Qt.SolidLine)
    paint.setPen(pen)
    for i in range(10):
        k = i*50 - 250
        paint.drawLine(k + 250, -3 + 250, k + 250, 3 + 250)
        paint.drawText(k + 255, -10 + 250, str(k))
        #canv.create_text(k + 265, -10 + 250, text = str(k), fill="purple",
font=("Helvetica", "10"))
        if (k != 0):
            paint.drawLine(-3 + 250, 500-(k + 250), 3 + 250, 500-(k + 250))
            paint.drawText(7 + 250, 500-(k + 250), str(k))

    paint.setPen(QtCore.Qt.red)
    paint.setBrush(QtGui.QColor(223, 135, 19, 255))

    #x increase
    x = 0
    y = 0
    step = 1
    for i in range(250):
        try:
            new_f = self.function.replace('x', str(x)).replace('y', str(y))

```

```

        newx = x+step
        newy =(y+step*eval(new_f))
        paint.drawLine(x+250, 500-(y+250), newx+250, 500-
(newy+250))
        x = newx
        y = newy
        paint.drawRect(x+250, 500-(y+250),2,2)

        if y < -250:
            break
        if y > 250:
            break

    except:
        pass

#x decrease
x = 0
y = 0
step = 1
for i in range(250):
    try:
        new_f = self.function.replace('x', str(x)).replace('y', str(y))
        newx = x-step
        newy = (y-step*eval(new_f))
        paint.drawLine(x+250, 500-(y+250), newx+250, 500-
(newy+250))
        x = newx
        y = newy
        paint.drawRect(x+250, 500-(y+250),2,2)

        if y < -250:
            break
        if y > 250:
            break

```

```
        except:
            pass

    paint.end()

app = QtGui.QApplication(sys.argv)
dt = PenStyles()
dt.show()
app.exec_()
```

Приклад 2. Метод Ейлера з використанням віконного менеджера TkInter

```
from math import *
from tkinter import *

root = Tk()

canv = Canvas(root, width = 500, height = 500, bg = "white")

e = Entry(root, width=80)
e.pack()

e.insert(0, "(x*y/10+y/3+1)/100")

def callback():
    f = e.get()
    canv.delete("all")
    canv.create_line(250,500,250,0,width=1,arrow=LAST)
    canv.create_line(0,250,500,250,width=1,arrow=LAST)

    for i in range(10):
        k = i*50 - 250
```



```

        canv.create_line(k + 250, -3 + 250, k + 250, 3 + 250, width = 0.5,
fill = 'black')
        canv.create_text(k + 265, -10 + 250, text = str(k), fill="purple",
font=("Helvetica", "10"))
        if (k != 0):
            canv.create_line(-3 + 250, 500-(k + 250), 3 + 250, 500-(k +
250), width = 0.5, fill = 'black')
            canv.create_text(20 + 250, 500-(k + 250), text = str(k),
fill="purple", font=("Helvetica", "10"))

#x increase
x = 0
y = 0
step = 1
for i in range(250):
    try:
        new_f = f.replace('x', str(x)).replace('y', str(y))
        newx = x+(step)
        newy =(y+(step*eval(new_f)))
        canv.create_line(x+250,500-(y+250),newx+250,500-
(newy+250),width = 0.5, fill = 'red')
        x = newx
        y = newy
        canv.create_rectangle(x+250, 500-(y+250), x + 2+250, 500-
(y+250) + 2, fill = 'black')

        if y < -250:
            break
        if y > 250:
            break

    except:
        pass

```

```

#x decrease
x = 0
y = 0
step = 1
for i in range(250):
    try:
        new_f = f.replace('x', str(x)).replace('y', str(y))
        newx = x-step
        newy = (y-step*eval(new_f))
        canv.create_line(x+250,500-(y+250),newx+250,500-
(newy+250),width = 0.5, fill = 'red')
        x = newx
        y = newy
        canv.create_rectangle(x+250, 500-(y+250), x + 2+250, 500-
(y+250) + 2, fill = 'black')

        if y < -250:
            break
        if y > 250:
            break
    except:
        pass

```

```

b = Button(root, text="введіть першу похідну  $y' = f(x,y)$  та натисніть цю
кнопку", width=60, command=callback)
b.pack()

```

```

canv.pack()
root.mainloop()

```

Приклад 3. Використання регулярних виразів для пошуку зразків в строках

```

import re

match = re.search(r'\d\d\D\d\d', r'Телефон 123-12-12')
print(match[0] if match else 'Not found')
# -> 23-12
match = re.search(r'\d\d\D\d\d', r'Телефон 1231212')
print(match[0] if match else 'Not found')
# -> Not found

match = re.fullmatch(r'\d\d\D\d\d', r'12-12')
print('YES' if match else 'NO')
# -> YES
match = re.fullmatch(r'\d\d\D\d\d', r'Т. 12-12')
print('YES' if match else 'NO')
# -> NO

print(re.split(r'\W+', 'Де, скажіть мені, мої окуляри??!'))
# -> ['Де', 'скажіть', 'мені', 'мої', 'окуляри', '']

print(re.findall(r'\d\d\.\d\d\.\d{4}',
                 r'Ця строка написана 19.01.2018, а могла би і 01.09.2017'))
# -> ['19.01.2018', '01.09.2017']

for m in re.finditer(r'\d\d\.\d\d\.\d{4}', r'Ця строка написана 19.01.2018, а могла би і 01.09.2017'):
    print('Дата', m[0], 'починається з позиції', m.start())
# -> Дата 19.01.2018 починається з позиції 20
# -> Дата 01.09.2017 починається з позиції 45

```

```
print(re.sub(r'\d\d\.\d\d\.\d{4}',
            r'DD.MM.YYYY',
            r'Ця строка написана 19.01.2018, а
            могла би і 01.09.2017'))
# -> Ця строка написана DD.MM.YYYY, а могла би і
DD.MM.YYYY
```

Приклад 4. Табулювання функції однієї змінної

```
# -*- coding: utf-8 -*-
import math
def p1():
    x = 1.0
    while x < 10.0:
        print x, "\t", math.log(x)
        x += 1.0

def main():
    p1()
    return 0

if __name__ == '__main__':
    main()
```

Додаток 2. Теми лекцій

- [1. Прості типи даних](#)
- [2. Математичні функції](#)
- [3. Рядки](#) та робота з ними
- [4. Списки і кортежі](#)

5.Словники і множини

6.Структури коду

- 6.1. Створення і перевірка умов
- 6.2. Вказівка розгалуження
- 6.3. Вказівка повторення
- 6.4. Генератори
- 6.5. Функції
- 6.6. Модулі і пакет
7. Стандартна бібліотека Python
8. Структуровані текстові файли
9. ООП і об'єкти
10. Регулярні вирази
11. Віконний менеджер TkInter , PyQt