ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Калининградский государственный технический университет»

(ФГБОУ ВО «КГТУ»)

Кафедра Систем управления и вычислительной техники

**Отчет защищен с оценкой \_\_\_\_\_\_\_\_**

**доцент, к.п.н Е.Ю. Заболотнова**

**28.12.2021**

**ОТЧЕТ**

по учебной практике

студента/ки группы 20 ВТ

\_\_\_\_\_Сафи А.А\_\_\_\_

Студент – практикант \_\_ Сафи.А.А\_\_\_\_\_

Руководитель практики к.п.н. доцент Заболотнова Е.Ю.

Калининград 2021

Федеральное агентство по рыболовству

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Калининградский государственный технический университет»

(ФГБОУ ВО «КГТУ»)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кафедра | | | Систем управления и вычислительной техники | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| УТВЕРЖДАЮ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Зав. кафедрой | | | | | |  | | | | | | / | Петрикин В.А. | | | | | | | | | | | | / | |
| « | 27 | » | декабря | | | | | 20 | | | | | 21 | | | г. | | |
| Индивидуальное задание | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| На учебную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | (вид, тип практики) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  | |
| студента | | Сафи Адам Абдель-Халим | | | | | | | | | | | | | | | | | | | | | | , | | | 20ВТ | | | | | | | |
| (Ф.И.О. полностью) | | | | | | | | | | | | | | | | | | | | | | | | | | | (группа) | | | | | | | |
| Направление подготовки (специальность) | | | | | 09.03.01 Информатика и вычислительная техника | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Место прохождения практики: | | | | | | | | ФГБОУ ВО КГТУ, кафедра Систем управления и вычислительной техники236022 Калининград, Советский проспект 1 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| За время прохождения практики: с | | | | | | | | | « | | 28 | » | | декабря | | | | | | | | 20 | | | | 21 | | | | г. | | | | |
| по | | | « | 27 | | » | | декабря | | | | | | | 20 | | | | 21 | | | | г. | | | | | |

студент должен выполнить следующие виды работ (заданий):

|  |  |  |
| --- | --- | --- |
| № п/п | **Содержание практики** (наименование работ/заданий) | **Рабочий график практики** |
| 1 | Получение индивидуальных заданий студентами | 29.06-01.07 |
| 2 | Выполнение индивидуального задания: разработка программы для тестирования по заданной теме на основе GUI на алгоритмическом языке Python | 02.07-10.07 |
| 3 | Подготовка отчета по учебной практике, защита практики | 10.07.21 - 12.07 |

