Муниципальное бюджетное общеобразовательное учреждение
“Кадетская средняя общеобразовательная школа имени Героя Российской Федерации В.И. Шарпатова”

Индивидуальный итоговый проект «Создание приложения «COLORS» на языке программирования Python»

учитель информатики
Шарапов Тимур Фархатович

Новый Уренгой, 2022

Оглавление

Введение 3
Знакомство со средой Python 4
Виджеты, используемые в программе 6
Геометрия 8
Проектирование приложения в среде программирования Python 10
Заключение 12
Список используемой литературы 13
Приложение 1 14

Введение

В нашем обществе компьютер играет очень важную роль, уже трудно представить труды ученых, математиков, бухгалтеров, экономистов без использования вычислительной техники. Но компьютер сам не способен даже на простые операции, поэтому для того чтобы люди могли его использовать, необходимо программное обеспечение.

Программа управляет ресурсами компьютера и предоставляет их в распоряжение пользователя. Существует большое количество разнообразных по своим функциям и возможностям программ, но среди них выделяют два основных типа: системное и прикладное программное обеспечение. К системному программному обеспечению относят операционную систему и все программы, предназначенные для создания рабочей среды для прикладных программ, которые в свою очередь выполняют ту самую, необходимую пользователю, работу.

Научная новизна моей работы:

- выявление условий перевода программы на машинный язык в среде программирования Python;
- создание программы «Colors» в среде программирования Python.

Цель – разработка программы «Colors» для уроков изобразительного искусства с применением способов создания программ в среде Python.

Задачи исследования:

- изучить основы среды Python;
- познакомиться с основными компонентами программы;
- познакомиться с технологией разработки программного обеспечения на всех этапа;
- изучить терминологию программиста.

Объект исследования – среда программирования Python.

Предмет исследования – программное обеспечение уроков информатики.

Методы:

- эмпирические: создание опытным путем приложения «Colors» в среде программирования Python;

• теоретические: изучение книг и статей по языку программирования Python, изучение основных этапов при создании программы и терминологии программиста с целью создания программы «Colors».

В качестве ***гипотезы исследования*** было выдвинуто предположение о том, что на уроках изобразительного искусства при изучении темы «Смешение красок» можно наглядно показать какой цвет может получиться, если смешать два цвета, а также использование визуальной объектно-ориентированной системы позволяет облегчить знакомство с современными методами создания программ.

Знакомство со средой Python

Python – это высокоуровневый, динамичный, объектно-ориентированный язык программирования. Он ориентирован на повышение производительности программиста и читаемости кода. Разработчиком кода является Гвидо ван Россум. Впервые язык увидел свет в 1991 году. Для создания Python автор вдохновлялся такими языками программирования как ABC, Haskell, Java, Lisp, Icon и Perl. Python является высокоуровневым, кроссплатформенным, но в то же время минималистичным языком. Одним из его основных преимуществ является отсутствие скобок и точек с запятой. Вместо этого Python использует отступы. Сегодня существует две основные ветви языка: Python 2.x и Python 3.x.

Сегодня Python поддерживается большим количеством добровольцев со всего мира. Язык имеет открытый исходный код. Python – это идеальный язык для тех людей, которые хотят научиться программировать.

Язык программирования Python поддерживает несколько стилей программирования. Он не принуждает разработчика придерживаться определенной парадигмы. Python поддерживает объектно-ориентированное и процедурное программирование.

Строить графический интерфейс пользователя (GUI, Graphical User Interface) для программ на языке Python можно при помощи

соответствующих библиотек компонентов графического интерфейса. Самым распространённой пакетом является Tkinter.

Tkinter – это графическая библиотека на основе Tk, которая входит в стандартную библиотеку Python. Tk является библиотекой базовых элементов графического интерфейса для языка Tcl. Tkinter реализован как оболочка Python для Tcl интерпретатора, встроенного в интерпретатор Python. Существует и несколько других популярных графических инструментов для Python. Наиболее популярными являются wxPython, PyQt и PyGTK.

Почти все современные графические интерфейсы общего назначения строятся по модели WIMP - Window, Icon, Menu, Pointer (окно, иконка, меню, указатель). Внутри окон рисуются элементы графического интерфейса, которые для краткости называются виджетами (widget - штучка). Меню могут располагаться в различных частях окна, но их поведение достаточно однотипно: они служат для выбора действия из набора предопределенных действий. Пользователь графического интерфейса "объясняет" компьютерной программе требуемые действия с помощью указателя. Обычно указателем служит курсор мыши или джойстика, однако есть и другие "указательные" устройства. С помощью иконок графический интерфейс приобретает независимость от языка и в некоторых случаях позволяет быстрее ориентироваться в интерфейсе.

