Федеральное государственное образовательное учреждение

высшего профессионального образования

«**КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

Кафедра систем управления и вычислительной техники

|  |  |
| --- | --- |
| Курсовая работа  допущена к защите  руководитель | Курсовая работа защищена  с оценкой …………………  руководитель |

**«Игра в «крестики-нолики»**

Курсовая работа по дисциплине

**“Высокоуровневые технологии программирования”**

Пояснительная записка

53.09.03.03.03

Нормконтроллер:

доцент, к.т.н.,

Высоцкий Л.Г.

Проект выполнил:

студент группы 19-ИЭ-1

Богданов М. Д.

Калининград

2021

**Содержание**

[Реферат 3](#_Toc74506208)

[Обоснование выбора языка программирования 4](#_Toc74506209)

[Постановка задачи 7](#_Toc74506210)

[Описание модулей программы и их взаимосвязь 8](#_Toc74506211)

[Вид программного интерфейса 9](#_Toc74506212)

[Блок-схемы модулей 10](#_Toc74506213)

[Листинг программы 11](#_Toc74506214)

[Вид результатов выполнения программы 14](#_Toc74506215)

[Заключение 17](#_Toc74506216)

[Список использованных источников 19](#_Toc74506217)

# РЕФЕРАТ

«Игра в «крестики-нолики». Для двух участников. Поле размером не менее 40х40. Компьютер определяет наличие пяти одинаковых символов в линию, при помощи проверки всех возможных выигрышных состояний поля.

Требования к техническому обеспечению:

* RAM: 3+ ГБ
* CPU: 2+ ядер

Требования к программному обеспечению:

* Windows - 64-битная x86, 32-битная x86; MacOS - 64-битная x86; Linux - 64-битная x86, 64-битная Power8 / Power9
* Установленная среда разработки Python версии 3.0 и выше

# ОБОСНОВАНИЕ ВЫБОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Был выбран язык разработки Python, по следующим причинам:

Высокая скорость разработки.

По сравнению с компилирующими или строго типизированными языками, такими как С, C++ и Java, Python во много раз повышает производительность труда разработчика. Объем программного кода на языке Python обычно составляет треть или даже пятую часть эквивалентного программногокода на языке C++ или Java. Это означает меньший объем ввода с клавиатуры, меньшее количество времени на отладку и меньший объем трудозатрат на сопровождение. Кроме того, программы на языке Python запускаются сразу же, минуя длительные этапы компиляции и связывания, необходимые в некоторых других языках программирования, что еще больше увеличивает производительность труда программиста.

Переносимость программ.

Большая часть программ на языке Python выполняется без изменений на всех основных платформах. Перенос программного кода из операционной системы Linux в Windows обычно заключается в простом копировании файлов программ с одной машины на другую. Более того, Python предоставляет массу возможностей по созданию переносимых графических интерфейсов, программ доступа к базам данных, веб-приложений и многих других типов программ. Даже интерфейсы операционных систем, включая способ запуска программ и обработку каталогов, в языке Python реализованы переносимым способом.

Библиотеки поддержки. В составе Python поставляется большое число собранных и переносимых функциональных возможностей, известных как стандартная библиотека. Эта библиотека предоставляет массу возможностей, востребованных в прикладных программах, начиная от поиска текста по шаблону и заканчивая сетевыми функциями. Кроме того, Python допускает расширение как за счет ваших собственных библиотек, так и за счет библиотек, созданных сторонними разработчиками. Из числа сторонних разработок можно назвать инструменты создания веб-сайтов, программирование математических вычислений, доступ к последовательному порту, разработку игровых программ и многое другое. Например, расширение NumPy позиционируется как свободный и более мощный эквивалент системы программирования математических вычислений Mathlab.

Интеграция компонентов.

