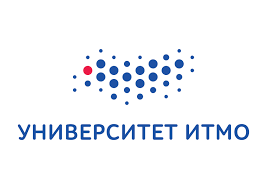
Министерство высшего образования и науки

Национальный исследовательский университет ИТМО

Мегафакультет компьютерных технологий и управление

Факультет программной инженерии и компьютерной техники



**название работы:**

**Использование встроенных средств автоматизации в приложении**

**Студент Ф.И.О:** Хоанг Ван Куан

**Группа №:** P3166

**Преподаватель Ф.И.О:** Климов Иговь Викторов

Санкт-Петербург, 2022

Catalog

[Задача : 3](#_Toc4901)

[I. Текст задания 3](#_Toc19976)

[II. Эскиз орнамента с определением координат 3](#_Toc16232)

[III. Текст расширения 3](#_Toc27610)

[IV. Print Screen 6](#_Toc28862)

**Задача :**

1. Текст задания

Придумайте линейный (горизонтальный) орнамент и разработайте расширение Inkscape для его рисования.

Расширение должно запрашивать у пользователя:

* Количество раппортов
* Высоту раппорта
* Ширину раппорта

1. Эскиз орнамента с определением координат

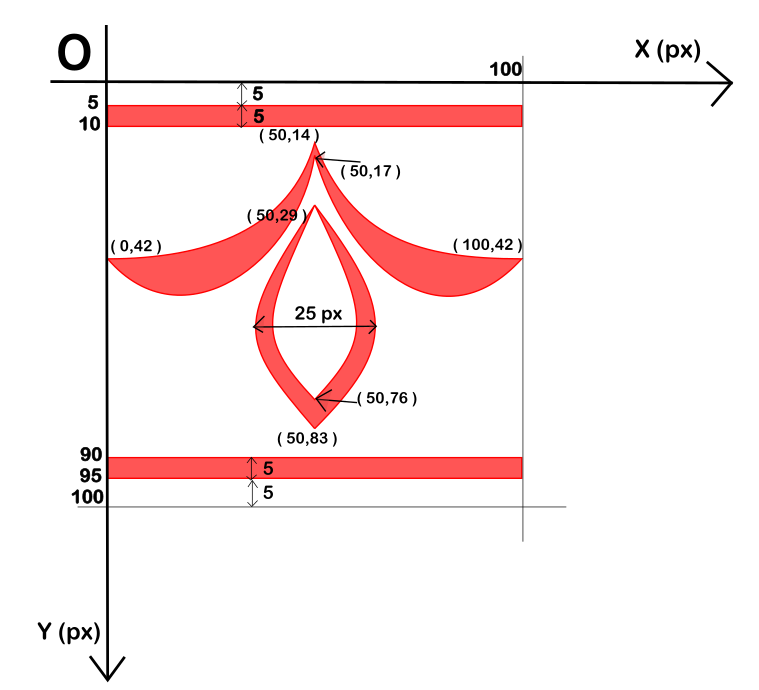


Рис 1: Эскиз орнамента с определением координат

1. Текст расширения

***1 ) Текст расширения .inx :***

<?xml version="1.0" encoding="UTF-8"?>

<inkscape-extension xmlns="http://www.inkscape.org/namespace/inkscape/extension">

<name>My Test</name>

<id>My.test</id>

<param name="s\_l" type="float" min="0.01" max="10000" gui-text="Side Length (px):">100.0</param>

<param name="c\_x" type="float" min="-1000" max="1000" gui-text="X :">0.0</param>

<param name="c\_y" type="float" min="-1000" max="1000" gui-text="Y :">0.0</param>

<param name="count" type="int" min="1" max="1000" gui-text="Count :">1</param>

<effect>

<object-type>all</object-type>

<effects-menu>

<submenu name="Render"/>

</effects-menu>

</effect>

<script>

<command location="inx" interpreter="python">test.py</command>

</script>

</inkscape-extension>

***2) Текст расширения .py :***

#!/usr/bin/env python

# coding=utf-8

import sys

import inkex

from lxml import etree

def draw\_SVG\_test(a,x,y,c, cur):

style = {'stroke': '#FF0000', 'stroke-width': '1', 'fill': '#FF5555'}

i = 0

while(i < c):

elem = cur.add(inkex.PathElement())

elem.update(\*\*{

'style': style,

'inkscape:label': 'MyTest',

'd': 'M' + str(x) + ',' + str(y + a\*0.42) +

'C' + str(x + a\*0.24) + ',' + str(y + a\*0.42) + ' ' + str(x + a\*0.44) + ',' + str(y + a\*0.35) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.14) +

'C' + str(x + a\*0.59) + ',' + str(y + a\*0.37) + ' ' + str(x + a\*0.75) + ',' + str(y + a\*0.42) + ' ' + str(x + a) + ',' + str(y + a\*0.42) +

'C' + str(x + a\*0.8) + ',' + str(y + a\*0.64) + ' ' + str(x + a\*0.57) + ',' + str(y + a\*0.42) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.17) +

'C' + str(x + a\*0.46) + ',' + str(y + a\*0.43) + ' ' + str(x + a\*0.18) + ',' + str(y + a\*0.63) + ' ' + str(x) + ',' + str(y + a\*0.42) +