Основная черта любой программы с графическим интерфейсом – интерактивность. Программа не просто что-то считает (в пакетном режиме) от начала своего запуска до конца: ее действия зависят от вмешательства пользователя. Фактически, графическое приложение выполняет бесконечный цикл обработки событий. Программа, реализующая графический интерфейс, событийно-ориентирована. Она ждет от интерфейса событий, которые и обрабатывает сообразно своему внутреннему состоянию.

Эти события возникают в элементах графического интерфейса (виджетах) и обрабатываются прикрепленными к этим виджетам

обработчиками. Сами виджеты имеют многочисленные свойства (цвет, размер, расположение), выстраиваются в иерархию принадлежности (один виджет может быть хозяином другого), имеют методы для доступа к своему состоянию.

Разработка настольных приложений с помощью python Tkinter – не сложная задача. Пустое окно верхнего уровня Tkinter можно создать, выполнив следующие действия:

- Импортировать модуль Tkinter.
- Создайте главное окно приложения.
- Добавьте в окно виджеты, такие как метки, кнопки, рамки и т. д.
- Вызовите основной цикл событий, чтобы действия могли отображаться на экране компьютера пользователя.

Виджеты, используемые в программе

Существуют различные виджеты, такие как button, canvas, checkbutton, entry и т. д., которые используются для создания приложений с графическим интерфейсом пользователя Python.

№	Виджеты	Описание
1	Button	Виджет кнопки используется для добавления различных типов кнопок в приложение Python.
2	Canvas	Этот виджет используется для рисования на холсте в окне.
3	Checkbutton	Используется для отображения кнопки-флажка в окне
4	Entry	Виджет ввода используется для отображения однострочного текстового поля. Обычно он используется для принятия пользовательских значений.
5	Frame	Этот виджет можно определить как контейнер, в который можно добавить и организовать другой виджет.
6	Label	Метка – это текст, используемый для отображения

		некоторого сообщения или информации о других виджетах.
7	Message	Виджет сообщения используется для отображения окна сообщения пользователю.
8	Text	Он отличается от Entry, поскольку предоставляет пользователю многострочное текстовое поле, чтобы пользователь мог писать текст и редактировать текст внутри него.

Виджет Tkinter Label в Python используется для указания метки контейнера, в котором мы можем разместить текст или изображения. Этот виджет используется для сообщения пользователю о других виджетах, используемых в приложении Python.

Существуют различные параметры, которые можно указать для настройки текста или части текста, отображаемого в метке. Синтаксис для использования метки приведен ниже.

```
1.  w = Label(master, options)
```

Виджет Python Tkinter Frame используется для организации группы виджетов. Он действует как контейнер, который можно использовать для хранения других виджетов. Прямоугольные области экрана используются для организации виджетов в приложении Python.

Синтаксис:

```
1.  w = Frame(parent,  options)
```

Tkinter Message виджет используется для отображения сообщения пользователю о действиях приложения python. Виджет сообщений показывает пользователю текстовые сообщения, которые нельзя редактировать. Текст сообщения содержит более одной строки. Однако сообщение может отображаться только одним шрифтом.

Синтаксис для использования виджета «Сообщение» приведен ниже.

```
1.  w = Message(parent, options)
```

Геометрия

Геометрия Tkinter определяет метод, с помощью которого виджеты отображаются на дисплее. Python Tkinter предоставляет следующие геометрические методы.

1) Метод Python Tkinter pack()

Метод pack() используется для организации виджета в блоке. Виджетами позиций, добавленными в приложение Python с помощью метода pack(), можно управлять с помощью различных параметров, указанных в вызове метода. Однако элементов управления меньше, а виджеты обычно добавляются менее организованно.

Синтаксис использования pack() приведен ниже:

```
1.  widget.pack(options)
```

Список возможных параметров, которые можно передать в pack(), приведен ниже.

expand: если для параметра «expand» задано значение «истина», виджет расширяется, заполняя любое пространство.

Fill: по умолчанию для заливки установлено значение NONE. Однако мы можем установить для него значение X или Y, чтобы определить, есть ли в виджете лишнее пространство.

size: представляет сторону родительского элемента, с которой виджет должен быть размещен в окне.

2) Метод Python Tkinter place()

Менеджер place() упорядочивает виджеты по определенным координатам x и y.

Синтаксис:

```
1.  widget.place(options)
```

Список необходимых вариантов для нашего приложения приведен ниже:

- height, width: это высота и ширина в пикселях.

•relx, rely: представлен в виде числа с плавающей запятой между 0,0 и 1,0, которое представляет собой смещение в горизонтальном и вертикальном направлениях.