Сценарии Python легко могут взаимодействовать с другими частями приложения благодаря различным механизмам интеграции. Эта интеграция позволяет использовать Python для настройки и расширения функциональных возможностей программных продуктов. На сегодняшний день программный код на языке Python имеет возможность вызывать функции из библиотек на языке C/C++, сам вызываться из программ, написанных на языке C/C++, интегрироваться с программными компонентами на языке Java, взаимодействовать с такими платформами, как СОМ и .NET, и производить обмен данными через последовательный порт или по сети с помощью таких протоколов, как SOAP, XML-RPC и CORBA. Python - не обособленный инструмент.

Качество программного обеспечения.

По своей природе Python имеет простой, удобочитаемый синтаксис и ясную модель программирования. Согласно лозунгу, выдвинутому на недавней конференции по языку Python, основное его преимущество состоит в том, что Python «каждому по плечу» - характеристики языка взаимодействуют ограниченным числом непротиворечивых способов и естественно вытекают из небольшого круга базовых концепций. Это делает язык простым в освоении, понимании и запоминании. На практике программистам, использующим язык Python, почти не приходится прибегать к справочным руководствам - это непротиворечивая система, на выходе которой, к удивлению многих, получается профессиональный программный код. Философия Python по сути диктует использование минималистского подхода. Это означает, что даже при наличии нескольких вариантов решения задачи в этом языке обычно существует всего один очевидный путь, небольшое число менее очевидных альтернатив и несколько взаимосвязанных вариантов организации взаимодействий. Более того, Python не принимает решения за вас, когда порядок взаимодействий неочевиден - предпочтение отдается явному описанию, а не «волшебству». В терминах Python явное лучше неявного, а простое лучше сложного.1 Помимо философии Python обладает такими возможностями, как модульное и объектно-ориентированное программирование, что естественно упрощает возможность многократного использования программного кода. Поскольку качество находится в центре внимания самого языка Python, оно также находится в центре внимания программистов.

Высокая скорость разработки

Во время бума развития Интернета во второй половине 1990-х годов было сложно найти достаточное число программистов для реализации программных проектов - от разработчиков требовалось писать программы со скоростью развития Интернета. Теперь, в эпоху экономического спада, картина изменилась. Сегодня от программистов требуется умение решать те же задачи меньшим числом сотрудников. В обоих этих случаях Python блистал как инструмент, позволяющий программистам получать большую отдачу при меньших усилиях. Он изначально оптимизирован для достижения высокой скорости разработки - простой синтаксис, динамическая типизация, отсутствие этапа компиляции и встроенные инструментальные средства позволяют программистам создавать программы за меньшее время, чем при использовании некоторых других инструментов. В результате Python увеличивает производительность труда разработчика во много раз по сравнению с традиционными языками программирования. Это значительное преимущество, которое с успехом может использоваться как во время бума, так и во время спада, а также во время любого промежуточного этапа развития индустрии программного обеспечения.

# ПОСТАНОВКА ЗАДАЧИ

Данная программа – это игра для двух участников. Поле размером 40х40 клеток. Компьютер определяет наличие пяти одинаковых символов в линию и выделяет их цветом, обозначая победителя.

1. Перед пользователем появляется игровое поле из квадратов 40 на 40
2. При нажатии на любой из квадратов, на его месте появляется крестик или нолик
3. Цель игры – составить ряд из пяти одинаковых знаков и помешать сделать то же самое оппоненту
4. При нажатии на команду главного меню появляется информация об авторе программы
5. При нажатии на кнопку “Новая игра” поле очищается

# ОПИСАНИЕ МОДУЛЕЙ ПРОГРАММЫ И ИХ ВЗАИМОСВЯЗЬ

Описание модулей программы представлено в таблице 1.

|  |  |  |
| --- | --- | --- |
| Модули | Чем представлены модули | Описание модулей |
| Unit 1 | Вложенный цикл в основной части программы | Создание игрового поля 40 на 40 кнопок посредством создания массива кнопок с одинаковыми свойствами |
| Unit 2 | Функция click() | Нажатие на кнопку на игровом поле. Далее определяется очерёдность хода игрока и в выбранную кнопку вписывается символ. |
| Unit 3 | Функция win\_check() | Проверка выигрышного состояния посредством проверки всех возможных выигрышный комбинаций на поле |
| Unit 4 | Функция line\_check() | Проверка линии на одинаковость символа в них и, если проверка пройдена, выделение выигрышной комбинации и конец игры |
| Unit 5 | Функция restart() | Приведение поля в исходное состояние посредством очищения надписей на всех кнопках и перезапуска игры |
| Unit 6 | Функция autor() | Вывод информации об авторе программы в информационном окне messagebox |