**Планируемые результаты практики**

|  |  |
| --- | --- |
| **Компетенции выпускника ОП ВО и этапы их формирования** | **Знания, умения, навыки и опыт профессиональной деятельности** |
| ОПК–1: способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем | |
| ОПК-1.3: практическая профессионализация в процессе инсталлирования программного и аппаратного обеспечения. | Должен знать: - основные характеристики и параметры аппаратно-программных средств, учитываемые при их инсталляции.  Должен уметь: - находить нужные в данной ситуации стандарты, нормативы, методики, инструкции.  Должен владеть: - методиками инсталляции аппаратно-программных средств.  Должен приобрести опыт: - разрешения конфликтных ситуаций при инсталляции разных программных пакетов на одну операционную и/или аппаратную платформу. |
| ОПК-4: Способность участвовать в настройке и наладке программно-аппаратных комплексов | |
| ОПК-4.1: формирование практического навыка настройки и наладки программно-аппаратных комплексов для решения задач по профилю подготовки. | Должен знать: - основные характеристики и параметры инсталлируемых аппаратно-программных средств.  Должен уметь: - находить нужные в данной ситуации стандарты, нормативы, методики, инструкции.  Должен владеть: - методиками настройки – средств  Должен приобрести опыт: - локализации и устранения различных нештатных ситуаций в настройке и наладке программно-аппаратных комплексов |
| ПК-2: способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования | |
| ПК-2.5: формирование первичных профессиональных умений и навыков научно-исследовательской работы. | Должен знать: - структуру обзоров научной литературы и электронных информационных ресурсов, базовый инструментарий научных исследований  Должен уметь: - применять на практике методологию научных исследований.  владеть: - многообразными подходами к анализу научных результатов.  Должен приобрести опыт: - самостоятельного проведения и анализа научных исследований. |
| ПКД-1: способность готовить обзоры научной литературы и электронных информационно-образовательных ресурсов, используя отечественный и зарубежный опыт, для профессиональной деятельности | |
| ПКД-1.3: формирование первичных профессиональных умений и навыков подготовки обзоров научной литературы и электронных информационных ресурсов | Должен знать: - структуру обзоров научной литературы и электронных информационных ресурсов.  Должен уметь: - использовать известные механизмы подготовки обзоров научной литературы и электронных информационных ресурсов.  Должен владеть: - методиками и рекомендациями по подготовке обзоров научной литературы и электронных информационных ресурсов.  Должен приобрести опыт: - подготовки обзоров научной литературы и электронных информационных ресурсов |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель практики от университета | |  | | | |  | | доцент, к.п.н Заболотнова Е.Ю. | | | | | |
|  | | (подпись) | | | |  | | (Фамилия И.О., должность) | | | | | |
| Руководитель практики от профильной организации | |  | | | |  | | доцент, к.п.н Заболотнова Е. | | | | | |
|  | | (подпись) | | | |  | | (Фамилия И.О., должность) | | | | | |
| Практикант |  | |  |  | | | | | | | | | |
|  | (подпись) | |  | (телефон, E-mail) | | | | | | | | | |
|  |  | |  |  | | | | | | | | | |
| « | | 28 | | » | декабря | 20 | 21 | г. |

**Введение**

Целью учебной практики является закрепление и расширение теоретических знаний и практических навыков, получаемых студентами в ходе учебного процесса. После первого курса практика нацелена на освоение студентами основ программирования на алгоритмическом языке Python, закрепление навыков работы с файлами, разработку программ на основе графического интерфейса, методов и принципов объектно-ориентированного программирования.

**Постановка задачи**

Написать программу для прохождения тестирования по теоретическому материалу дисциплины «Программирование». Тема для тестирования: «Модуль graphics. Создание графических примитивов.»

Тест должен состоять из 10 вопросов для выбора одного (или нескольких) правильных ответов. Количество предложенных вариантов ответа на вопрос должно быть не менее трех. Формулировки вопросов и ответов на них хранятся в тестовом файле.

Вопросы и варианты ответов студент придумывает самостоятельно.

Не допускается, чтобы верными были все первые (вторые или третьи) ответы в тесте. Возможен случайный выбор вопросов из файла и случайное расположение вариантов ответов на экране. По желанию студента можно ввести временное ограничение на выполнение каждого вопроса или всего теста полностью.

**Виджеты**

Label – метки для вывода информации и вопросов для теста

Entry – однострочное текстовое поле для ввода данных (имени, фамилии и группы)

Button – кнопки для закрытия/начала теста/перелистывания вопросов/завершения теста

**Описание событий и связанных с ними методов**

Работа в программе начинается с регистрации пользователя.

Открывается окно (screen) в котором пользователю предлагается ввести: Фамилию, имя и группу.

По нажатию кнопки «Записать и начать» запускается метод (write\_File) который проверяет заполнены ли все окна и записывает данные в текстовый файл (users.txt). Если условие выполнено, то по нажатию «Записать и начать»  
окно регистрации закрывается и открывается окно (window).

В окне (window) пользователю предлагается выбор: «Начать» или «Выход».

1. Запускает тестирование.
2. Закрывается окно, используя метод (exit).

Нажатие кнопки «Начать» запускает метод (question) который присваивает кнопкам (but1, but2, but3, but4) названия ответов, и меняет текст надписи (info) с приветствия пользователя на сам вопрос, а также убирает кнопку «Начать».