•x, y: это относится к горизонтальному и вертикальному смещению в пикселях.

Проектирование приложения в среде программирования Python

Давайте попробуем научиться создавать графические приложения, т. е. оконные приложения для операционной системы Windows. Мы узнаем, как создаются и появляются на нашем экране монитора окошки Windows. В этом нам поможет разобраться специальный модуль tkinter. Наша цель научиться создавать окошки и надписи, кнопки и др., берем на вооружение язык программирования Python.

Основной задачей графического интерфейса является упрощение коммуникации между пользователем и компьютером. Об этом следует постоянно помнить при проектировании интерфейса. Применение имеющихся в наличии у программиста (или дизайнера) средств при создании графического интерфейса нужно свести до минимума, выбирая наиболее удобные пользователю виджеты в каждом конкретном случае.

Проведём нашу работу по шагам. Вначале создадим окно, указываем заголовок и размер окна.

```
from tkinter import*          // загружаем графическую библиотеку
root=Tk()                     // создаём окно root
root.title("COLORS")          // указываем заголовок окна
root.geometry('550x550')      // указываем размер окна
root.mainloop()               // запускаем первое окно
```

Следующим шагом мы создадим надписи(Label), поля ввода(Entry) и кнопку(Button) в нашем окне.

```
EntryA=Entry(root,width=10,font='Arial 16')
EntryA.pack()
labelw=Label(root,text='+').pack()
EntryB=Entry(root,width=10,font='Arial 16')
EntryB.pack()
labely=Label(root,text='=').pack()
but=Button(root,text='Смешать',command=red_label).pack()
root.mainloop()
```

Далее создается программный код функции, в котором записан алгоритм смешивания цветов. Работоспособность программы представлена на рис. 1, рис. 2 (приложение 1).

```
def red_label():
    r='красный';g='зеленый';y='желтый';w='белый';  bk='черный';bl='синий';
v='фиолетовый';o='оранжевый';p='розовый';br='коричневый';r1='red';g1='green'
y1='yellow';w1='white';bk1='black';bl1='blue';v1='violet';o1='orange';p1='pink';
br1='brown';
    a=EntryA.get()
    b=EntryB.get()
    if a==g and b==r or a==r and b==g:
        label1['bg']=g1
        label2['bg']=r1
        label['bg']=br1
   .
   .
   .
    elif a==" and b!=":
        label1['bg']='black'
        label2['bg']='black'
        label['bg']='black'
        messagebox.showwarning('Ошибка! Введите цвет в верхнее окошко')
    elif b==" and a!=":
        label['bg']='black'
        label1['bg']='black'
        label2['bg']='black'
        messagebox.showwarning('Ошибка! Введите цвет в нижнее окошко')
     elif a==" and b==" or b==" and a==":
        label['bg']='black'
        label1['bg']='black'
```

```
        label2['bg']='black'
        messagebox.showwarning('Ошибка! Введите цвет в оба окошка')
    else:
        label['bg']='white'
        label1['bg']='white'
        label2['bg']='white'
        messagebox.showwarning('Ошибка! Проверьте правильность написания, либо пока данного цвета нет в нашей палитре')
```

Заключение

Я считаю, что на современном этапе информатизации общества программирование на разных языках программирования остается очень актуальным.

Цель работы достигнута в полном объеме. Результатом нашей работы стало готовое к работе приложение.

В ходе проведенной работы нами были изучены основы языка Python, мы познакомились с технологией разработки программного обеспечения на всех этапах: от постановки задачи, выбора алгоритма, его реализации до сопровождения программы и разработали приложение. Оценивая проделанную работу, мы научились.

1. Создавать оконные приложения
2. Задавать различные свойства окнам (цвет, размер)
3. Создавать надписи и кнопки
4. Разрабатывать программный код

В результате проекта составлена программа «Colors», которая демонстрируется в качестве ознакомления со средой программирования Python, а также на уроках изобразительного искусства при изучении темы «Смешивания красок» в начальной школе. Так же возможно обновление и добавление новых функций

Выдвинутая в начале исследования гипотеза полностью подтверждена.

Список используемой литературы

1. Горбачев Л.И. «Программирование приложений в среде Python 3.7», 2012.

2. Керман, Митчелл «Программирование и отладка в Python». Учебный курс.: Пер. с англ.- М.: Издательский дом «Вильямс», 2012, 672с.:ил.

3. Фараонов В.В. Python 3.7. Учебный курс.- М.: Издатель Молгачева С.В., 2012.-672с., ил.