Табл. 1 – Список модулей и их описание

Связи модулей:

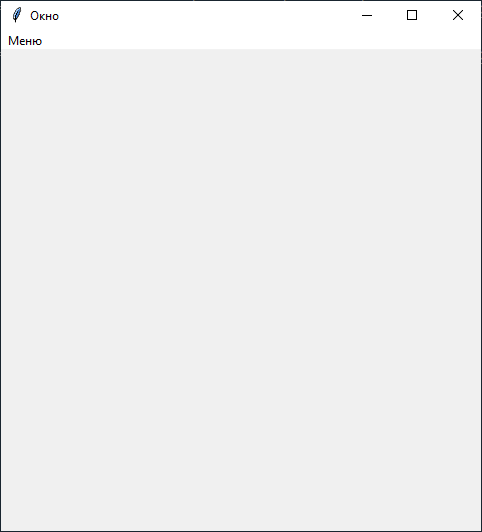
* Все модули находятся в области работы основной части программы
* Функция click(), при каждом вызове, задействует функцию win\_check()
* Функция win\_check(), при каждом вызове, задействует функцию line\_check()

Работа модулей реализуется посредством стандартных библиотек языка Python 3, а также библиотеки Tkinter и её модуля messagebox.

# ВИД ПРОГРАММНОГО ИНТЕРФЕЙСА

Для расположения необходимых виджетов и дальнейшего функционировании программы необходимо:

* Окно Tkinter
* Расположенное на нём меню



# БЛОК-СХЕМЫ МОДУЛЕЙ

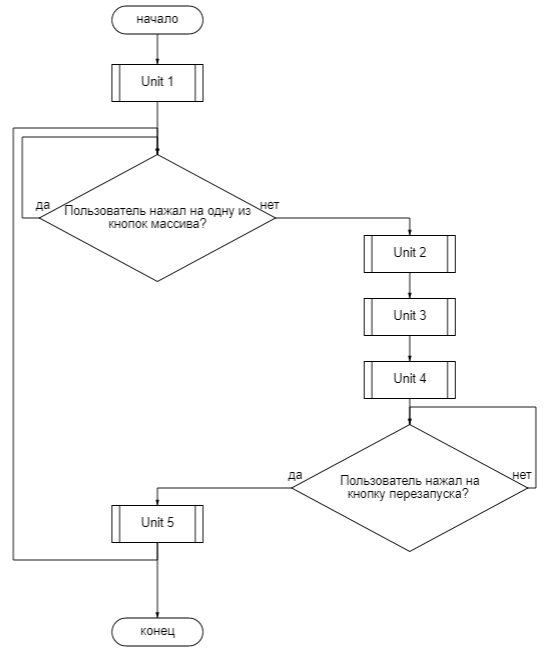


Рис. 4 – Блок-схема работы программы

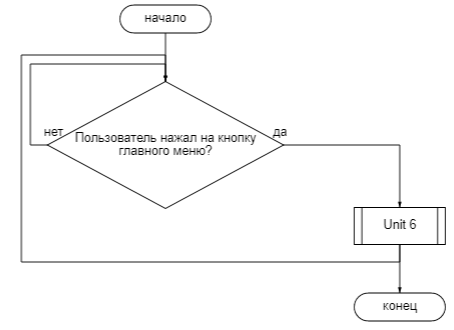


Рис. 5 – Блок-схема работы главного меню

# ЛИСТИНГ ПРОГРАММЫ

from tkinter import \*

from tkinter import messagebox

root = Tk()

root.title('Курсовая работа Богданов М. 19-ИЭ-1')

work = True

arr = []

count = 0

turn = 0

#Функция главного меню с информацией об авторе работы

def autor():

messagebox.showinfo(title="Автор", message="Работу выполнил студент\n2 курса группы 19-ИЭ-1\nБогданов Максим Дмитриевич")

