TRƯỜNG ĐẠI HỌC

SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

HCMC University of Technology and Education



KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

NGÀNH CÔNG NGHỆ THÔNG TIN

****

BÁO CÁO ĐỒ ÁN I

4 BÀI TẬP LÀM TRÊN THƯ VIỆN PYTHON TKINTER

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện: | |
| Nguyễn Văn Thông | 17151034 |
| Vũ Ngọc Bội | 17110102 |

GVHD: TS. Trần Tiến Đức

Tp. Hồ Chí Minh, tháng 12 – 2019

ĐIỂM SỐ

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(*ký và ghi họ tên*)

Trần Tiến Đức

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Trần Tiến Đức, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và ngành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

TP.HCM, ngày 10 tháng 12 năm 2019

Nhóm sinh viên thực hiện

MỤC LỤC

[Chương 1: Tổng quan chương trình 1](#_Toc27031092)

[1. Giới thiệu chung 1](#_Toc27031093)

[1.1. Về đồ án 4 bài tập python tkinter 1](#_Toc27031094)

[1.1.1. Yêu cầu đồ án 1](#_Toc27031095)

[1.1.2. Phương hướng thực hiện 1](#_Toc27031096)

[1.2. 4 bài tập Python tkinter 1](#_Toc27031097)

[1.2.1. Giới thiệu về tkinter 1](#_Toc27031098)

[1.2.2. Giới thiệu về ngôn ngữ Python 1](#_Toc27031099)

[2. Đặc tả phần mềm 4 bài Python tkinter 2](#_Toc27031100)

[2.1. 4 bài python tkinter 2](#_Toc27031101)

[2.1.1. Giới thiệu 2](#_Toc27031102)

[2.1.2. Tính năng chính 2](#_Toc27031103)

[2.1.3. Ứng dụng 2](#_Toc27031104)

[2.2. Yêu cầu kĩ thuật 2](#_Toc27031105)

[2.3. Công cụ và công nghệ sử dụng 2](#_Toc27031106)

[Chương 2: Kế hoạch thực hiện 3](#_Toc27031107)

[1. Kế hoạch 3](#_Toc27031108)

[2. Phân công công việc 3](#_Toc27031109)

[Chương 3: Thiết kế phần mềm 4](#_Toc27031110)

[1. Thiết kế giao diện 4](#_Toc27031111)

[1.1. Giao diện chương trình 4](#_Toc27031112)

[2. Code 5](#_Toc27031113)

[Chương 4: Kết luận và hướng phát triển 11](#_Toc27031114)

[1. Kết luận 11](#_Toc27031115)

[2. Hướng phát triển 11](#_Toc27031116)

[Tài liệu tham khảo 12](#_Toc27031117)

# Chương 1: Tổng quan chương trình

## Giới thiệu chung

### Về đồ án 4 bài tập python tkinter

#### Yêu cầu đồ án

Thiết kế và xây dựng phần mềm hướng đối tượng giải quyết yêu cầu thực hiện bằng Python với thư viện là tkinter .

Phân tích đồ án

* Xây dựng phần mềm hướng đối tượng.
* Phân tích các chức năng .

#### Phương hướng thực hiện

* Xây dựng phần mềm hướng đối tượng bằng Python
* Ứng dụng thư viện Tkinter để thiết kế giao diện người dùng.

### 4 bài tập Python tkinter

#### Giới thiệu về tkinter

Tkinter *là một gói trong Python có chứa module* Tk*hỗ trợ cho việc lập trình GUI*. Tk *ban đầu được viết cho ngôn ngữ****Tcl***. *Sau đó Tkinter được viết ra để sử dụng Tk bằng trình thông dịch Tcl trên nền Python. Ngoài Tkinter ra còn có một số công cụ khác giúp tạo một ứng dụng GUI viết bằng Python như wxPython, PyQt, và PyGTK.*

