

Содержание

ВВЕДЕНИЕ.....	5
1.ПОСТАНОВКА ЗАДАЧИ.....	6
2.БЛОК СХЕМА	7
3.ЛИСТИНГ ПРОГРАММЫ.....	8
4.РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ.....	12
ЗАКЛЮЧЕНИЕ.....	13
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	14

ВВЕДЕНИЕ

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Питоне организовывается в функции и классы, которые могут объединяться в модули (которые в свою очередь могут быть объединены в пакеты).

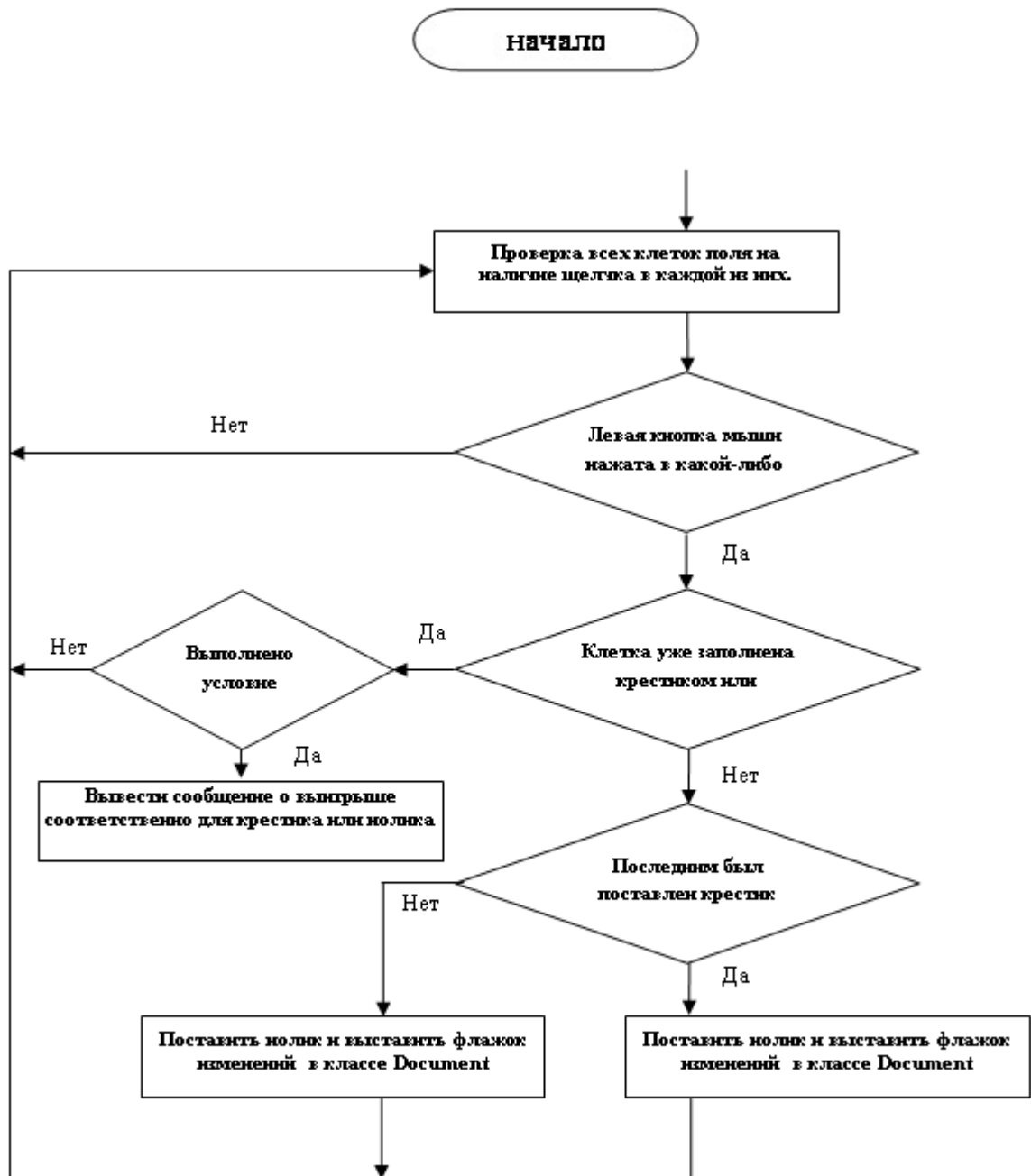
Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализации интерпретаторов для JVM (с возможностью компиляции), MSIL (с возможностью компиляции), LLVM и других. Проект PyPy предлагает реализацию Питона на самом Питоне, что уменьшает затраты на изменения языка и постановку экспериментов над новыми возможностями.

Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие официальные стандарты, их роль выполняет CPython.

1. ПОСТАНОВКА ЗАДАЧИ

Создать интерактивную игру «Крестики-нолики», на языке программирования Python

2.БЛОК-СХЕМА



Блок схема 1.Алгоритм работы программы.