#Функция перезапуска игры

def restart():

global count

count = 0

global work

work = True

for row in range(40):

for col in range(40):

arr[row][col]['text'] = ' '

arr[row][col]['background'] = 'lavender'

#Функция, вызываемая нажатием на кнопку, определяет текущий ход и запускает проверку победы

def click(row, col):

global turn, count

if count % 2 == 0:

turn = 0

else: turn = 1

if turn == 1:

if work and arr[row][col]['text'] == ' ':

arr[row][col]['text'] = 'X'

count += 1

win\_check('X')

if turn == 0:

if work and arr[row][col]['text'] == ' ':

arr[row][col]['text'] = 'O'

count += 1

win\_check('O')

#Проверяется каждое возможное выигрышное состояние из 5 клеток

def win\_check(symbol):

for i in range(36):

for n in range(40):

line\_check(arr[n][i], arr[n][i+1], arr[n][i+2], arr[n][i+3], arr[n][i+4], symbol) #проверка горизонтали

line\_check(arr[i][n], arr[i+1][n], arr[i+2][n], arr[i+3][n], arr[i+4][n], symbol) #проверка вертикали

for i in range(36):

for n in range(36):

line\_check(arr[n][i], arr[n+1][i+1], arr[n+2][i+2], arr[n+3][i+3], arr[n+4][i+4], symbol) #проверка косой верх лево

line\_check(arr[i][n+4], arr[i+1][n+3], arr[i+2][n+2], arr[i+3][n+1], arr[i+4][n], symbol) #проверка косой верх право

#Проверяется одинаковость символов в линии и в этом случае выделяет выигрышную комбинацию на поле

def line\_check(a1,a2,a3,a4,a5,symbol):

if a1['text'] == symbol and a2['text'] == symbol and a3['text'] == symbol and a4['text'] == symbol and a5['text'] == symbol:

a1['background'] = a2['background'] = a3['background'] = a4['background'] = a5['background'] = 'pink'

global work

work = False

#Построение поля 40 на 40

for row in range(40):

line = []

for col in range(40):

button = Button(root, text=' ', width=2, height=1,

font=('Verdana 8 bold'), background='lavender',

command=lambda row=row, col=col: click(row,col))

button.grid(row=row, column=col, sticky='nsew')

line.append(button)

arr.append(line)

newButton = Button(root, text='Новая игра', command=restart)

newButton.grid(row=40, column=0, columnspan=40, sticky='nsew')

mainMenu = Menu( root )

root.config(menu = mainMenu)

mainMenu.add\_command(label = "Сведения об авторе", command = autor)

root.mainloop()

