**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО РЫБОЛОВСТВУ**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Калининградский государственный технический университет»**

Институт цифровых технологий

Кафедра цифровых систем и автоматики

**ОТЧЕТ**

по ознакомительной практике

Выполнил студент группы 23-ВТ-1,

Форма обучения очная, курс 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Сурудин

*Подпись (И.О. Фамилия)*

Руководитель практики от кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ доц. Е.Ю. Заболотнова

*Подпись (И.О. Фамилия)*

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

Дата защиты отчета \_\_\_\_\_ 11.07. 2024

Калининград

2024

Федеральное агентство по рыболовству

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Калининградский государственный технический университет»

(ФГБОУ ВО «КГТУ»)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кафедра | | Цифровых систем и автоматики | | | | | | | | | | | | | | | | | | | | | | |
| УТВЕРЖДАЮ | | | | | | | | | | | | | | | | | | | | | |
| Зав. кафедрой | | | |  | | | | / | В.И. Устич | | | | | | | | | | / |
| « | 28 | » | июня | | | | 2024 | | | |  | | | г. |
| Индивидуальное задание | | | | | | | | | | | | | | | | | | | | | | | | |
| на ознакомительную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности | | | | | | | | | | | | | | | | | | | | | | | | |
| студента | Сурудина Андрея Витальевича | | | | | | | | | | | | | | | | | , | | 23-ВТ | | | | |
| (Ф.И.О. полностью) | | | | | | | | | | | | | | | | | | | | (группа) | | | | |
| Направление подготовки (специальность) 09.03.01 Информатика и вычислительная техника | | | | | | | | | | | | | | | | | | | | | | | | |
| Место прохождения практики:  ФГБОУ ВО КГТУ, кафедра, 236022 Калининград, Советский проспект 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| За время прохождения практики: с | | | | | | « | 28 | » | | июня | | | | | | 2024 | | | |  | | | г. | |
| по | | « | 11 | » | | июля | | | | | 2024 | | | |  | | | г. | | |

студент должен выполнить следующие виды работ (заданий):

|  |  |  |
| --- | --- | --- |
| № п/п | Содержание практики  (наименование работ/заданий) | Рабочий график практики |
| 1 | Получение индивидуальных заданий студентами | 28.06-01.07 |
| 2 | Выполнение индивидуального задания: разработка программы для тестирования по заданной теме на основе GUI на алгоритмическом языке Python | 02.07-10.07 |
| 4 | Подготовка отчета по учебной практике, защита практики | 10.07.19- 11.07 |

**Планируемые результаты практики**

|  |  |
| --- | --- |
| **Компетенции выпускника ОП ВО и этапы их формирования** | **Знания, умения, навыки и опыт профессиональной деятельности** |
| ОПК-5: Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.  ОПК-5.4: Владеет практическими навыками инсталляции программного и аппаратного обеспечения. | Должен знать: - основные характеристики и параметры аппаратно-программных средств, учитываемые при их инсталляции; - структуру обзоров научной литературы и электронных информационных ресурсов.  Должен уметь: - находить нужные в данной ситуации стандарты, нормативы, методики, инструкции; - использовать известные механизмы подготовки обзоров научной литературы и электронных информационных ресурсов.  Должен владеть: - методиками инсталляции аппаратно- программных средств; - методиками и рекомендациями по подготовке обзоров научной литературы и электронных информационных ресурсов. Должен приобрести опыт: - разрешения конфликтных ситуаций при инсталляции разных программных пакетов на одну операционную и/или аппаратную платформу; - подготовки обзоров научной литературы и электронных информационных ресурсов |
| ПК-1: Способен разрабатывать требования, проектировать программное обеспечение (далее - ПО), документировать процедуру тестирования.  ПК-1.4: Знакомится с нормативными документами с применением электронных информационно-образовательных ресурсов в профессиональной деятельности. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель практики от университета | |  | | | |  | | доцент Е.Ю. Заболотнова | | | | | |
|  | | (подпись) | | | |  | | (Фамилия И.О., должность) | | | | | |
|  | |  | | | |  | |  | | | | | |
| Практикант |  | |  |  | | | | | | | | | |
|  | (подпись) | |  | (телефон, E-mail) | | | | | | | | | |
|  |  | |  |  | | | | | | | | | |
| « | | 11 | | » | июля | 2024 |  | г. |

***Введение***

Целью ознакомительной 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 и методов и принципов объектно-ориентированного программирования.