#### Giới thiệu về ngôn ngữ Python

Python là một ngôn ngữ lập trình hướng đối tượng, python hoàn toàn tạo kiểu động, cấp phát bộ nhớ động. Mục đích ra đời của Python là cung cấp một ngôn ngữ lập trình có cấu trúc rõ ràng, sáng sủa, thuận tiện cho người mới học lập trình. Python được phát triển bởi Guido và Rossum. Phiên bản đầu tiên được phát hành vào năm 1991. Python được lấy cảm hứng từ ABC, Haskell, Java, Lisp, Icon và Perl. Python là một ngôn ngữ thông dịch, đa nền tảng. Một trong những đặc điểm độc nhất của Python là ngôn ngữ này không dùng đến dấu chấm phẩy, dấu mở-đóng ngoặc {} để kết thúc câu lệnh hay khối lệnh, mà cách duy nhất để nó nhận biết một lệnh là dấu thụt đầu dòng.

Hiện tại Python có hai dòng phiên bản là dòng 2.x và 3.x. Phiên bản Python mà series này sử dụng là phiên bản **3.5.1**. Ngày nay Python được phát triển bởi một cộng đồng tình nguyện trên khắp thế giới, và lẽ dĩ nhiên nó là một phần mềm mã nguồn mở.

## Đặc tả phần mềm 4 bài Python tkinter

### 4 bài python tkinter

#### Giới thiệu

Tkinter là một gói trong ngôn ngữ lập trình Python có chứa module Tk hỗ trợ cho việc lập trình GUI.

#### Tính năng chính

* Tạo giao diện trên nền Python tkinter.
* Thể hiện các chức năng như đã yêu cầu .

#### Ứng dụng

Giúp người dùng có cái nhìn trực quan về tkinter viết bằng ngôn ngữ tkinter .Tkinter không cần cài đặt phức tạp và sử dụng khá dễ dàng.

### Yêu cầu kĩ thuật

* Thực hiện được yêu cầu mà đồ án đề ra.
* Áp dụng lập trình hướng đối tượng.
* Dung lượng phần mềm nhẹ, chạy ổn định.

### Công cụ và công nghệ sử dụng

* Viết bằng ngôn ngữ tkinter với thư viện tkinter.
* Thiết kế giao diện người dùng (GUI) bằng Visual Studio 2019.

# Chương 2: Kế hoạch thực hiện

## Kế hoạch

|  |  |
| --- | --- |
| Tuần | Công việc |
| 3 | Tìm hiểu về các thuộc tính và chức năng của python tkinter |
| 7 | Bắt đầu thiết kế giao diện |
| 8 | Xây dựng và cài chức năng cho từng giao diện |
| 10 | Tiếp tục xây dựng, cài đặt chức năng và hoàn thành phần mềm. |
| 11 | Chạy từng giao diện và nếu có lỗi không phù hợp điều chỉnh |

## Phân công công việc

Bảng 2: Phân công công việc & đóp góp của mỗi sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên sinh viên | Miêu tả công việc | Đóng góp |
| 1 | Vũ Ngọc Bội | * Thiết kế giao diện các form của các bài * Thiết kế chức năng như đã yêu cầu | 60% |
| 2 | Nguyễn Văn Thông | * Thiết kế giao diện phần mềm và báo cáo * Báo cáo * Kiểm tra lỗi của mỗi bài (nếu có) | 40% |

# Chương 3: Thiết kế phần mềm

## Thiết kế giao diện

### Giao diện chương trình

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | Cho phép người dùng thao tác chuột click vào từng ô .Click đầu sẽ hiện lên hình X .Click thứ hai sẽ mất hình X. |  |
| 2 |  | Thiết kế form có chức năng khi click chuột vào thì hình ảnh và màu sắc sẽ thay đổi theo. |  |
| 3 |  | Thiết kế đồng hồ thời gian thật với giao diện số 7 đoạn. |  |
| 4 |  | Thiết kế form có chức năng khi click chuột vào nút start sẽ làm cho hình vuông chạy trên hình tròn, và khi click chuột vào nút stop sẽ làm cho hình vuông đó dừng lại. |  |

1. Code

Bài 1.