4.ЛИСТИНГ ПРОГРАММЫ.

```
# -*- coding: utf-8 -*-
import pygtk
pygtk.require('2.0')
import gtk

class XO_Field:
    list = [[' ', ' ', ' ', ' ', ' '],
            [' ', ' ', ' ', ' ', ' '],
            [' ', ' ', ' ', ' ', ' ']]
    win = False
    chr = ' '

    def check(self):
        a = self.list
        for i in range(3):
            if (a[i][0] == a[i][1] == a[i][2]):
                return a[i][0]

        for i in range(3):
            if (a[0][i] == a[1][i] == a[2][i]):
                return a[0][i]

        if (a[0][0] == a[1][1] == a[2][2]):
            return a[1][1]
        if (a[0][2] == a[1][1] == a[2][0]):
            return a[1][1]

        return False

    def post(self, symbol, x, y):
        self.list[y][x] = symbol
        self.chr = self.check()
        if ((self.chr != False) & (self.chr != ' ')):
            self.win = True

class XO_Win:

    def create_button(self, box):
        button = gtk.Button(self.field.chr)
        box.pack_start(button, True, True, 0)
```

```

        button.show()
        return button

def __init__(self):
    self.field = XO_Field()
    self.set = True
    self.boxes = []
    self.buttons = [[], [], []]

    self.window =
gtk.Window(gtk.WINDOW_TOPLEVEL)
    self.window.connect("destroy",
self.destroy)
    self.window.set_title("Крестики-
Нолики.py!")
    self.window.set_border_width(10)
    self.window.set_size_request(400, 400)

    self.vbox = gtk.VBox(False, 0)
    self.window.add(self.vbox)

    for i in range(3):
        box = gtk.HBox(False, 0)
        self.boxes.append(box)
        for j in range(3):
            self.buttons[i].append(self.create_
button(box) )
            self.buttons[i]
[j].connect("clicked", self.on_btn_click, i, j)
            self.vbox.pack_start(box, True, True, 0)
            box.show()

    self.vbox.show()
    self.window.show()

def main(self):
    gtk.main()

def destroy(self, widget, data=None):
    gtk.main_quit()

def postgui(self, chr, i, j):
    self.buttons[i][j].set_label(chr)

```

```

def on_btn_click(self, button, i, j):
    if self.field.win == True:
        return

    if self.field.list[i][j] == ' ':
        if self.set == True:
            self.field.post('X', j, i)
            self.buttons[i][j].set_label("X")
            map = button.get_colormap()
            color = map.alloc_color("red")
            style = button.get_style().copy()
            style.bg[gtk.STATE_NORMAL] = color
            self.buttons[i]
[j].set_style(style)
            self.set = False
        else:
            self.field.post('O', j, i)
            self.buttons[i][j].set_label('O')
            map = button.get_colormap()
            color = map.alloc_color("green")
            style = button.get_style().copy()
            style.bg[gtk.STATE_NORMAL] = color
            self.buttons[i]
[j].set_style(style)
            self.set = True

    both = 0
    if self.field.win == True:
        self.window.set_title("Выиграл: " +
self.field.chr + '!')
    else:
        for i in range(3):
            for j in range(3):
                both = both + 1
        if both == 0:
            self.window.set_title("Ничья!")

win = XO_Win()
win.main()

```

5.РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ.

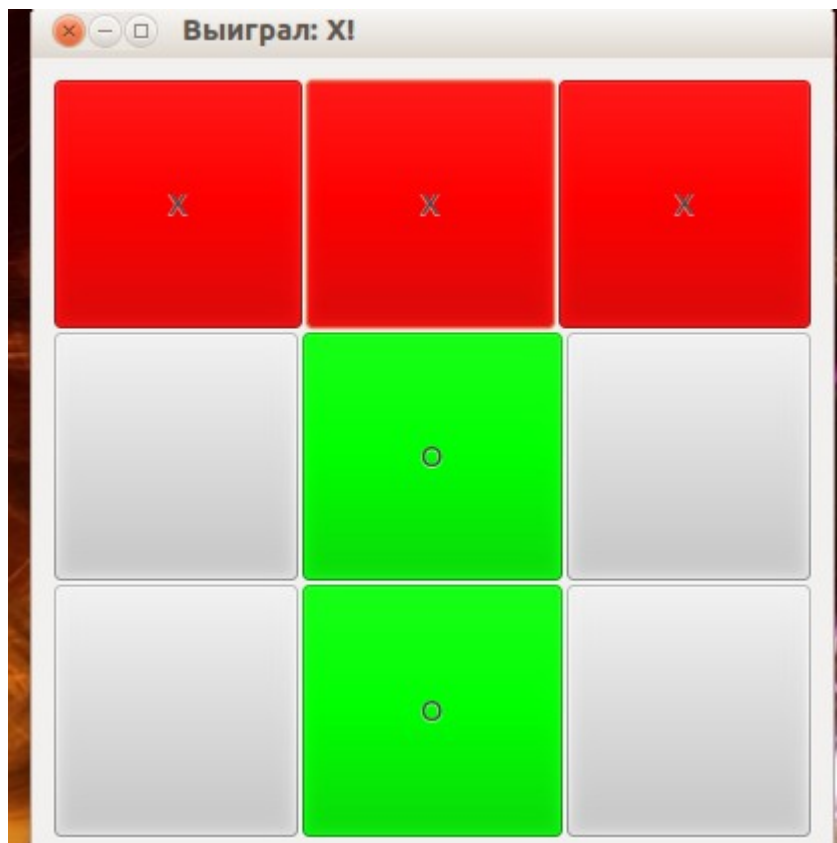


Рис. Результат работы программы.

На терминале представлено рабочее окно выполненной программы. Окно 3x3, программа показывает какой из элементов выстроен в ряд, и каждый элемент имеет свой цвет..

ЗАКЛЮЧЕНИЕ

В ходе данной курсовой работы был изучен метод написания кода программы для интерактивной игры «Крестики -нолики» при помощи языка программирования Python. Для удобства каждый элемент имел свой цвет.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Россум Г., Дрейк Ф.Л.Дж. Язык программирования Python.
2. Сузи Р.А. Язык программирования Python.
3. Gift N., Jones J. M. Python for Unix and Linux System Administration. Vaingast S. Beginning Python Visualization - Crafting Visual Transformation Scripts.