# ВИД РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Пустое поле

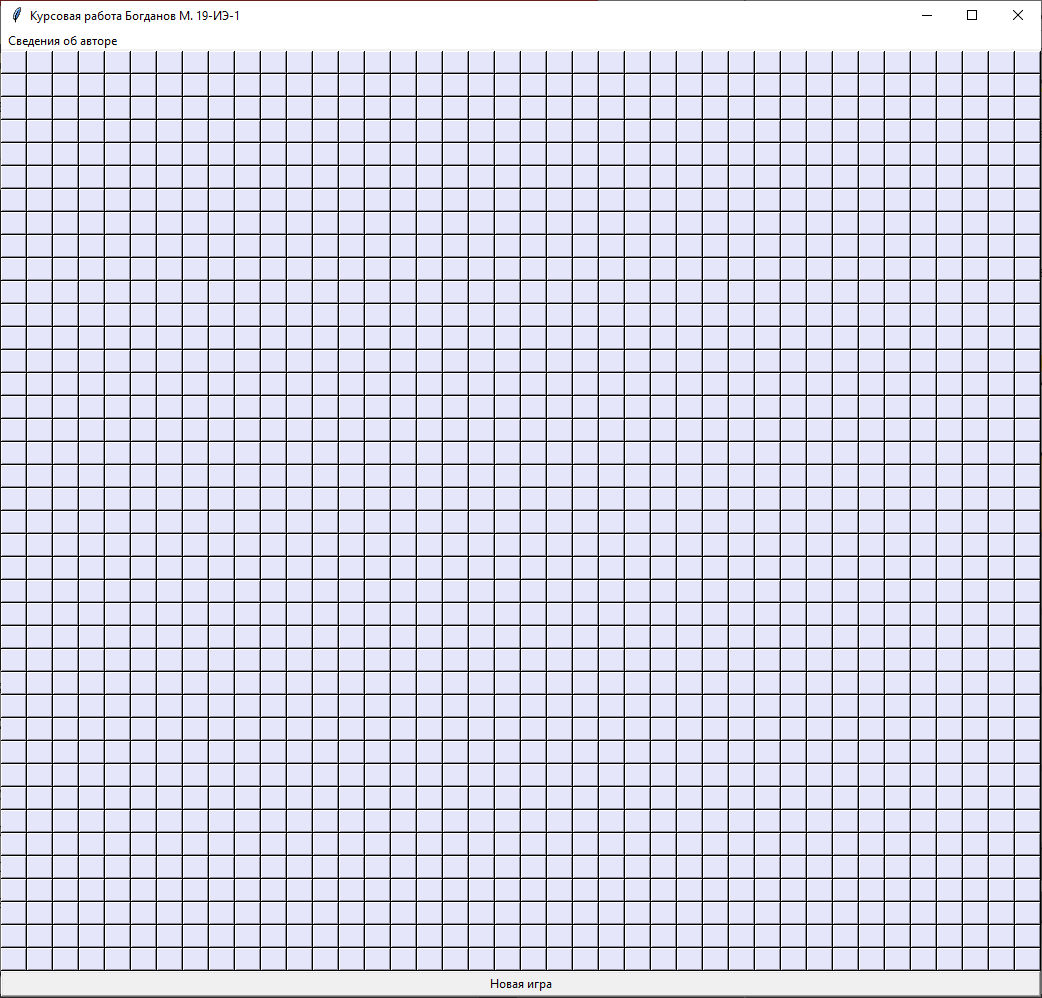


Рис. 1 – Пустое поле

Результат игры

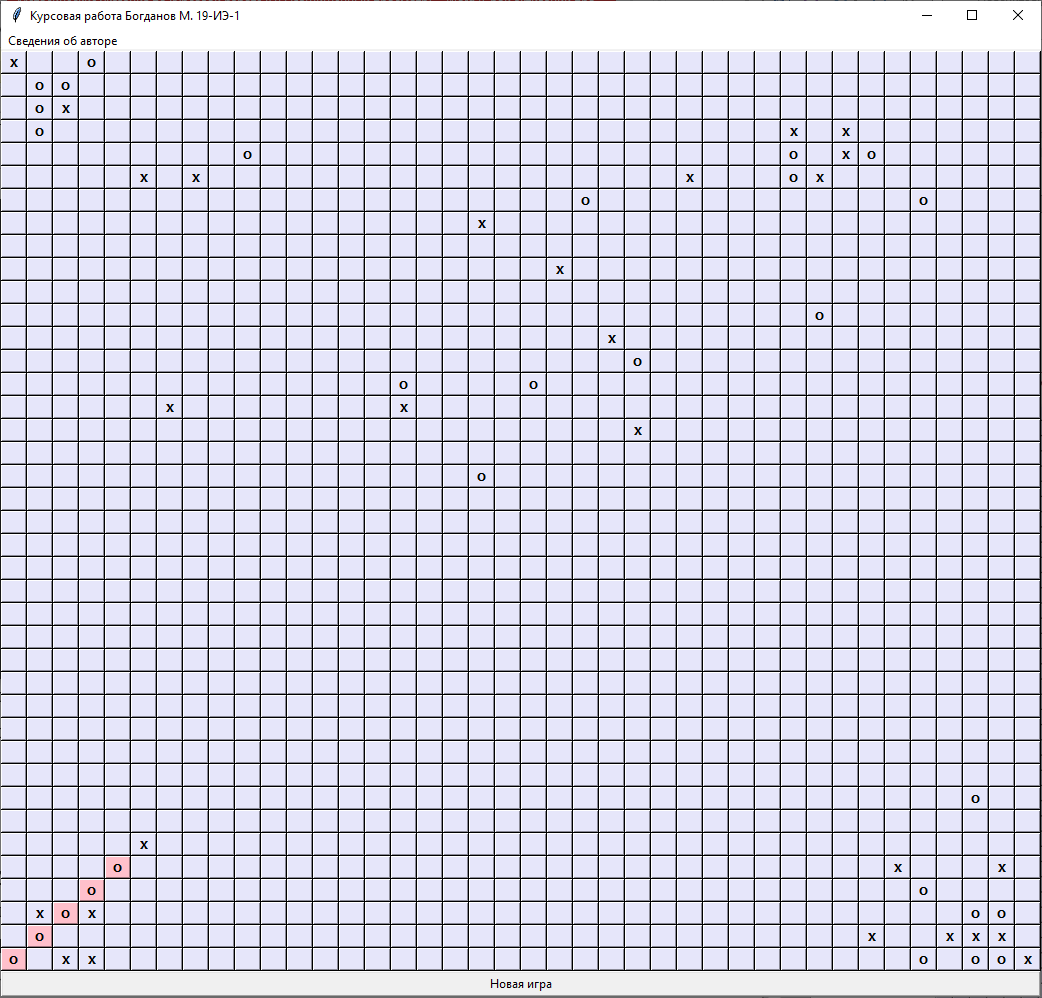


Рис. 2 – Результат игры

Окно с информацией об авторе

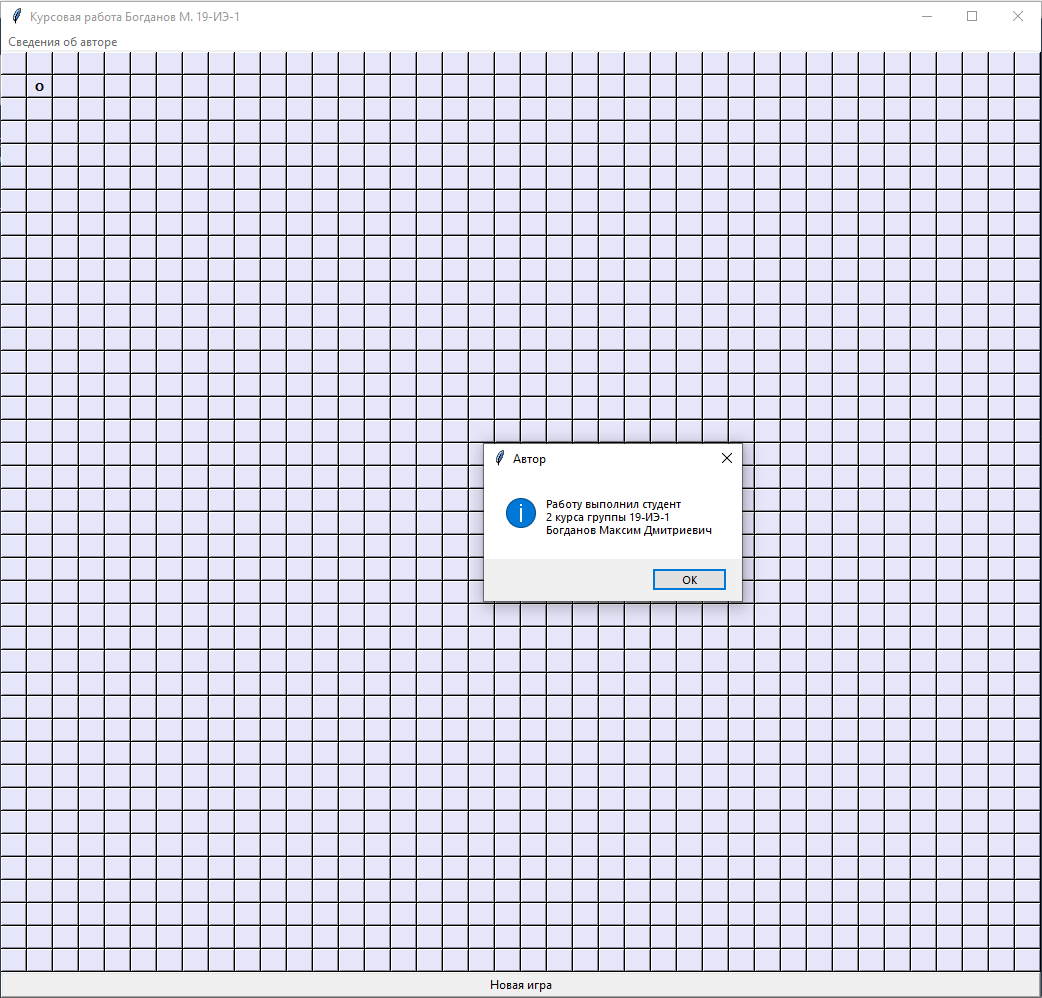


Рис. 3 – Окно с информацией об авторе

# ЗАКЛЮЧЕНИЕ

В результате проделанной работы была достигнута поставленная цель, а именно, разработана игра “крестики-нолики”.

Теоретическая часть помогла избежать лишних ошибок и сыграла немалую роль в понимании основных процессов и структур

В ходе работы мною были укреплены знания и умения в написании программ на языке Python с использованием библиотеки создания графических интерфейсов Tkinter. Были задействованы такие умения как:

1. Создание и обработка объёмного массива данных
2. Работа с модулем messagebox из библиотеки Tkinter
3. Обработка событий
4. Использование возможностей главного меню для вывода требуемой информации

Эти умения будут полезны при последующем написании последующих курсовых, практических лабораторных, а также дипломной работы.