|  |  |
| --- | --- |
| class FieldUI(tk.Canvas):  def \_\_init\_\_(self, parent, sizew, sizeh, variable):  tk.Canvas.\_\_init\_\_(self, parent, width=sizew, height=sizeh)  self.sizew = sizew  self.sizeh = sizeh  self.variable = variable  self.variable.trace('w', self.update\_display)  self.update\_display()  def update\_display(self, \*\_args):  self.delete(tk.ALL)  value = self.variable.get()  if value == 'x':  self.create\_line(0, 0, self.sizew, self.sizeh, width=1)  self.create\_line(self.sizew, 0, 0, self.sizeh, width=1)  elif value == 'y':  self.delete() | Tạo giao cho chường trình |
| class Board(tk.Frame):  def \_\_init\_\_(self, parent = None):  tk.Frame.\_\_init\_\_(self, parent, background='black') ## call board  self.field\_vars = [tk.StringVar() for \_ in range(12)] ## size board  for i, field\_var in enumerate(self.field\_vars):  field\_ui = FieldUI(self, 238, 138, field\_var) ##size board mini  field\_ui.bind('<Button-1>', partial(self.on\_click1, i))  field\_ui.bind('<Button-3>', partial(self.on\_click2, i))  column, row = divmod(i, 3) ## number row  field\_ui.grid(row=row, column=column, padx=0, pady=0) ##line bettwen row and column  field\_ui.configure(highlightthickness=1, highlightbackground="black")  def on\_click1(self, field\_index, \_event=None):  field\_var = self.field\_vars[field\_index]  field\_var.set('x')  def on\_click2(self, field\_index, \_event=None):  field\_var = self.field\_vars[field\_index]  field\_var.set('y')  def deleteline(self,field\_ui, event):  field\_ui.delete() | Tạo sự kiện cho thao tác click chuột. |

Bài 2

|  |  |
| --- | --- |
| def \_\_init\_\_(self):  window = Tk()  window.title("Ve Hinh")  self.canvas = Canvas(window, width = 530, height = 300, bg = "whitesmoke")  self.canvas.pack()  frame = Frame(window)  frame.pack()  #Label  labelframe1 = LabelFrame(window, text="Figure")  labelframe1.place(x=410,y=0,height=150,width=120)  labelframe2= LabelFrame(window, text="Color")  labelframe2.place(x=410,y=150,height=150,width=120)  #radio button  self.figure = IntVar() | Tạo giao diện cho chương trình. |
| #Set Checkbox = true Startup  self.figure.set(1)  Radiobutton(labelframe1, text = "Elip",variable = self.figure, value = "1", command = self.Change).place(x = 10, y = 10)  Radiobutton(labelframe1, text = "Chu Nhat",variable = self.figure, value = "2", command = self.Change).place(x = 10, y = 40)  Radiobutton(labelframe1, text = "Tam Giac",variable = self.figure, value = "3", command = self.Change).place(x = 10, y = 70)  #check button  self.var1 = IntVar()  self.var2 = IntVar()  self.var3 = IntVar()  #Set Checkbox = true Startup  self.var1.set(1)  Checkbutton(labelframe2, text = "Red", variable = self.var1, command = self.Change).place(x = 10, y = 10)  Checkbutton(labelframe2, text = "Green", variable = self.var2, command = self.Change).place(x = 10, y = 40)  Checkbutton(labelframe2, text = "Blue", variable = self.var3, command = self.Change).place(x = 10, y = 70) | Tạo các nút nhấn và sự kiện cho nó. |
| window.mainloop()  ## Value Get  def Value(self):  self.on = str(self.figure.get())  self.red = str(self.var1.get())  self.green = str(self.var2.get())  self.blue = str(self.var3.get()) | Cài đặt giá trị. |
| ##Change  def Change(self):  self.clearCanvas()  self.draw\_Blackcolor()  ##Delete Canvas Before Draw  def clearCanvas(self):  self.canvas.delete('elip','chunhat','tamgiac') | Hàm dùng để xóa Canvas |
| ##Color RBG TransLates  def colorRBG(self):  def \_from\_rgb(rgb):  return "#%02x%02x%02x" % rgb  self.Value()  if self.red == "1":  self.x = 255  else:  self.x = 0  if self.green == "1":  self.y = 255  else:  self.y = 0  if self.blue == "1":  self.z = 255  else:  self.z = 0  return \_from\_rgb((self.x, self.y, self.z)) | Cài đặt màu sắc khi chọn nhiều màu. |
| ##Color Change  def draw\_Blackcolor(self):  self.Value()  abc = self.colorRBG()  if self.on == "1":  self.canvas.create\_oval(20,300,280,30, fill = abc, tags="elip")  elif self.on == "2":  self.canvas.create\_rectangle(20,200,280,30, fill = abc,tags="chunhat")  elif self.on == "3":  self.canvas.create\_polygon(150,65,245,240,45,240,150,65, fill=abc, outline ="black",tags="tamgiac") | Thay đổi màu sắc cho các vật thể |