***Основная часть***

***Постановка задачи*:**

Написать программу для прохождения тестирования по теоретическому материалу дисциплины «Программирование». Тема для тестирования совпадает с темой из первой части курсовой работы студента.

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.

Программа работает с пользовательским графическим интерфейсом, который должен содержать следующие элементы, в скобках указано название виджета для реализации:

Название дисциплины: Программирование (метка);

Название темы: смотри первую часть своей курсовой работы (метка);

Кнопки для начала и завершения тестирования;

Номер и текст вопроса (метка или текстовое поле);

Варианты ответов (метки или текстовые поля);

Выбор правильного ответа (ответов) можно реализовать через радиокнопки или флажки или путем ввода номера правильного ответа в текстовое поле.

По окончании теста программа должна вывести правильные ответы и ответы студента, возможно обозначение цветом или иным способом правильных и неправильных ответов студента, подсчитать количество правильных ответов и набранный балл по результатам теста (1 балл за один верный ответ).

***Описание классов:***

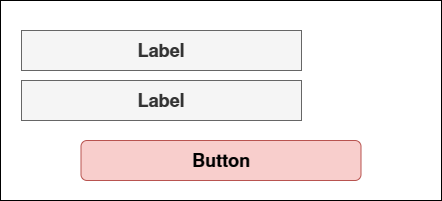
class App() – класс является основной частью программы в данной работе.  
Он содержит в себе виджеты, а также функции: start\_test() - отвечает за запуск тестирования; read\_file() – отвечает за открытие и обработку данных из файла; end\_test()- отвечает за подсчет результатов тестирования, on\_click()-отвечает за завершение теста, next\_test()-отвечает за переход к следующему вопросу теста, prev\_test()-отвечает за возвращение к предыдущему вопросу теста, generate\_widgets()-отвечает за создание виджетов вопросов на основе данных из файла

***Макет GUI:***

При разработке графического интерфейса использовались виджеты «Label», «Radiobox», «Button», а именно:

1. Основное окно (Рис. 1):

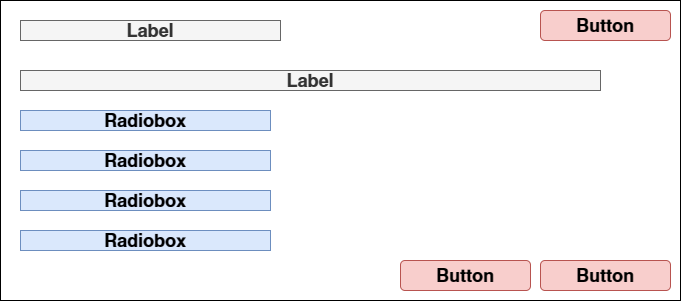
* Надпись с названием дисциплины
* Надпись с названием темы
* Кнопка для начала тестирования

******

***Рис. 1 – Основное окно***

1. Окно тестирования (Рис. 2):

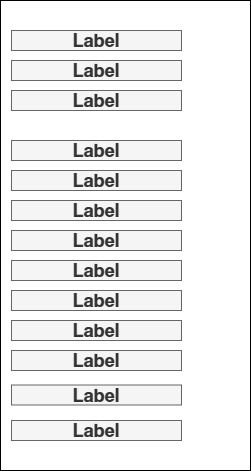
* Надпись с номером вопроса
* Надпись с формулировкой вопроса
* Радиокнопки с вариантами ответов
* Кнопка завершения теста
* Кнопка для перехода к следующему вопросу
* Кнопка для возвращения к предыдущему вопросу

******

***Рис. 2 – Окно тестирования***

1. Окно с результатами тестирования (Рис. 3):

* Надпись с указанием количества полученных баллов
* Надписи с правильными ответами и выбранными пользователем