Каждой из четырёх кнопок присвоен соответствующий метод (r\_q1, r\_q2, r\_q3, r\_q4) который сверяет текст нажатой пользователем кнопки, с ответами в начале, в списке (B). В случае если ответ дан верно, делается запись в консоль (проверка) и в файл (users.txt). Аналогично при неверном ответе.

Когда счётчик (Num) достигает конца, вызывается метод (end), выводится общее количество набранных баллов (ball), все кнопки удаляются, кроме кнопки «Выход», заканчивается запись в текстовый файл (users.txt)

**Текст программы**

from tkinter import \*

from tkinter import messagebox

spi = []

Num = -1

ball=0

B=["Импортировать модуль",

"Cоздать окно графики",

"Имя объекта = Circle(Point(x, y), R)",

"Имя объекта = Rectangle(Point(x1, y1), Point(x2, y2))",

"Имя объекта = Oval(Point(x1, y1), Point(x2, y2))",

"Имя объекта = Polygon(Point(x1, y1), Point(x2, y2),…, Point(xn, yn))",

"Имя объекта.move(dx, dy)",

"Имя объекта = Line(Point(x1, y1), Point(x2, y2))",

"Имя текстового объекта = Text(x, y, “Текст”)",

"Имя объекта.delete()"]

f = open('TestProgramm.txt', 'r', encoding='utf=8')

for i in range(10):

spi.append(f.readline().split(" \* "))

f.close()

def exit():

window.quit()

print(ball)

def write\_File():

file = open("users.txt","w",encoding = "utf-8" )

name\_File = input\_name.get()

forname\_File = input\_forname.get()

group\_File = input\_group.get()

text=('Имя: '+name\_File+'\n'+'Фамилия: '+forname\_File+'\n'+'Группа: '+group\_File+'\n')

file.write(text)

file.close()

#проверка на "дурака" с выводом места ошибки.

if len(name\_File) <=0:

messagebox.showinfo('Внимание!','Не заполненно Имя')

if len(forname\_File) <=0:

messagebox.showinfo('Внимание!', 'Не заполненна Фамилия')

if len(group\_File)<=0:

messagebox.showinfo('Внимание!', 'Не заполненна Группа')

if (len(name\_File)>0 and len(group\_File)>0 and len(forname\_File)>0)==True:

screen.destroy()#Закрытие окна регистрации

#Так сделанно для того чтоб по нажатию на крест не открывалось окно теста

def question():

global Num

if Num < len(spi) - 1:

Num += 1

info.config(text=spi[Num][0])

but1.config(text=spi[Num][1])

but2.config(text=spi[Num][2])

but3.config(text=spi[Num][3])

but4.config(text=spi[Num][4])

but6.destroy()

info.grid(row=1, column=15)

but1.grid(row=3, column=15)

but2.grid(row=4, column=15)

but3.grid(row=5, column=15)

but4.grid(row=6, column=15)

but5.grid(row=7, column=15)

else:

Num += 1

end()

def r\_q1():

global Num

global ball

file = open("users.txt", "a", encoding="utf-8")

if spi[Num][1] in B:

ball+=1

print(ball)

file.write('Вопрос № '+str(Num + 1)+' отвечен верно'+'\n')

file.close()

else:

print('Не верно')

file.write('Вопрос № '+str(Num + 1)+' отвечен неверно'+'\n')

file.close()

question()

def r\_q2():

global Num

global ball

file = open("users.txt", "a", encoding="utf-8")

if spi[Num][1] in B:

ball += 1

print(ball)

file.write('Вопрос № ' + str(Num + 1) + ' отвечен верно' + '\n')

file.close()

else:

print('Не верно')

file.write('Вопрос № ' + str(Num + 1) + ' отвечен неверно' + '\n')

file.close()

question()

def r\_q3():

global Num

global ball

file = open("users.txt", "a", encoding="utf-8")

if spi[Num][1] in B:

ball += 1

print(ball)

file.write('Вопрос № ' + str(Num + 1) + ' отвечен верно' + '\n')

file.close()

else:

print('Не верно')

file.write('Вопрос № ' + str(Num + 1) + ' отвечен неверно' + '\n')

file.close()

question()

def r\_q4():

global Num

global ball

file = open("users.txt", "a", encoding="utf-8")

if spi[Num][1] in B:

ball += 1

print(ball)

file.write('Вопрос № ' + str(Num + 1) + ' отвечен верно' + '\n')

file.close()

else:

print('Не верно')

file.write('Вопрос № ' + str(Num + 1) + ' отвечен неверно' + '\n')

file.close()

question()

def end():

but1.destroy()

but2.destroy()

but3.destroy()

but4.destroy()

info["text"]='Правильных ответов: '+str(ball)

info.place(x=110,y=50)

but5.place(x=150,y=80)

screen = Tk()

input\_name = Entry(screen)#имя

input\_forname = Entry(screen)#фамилия

input\_group = Entry(screen)#группа

name=Label(screen,text='Имя')

forname=Label(screen,text='Фамилия')

group=Label(screen,text='Группа')

name.grid(row=1, column=0)

forname.grid(row=2, column=0)

group.grid(row=3, column=0)

input\_name.grid(row=1, column=1)

input\_forname.grid(row=2, column=1)

input\_group.grid(row=3, column=1)

button\_Write =Button(screen, text = "Записать и начать", command=write\_File).grid(row=10, column=1)

screen.mainloop()

window = Tk()

window.geometry('380x200')

window.title("Сафи Адам")

window.configure(bg='orange')

info = Label(window, text='Добро пожаловать', bg='orange', fg='black')

but1 = Button(window, bg='cornsilk2',command=r\_q1)

but2 = Button(window, bg='cornsilk2',command=r\_q2)

but3 = Button(window, bg='cornsilk2',command=r\_q3)

but4 = Button(window, bg='cornsilk2',command=r\_q4)

but5 = Button(window, text="Выход",command=exit)

but6 = Button(window, text='Начать',command=question)

# Располагаем кнопки

info.place(x=110,y=50)

but5.place(x=150,y=100)

but6.place(x=148,y=75)

window.mainloop()

**Текст тестового файла (TestProgramm)**

1) Первым делом для работы с модулем graphics нужно: \* Импортировать модуль \* Создать окно графики \* Создать модуль \* Написать программу

2) Для работы с графикой в модуле graphics нужно: \* Импортировать модуль \* Cоздать окно графики \* Нарисовать квадрат \* Спросить у друга

3) Метод создающий линию: \* Line((x,y),(x,y)) \* Название окна = Line((x1, y1),(x2, y2)) \* Имя объекта = Line(Point(x1, y1), Point(x2, y2)) \* Line(Point(x1, y1), Point(x2, y2))

4) Метод создающий окружность: \* Имя объекта = Circle(Point(x, y), R) \* Circle(Point(x, y)) \* Circl(Point(x, y), R) \* Имя объекта = Krug(Point(x, y), R)

5) Метод создающий прямоугольник: \* Имя объекта = Rectangle(Point(x1, y1), Point(x2, y2)) \* Имя = Rectangle((x1, y1),(x2, y2)) \* Объект = Rectangl(Point(x1, y1), Point(x2, y2)) \* Имя объекта = Rectangle((x1, y1), (x2, y2))

6) Метод создающий эллипс: \* Имя объекта = Oval(Point(x1, y1), Point(x2, y2)) \* Имя объекта = Elips(Point(x1, y1), Point(x2, y2)) \* Имя объекта = Circle(Point(x, y), R) \* Имя объекта Polygon(Point(x1, y1), Point(x2, y2))

7) Метод создающий многоугольник: \* Имя объекта = Polygon(Point(x1, y1), Point(x2, y2)) \* Имя объекта = Polygon(Point(x1, y1), Point(x2, y2),…, Point(xn, yn)) \* Имя точки = Point(Point(x1, y1)) \* Имя объекта = Polygon((x1, y1), (x2, y2),…,(xn, yn)))