Bài 3

|  |  |
| --- | --- |
| import tkinter as tk  from time import strftime  root = tk.Tk()  root.wm\_title("Clock Time")  root.geometry("1200x300+200+150")  root.resizable(width= False, height= False)  screen = tk.Canvas(root)  screen.config(width = 1200, height = 300)  screen.grid() | Tạo giao diện cho chương trình. |
| ##setting  offsets =(  (7, 6, 11, 2, 31, 2, 35, 6, 31, 10, 11, 10),  (6, 7, 10, 11, 10, 31, 6, 35, 2, 31, 2, 11),  (36, 7, 40, 11, 40, 31, 36, 35, 32, 31, 32, 11),  (7, 36, 11, 32, 31, 32, 35, 36, 31, 40, 11, 40 ),  (6, 37, 10, 41, 10, 61, 6, 65, 2, 61, 2, 41),  (36, 37, 40, 41, 40, 61, 36, 65, 32, 61, 32, 41),  (7, 66, 11, 62, 31, 62, 35, 66, 31, 70, 11, 70)  )  # Segments used for each digit; 0, 1 = off, on.  digits = (  (1, 1, 1, 0, 1, 1, 1), # 0  (0, 0, 1, 0, 0, 1, 0), # 1  (1, 0, 1, 1, 1, 0, 1), # 2  (1, 0, 1, 1, 0, 1, 1), # 3  (0, 1, 1, 1, 0, 1, 0), # 4  (1, 1, 0, 1, 0, 1, 1), # 5  (1, 1, 0, 1, 1, 1, 1), # 6  (1, 0, 1, 0, 0, 1, 0), # 7  (1, 1, 1, 1, 1, 1, 1), # 8  (1, 1, 1, 1, 0, 1, 1), # 9  )  #Position X, Y  ptColon1 = (462,85, 477, 100, 462, 115, 447, 100)  ptColon2 = (462,175,477,190,462,205,447,190)  ptColon3 = (812,85,827,100,812,115,797,100)  ptColon4 = (812,175,827,190,812,205,797,190) | Tạo hình cho các số và vị trí |
| class Digit:  def \_\_init\_\_(self, canvas, x=10, y=40, length=3):  self.canvas = canvas  l = length  self.segs = []  for x0, y0, x1, y1, x2, y2, x3, y3, x4, y4, x5, y5 in offsets:  self.segs.append(canvas.create\_polygon(x + x0\*l, y + y0\*l, x + x1\*l, y + y1\*l,x + x2\*l, y + y2\*l, x + x3\*l, y + y3\*l, x + x4\*l, y + y4\*l, x + x5\*l, y + y5\*l, state='hidden', fill ="red"))  screen.create\_polygon(ptColon1, fill = "red")  screen.create\_polygon(ptColon2, fill = "red")  screen.create\_polygon(ptColon3, fill = "red")  screen.create\_polygon(ptColon4, fill = "red")  def show(self, num):  for iid, on in zip(self.segs, digits[num]):  self.canvas.itemconfigure(iid, state = 'normal' if on else 'hidden')  digh1 = Digit(screen, 150, 35) ##  digh2 = Digit(screen, 300, 35) ##  digm1 = Digit(screen, 500, 35) ##  digm2 = Digit(screen, 650, 35) ##  digs1 = Digit(screen, 850, 35) ##  digs2 = Digit(screen, 1000, 35) ## | Hiển thị các số lên trên giao diện |
| def update():  global s  X = int(strftime("%I%M%S"))  s2 = X%10  s1 = int(X/10) %10  m2 = int(X/100)%10  m1 = int(X/1000)%10  h2 = int(X/10000)%10  h1 = int(X/100000)%10  digs1.show(s1)  digs2.show(s2)  digm1.show(m1)  digm2.show(m2)  if h1 > 0:  digh1.show(h1)  digh2.show(h2)  root.after(1000, update)  root.after(1000, update) | Cập nhật thời gian. |