4. Учебные материалы URL: https://pythonpip.ru/examples/kak-napisat-kod-dlya-pechati-ierarhii-isklyucheniy-i-oshibok-v-python

Приложение 1

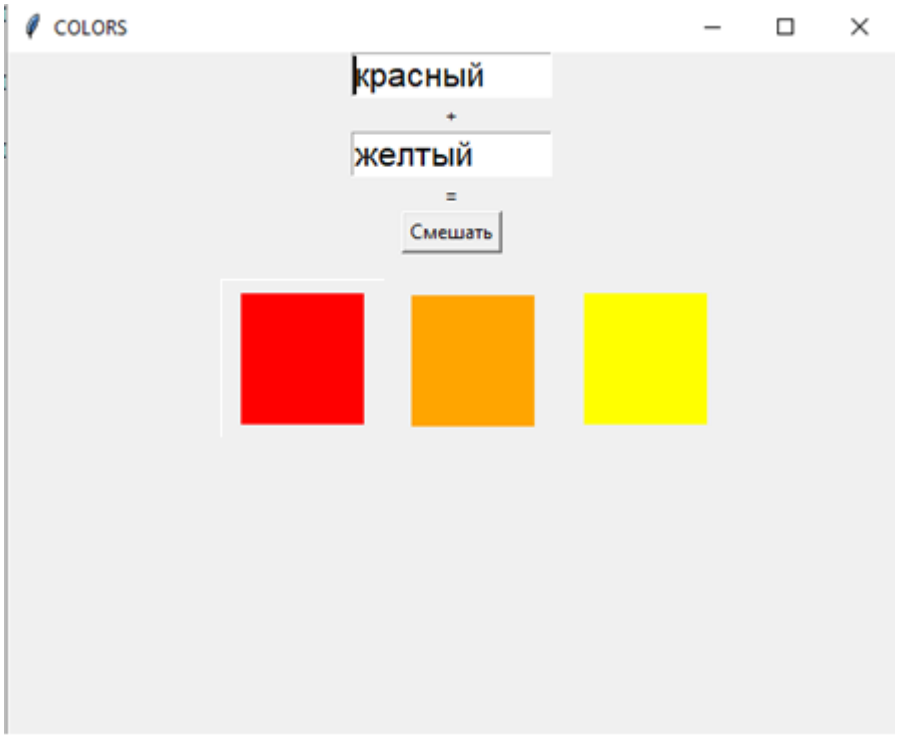


Рис. 1

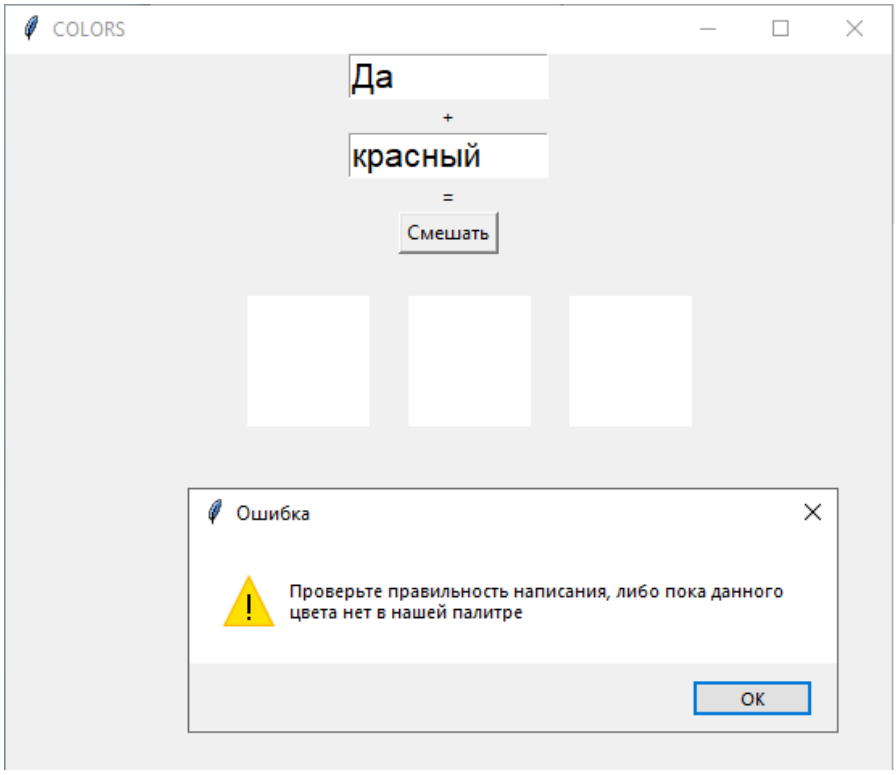


Рис. 2