Главное достоинство Python – простота, в которой я убедился, достигая цели – создавая программу, и о которой говорится в послании разработчиков – философии Python. В большей степени к упрощению написания программ приводит отсутствие необходимости объявлять переменные, тип которых определяется по ходу написания кода автоматически, а также удобная система отступов, что придает коду компактность и структурированность.

Безусловно, как и любой язык программирования, Python имеет некоторые свои недостатки. Но преимущества, которые он предоставляет при создании программного обеспечения, гораздо существеннее и не оказывают особого влияния на его функциональность и практичность. В случае же критичности некоторых недостатков при решении некоторых специфических задач, универсальность и гибкость Python позволяет обходить эти недостатки без ущерба для решаемой задачи. Таким образом, можно с большой уверенностью утверждать, что Рython подходит для решения подавляющего большинства повседневных задач, будь то подключение к сети интернет, чтение-отправка электронной почты, резервное копирование, либо же какая-нибудь игрушка. Язык программирования Python практически не имеет никаких ограничений или запретов на использование, поэтому также может свободно использоваться при создании крупных проектов. К примеру, Рython интенсивно применяется многими IT-гигантами, такими как, например, Google и Yandex. А универсальность и простота и Рython делают его одним из лидеров среди языков программирования как для профессионалов, так и для тех, кто только начинает пробовать свои силы в программировании.

Подводя итоги, можно сделать вывод, что Python вполне может подойти для изучения как начинающим программистам, так разработчикам с большим опытом.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://habr.com/ru/post/133337/>
2. <http://ilnurgi1.ru/docs/python/modules/tkinter/widget.html>
3. <http://ilnurgi1.ru/docs/python/modules/tkinter/canvas.html>
4. <https://ru.stackoverflow.com/questions/478561/>
5. <http://python-3.ru/page/pochemu-programmisty-ispolzujut-python>