'M' + str(x + a\*0.5) + ',' + str(y + a\*0.83) +

'C' + str(x + a\*0.31) + ',' + str(y + a\*0.61) + ' ' + str(x + a\*0.31) + ',' + str(y + a\*0.58) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.29) +

'C' + str(x + a\*0.38) + ',' + str(y + a\*0.56) + ' ' + str(x + a\*0.35) + ',' + str(y + a\*0.6) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.76) +

'C' + str(x + a\*0.64) + ',' + str(y + a\*0.61) + ' ' + str(x + a\*0.63) + ',' + str(y + a\*0.56) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.29) +

'C' + str(x + a\*0.69) + ',' + str(y + a\*0.54) + ' ' + str(x + a\*0.7) + ',' + str(y + a\*0.63) + ' ' + str(x + a\*0.5) + ',' + str(y + a\*0.83) +

'M' + str(x) + ',' + str(y + a\*0.05) +

'L' + str(x) + ',' + str(y + a\*0.1) + 'L' + str(x + a) + ',' + str(y + a\*0.1) + 'L' + str(x + a) + ',' + str(y + a\*0.05) + 'Z' +

'M' + str(x) + ',' + str(y + a\*0.9) +

'L' + str(x) + ',' + str(y + a\*0.95) + 'L' + str(x + a) + ',' + str(y + a\*0.95) + 'L' + str(x + a) + ',' + str(y + a\*0.9) + 'Z'

})

x = x + a

i = i + 1

return elem

class MyTest(inkex.EffectExtension):

def add\_arguments(self, pars):

pars.add\_argument("--s\_l", type=float, default=100.0, help="Side Length")

pars.add\_argument("--c\_x", type=float, default=0.0, help="X")

pars.add\_argument("--c\_y", type=float, default=0.0, help="Y")

pars.add\_argument("--count", type=int, default=1, help="Count")

def effect(self):

cur = self.svg.get\_current\_layer()

a = self.svg.unittouu(str(self.options.s\_l) + 'px')

x = self.svg.unittouu(str(self.options.c\_x) + 'px')

y = self.svg.unittouu(str(self.options.c\_y) + 'px')

c = self.options.count

draw\_SVG\_test(a,x,y,c, cur)

if \_\_name\_\_ == '\_\_main\_\_':

MyTest().run()

1. Print Screen

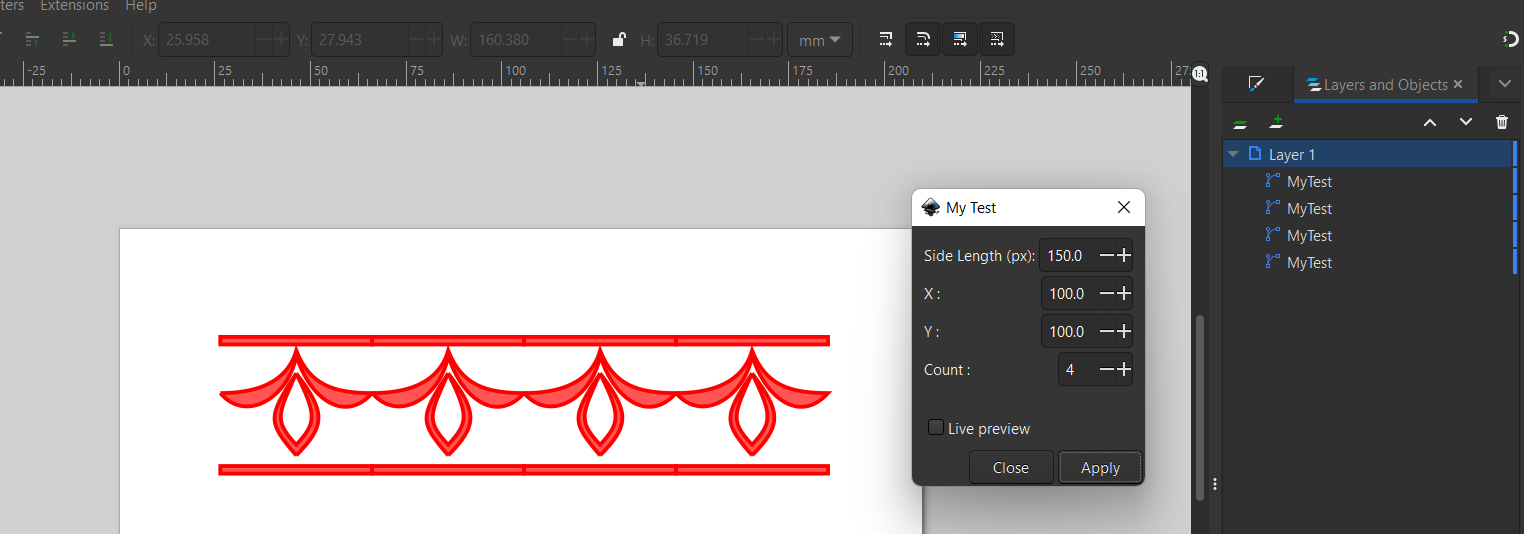


Рис 2: «printscreen» окон с введенными данными для расширения и нарисованного расширением орнамента