Bài 4

|  |  |
| --- | --- |
| class Celestial(object):  # Constants  COS\_0, COS\_180 = cos(0), cos(180)  SIN\_90, SIN\_270 = sin(90), sin(270)  def \_\_init\_\_(self, x, y, radius):  self.x, self.y = x, y  self.radius = radius  def bounds(self):  """ Return coords of rectangle surrounding circlular object. """  return (self.x + self.radius\*self.COS\_0, self.y + self.radius\*self.SIN\_270,  self.x + self.radius\*self.COS\_180, self.y + self.radius\*self.SIN\_90) | Định hình đường chuyển động |
| def circular\_path(x, y, radius, delta\_ang, start\_ang=0):  """ Endlessly generate coords of a circular path every delta angle degrees. """  ang = start\_ang % 360  while True:  yield x + radius\*cos(ang), y + radius\*sin(ang)  ang = (ang+delta\_ang) % 360 | Tạo vòng lập vô hạn cho chuyển động. |
| def start():  def update\_position(canvas, id, celestial\_obj, path\_iter):  celestial\_obj.x, celestial\_obj.y = next(path\_iter)  x0, y0, x1, y1 = canvas.coords(id)  oldx, oldy = (x0+x1) // 2, (y0+y1) // 2  dx, dy = celestial\_obj.x - oldx, celestial\_obj.y - oldy  canvas.move(id, dx, dy)  canvas.after(DELAY, update\_position, canvas, id, celestial\_obj, path\_iter)    top.after(DELAY, update\_position, canvas, planet1, planet\_obj1, path\_iter) | Tạo sự kiện để hình vuông bắt đầu di chuyển. |

# Chương 4: Kết luận và hướng phát triển

## Kết luận

Về cơ bản, nhóm tự nhận xét phần mềm của nhóm đã giải quyết được được 75% yêu cầu mà đồ án đặt đặt ra. Sau đây là ưu điểm cũng như tồn tại của phần mềm .

* Ưu điểm:
  + Giao diện gọn gàng, dễ tiếp cận, dễ làm quen.
  + Dung lượng khá nhẹ.
  + Chương trình tốn rất ít tài nguyên hệ thống khi hoạt động.
  + Chương trình chạy ổn định, cho ra kết quả chính xác, không bị crash trong quá trình thực thi yêu cầu người dùng.
* Nhược điểm:
  + Vẫn chưa hoàn thành hết chức năng như đã yêu cầu
  + Vẫn còn nhiều thiếu sót trong khâu chức năng của từng form

## Hướng phát triển

* Cải thiện giao diện người dùng.
* Chỉnh sửa form cho đẹp hơn và hoàn thiện hơn
* Thêm các chức năng phức tạp hơn

# Tài liệu tham khảo

[1]. Tài liệu về tkinter

Tkinter GUI Application Development Blueprints\_ First Editon (2015).pdf

Tkinter GUI application development blueprints-Second Edition (2018).pdf

Tkinter GUI application development cookbook-Packt Publishing (2018).pdf

[3]. Bắt đầu với python tkinter

<https://www.youtube.com/watch?v=RJB1Ek2Ko_Y&list=PL6gx4Cwl9DGBwibXFtPtflztSNPGuIB_d>

<https://www.youtube.com/watch?v=yQSEXcf6s2I&list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV>

<https://www.youtube.com/watch?v=yuoSKkSEhQg&list=PL6lxxT7IdTxGoHfouzEK-dFcwr_QClME_>

[5]. Tài liệu về python tkinter trên các trang uy tín

<https://www.tutorialspoint.com/python/python_gui_programming.htm>

<https://www.python-course.eu/python_tkinter.php>

<https://www.geeksforgeeks.org/python-gui-tkinter/>