8) Метод смещающий объект: \* Имя объекта.move(dx, dy) \* Имя объекта = Move(x, y) \* Имя объекта.replace(x, y) \* Группа объектов = move(dx, dy)

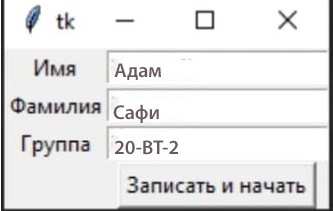
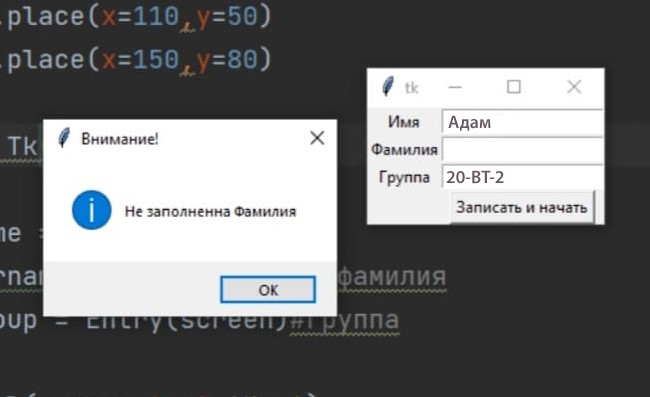
9) Метод создающий текст: \* Имя текстового объекта = Text(x, y, “Текст”) \* Имя текстового объекта = Text\_Print(x, y, “Текст”) \* Имя объекта = text(x, y, “Текст”) \* Имя текстового объекта = Print(x, y, “Текст”)

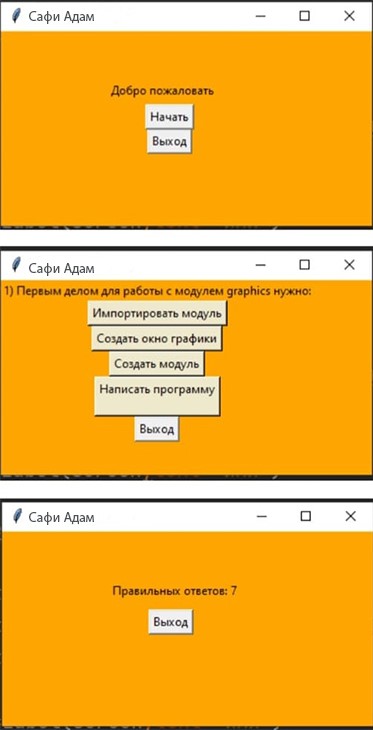
10) Метод удаляющий объект: \* Имя объекта.undraw() \* Имя объекта.delete() \* Имя объекта.annihilation() \* Имя объекта.undra-draw()

**Текст тестового файла(users.txt)**

Имя: Адам  
Фамилия: Сафи  
Группа: 20-Вт-2  
Вопрос № 1 отвечен верно  
Вопрос № 2 отвечен верно  
Вопрос № 3 отвечен неверно  
Вопрос № 4 отвечен верно  
Вопрос № 5 отвечен верно  
Вопрос № 6 отвечен верно  
Вопрос № 7 отвечен неверно  
Вопрос № 8 отвечен верно  
Вопрос № 9 отвечен верно  
Вопрос № 10 отвечен неверно

**Скриншоты выполнения программы:**

****

****

**Заключение**

Была написана программа-тест по теме «Модуль graphics. Создание графических примитивов.». Вопросы и ответы статичны, случайной постановки нет. Упор в работе программы был сделан на исправную работу программы, нежели на графическую составляющую. Вопросы для теста составлялись согласно информации из курсовой.

**Список использованных источников:**

Курс лекций доцент Заболотновой Е.Ю.

https://pythoner.name/

<https://docs-python.ru/tutorial/>

<https://metanit.com/python/tutorial/>

<https://pythonworld.ru/>

http://espressocode.top/

Курсовая работа студента Сафи.А.А.