******

***Рис. 3 – Окно с результатами тестирования***

***Описание событий и методов:***

При запуске программы открывается основное окно (рис. 1). При нажатии на кнопку «Начать тестирование» программа на основе извлеченных данных из файла создает виджеты с названием вопроса и с вариантами ответов, представленными в программе в виде радиокнопок. При нажатии на кнопку «Следующий вопрос» формулировка вопроса и варианты ответов меняются на последующие, а при нажатии на «Предыдущий вопрос» соответственно меняются на предыдущие. При нажатии на кнопку «Завершить тест» открывается диалоговое окно с подтверждением завершения теста. При подтверждении завершения теста все виджеты стираются; выводятся правильные ответы на вопросы и ответы пользователя, если они совпадают задний фон надписи окрашивается в зеленый, а если не совпадают, то в красный; подсчитывается итоговой балл за тест

***Текст программы:***

***import TKinterModernThemes as TKMT***

***from tkinter import ttk, Variable, messagebox***

***class App(TKMT.ThemedTKinterFrame):***

***def \_\_init\_\_(self):***

***super().\_\_init\_\_("Сурудин Андрей 23 ВТ-1", "sun-valley", "dark", True, True)***

***self.q\_numb = 0***

***self.frame = ttk.Frame(name='main')***

***self.d\_name = ttk.Label(self.frame.master, text='Дисциплина: Программирование')***

***self.t\_name = ttk.Label(self.frame.master, text='Тема: "Условный оператор в Python"')***

***self.start\_btn = ttk.Button(self.frame.master, text='Начать тестирование', command=self.start\_test)***

***self.test = []***

***ans1, ans2, ans3, ans4, ans5, ans6, ans7, ans8, ans9, ans10 = Variable(), Variable(), Variable(), Variable(), Variable(), Variable(), Variable(), Variable(), Variable(), Variable()***

***self.vars = {***

***0: ans1,***

***1: ans2,***

***2: ans3,***

***3: ans4,***

***4: ans5,***

***5: ans6,***

***6: ans7,***

***7: ans8,***

***8: ans9,***

***9: ans10***

***}***

***self.frame.grid(row=0, column=0, padx=10, pady=10, sticky='nsew')***

***self.d\_name.grid(row=0, column=0, padx=10, pady=10, sticky='nsew')***

***self.t\_name.grid(row=1, column=0, padx=10, pady=10, sticky='nsew')***

***self.start\_btn.grid(row=2, column=0, padx=10, pady=10, sticky='nsew')***

***self.read\_file()***

***self.run(cleanresize=False)***

***def start\_test(self):***

***self.generate\_widgets()***

***def read\_file(self):***

***with open('C:/Users/5050429/Desktop/ВСЁ ЧТО СВЯЗАНО С КГТУ/tests.txt', 'r', encoding='utf-8') as file:***

***content = file.readlines()***

***ans\_index = content.index('Ответы:\n')***

***for i in range(len(content)):***

***if content[i] == '\n' and content[i + 1] != 'Ответы:\n':***

***ans\_index += 1***

***question = {'title': content[i + 1].rstrip(),***

***'answers': [content[i + 2].rstrip(), content[i + 3].rstrip(),***

***content[i + 4].rstrip(), content[i + 5].rstrip()],***

***'true': content[ans\_index].rstrip()}***

***self.test += [question]***

***def end\_test(self):***

***k = 0***

***label = ttk.Label(self.frame.master, text='Вы завершили тестирование')***

***label3 = ttk.Label(self.frame.master, text='Правильные ответы:')***

***answers = []***

***for i in self.vars.values():***

***answers += [i]***

***for i in range(len(self.vars)):***

***if self.test[i]['true'] == answers[i].get():***

***color = 'green'***

***k += 1***

***else:***

***color = 'red'***

***quest = ttk.Label(self.frame.master,***

***text=f'{i + 1}) {self.test[i]["true"]} (Ваш ответ: "{answers[i].get()}")',***

***background=color)***

***quest.grid(row=i + 4, column=0, padx=10, pady=10, sticky='nsew')***

***label2 = ttk.Label(self.frame.master, text=f'Вы набрали {k} балл(-а)(-ов) из 10 возможных')***

***label.grid(row=0, column=0, padx=10, pady=10, sticky='nsew')***

***label2.grid(row=1, column=0, padx=10, pady=10, sticky='nsew')***

***label3.grid(row=3, column=0, padx=10, pady=10, sticky='nsew')***

***def on\_click(self):***

***result = messagebox.askquestion('Диалоговое окно', 'Вы точно хотите завершить тест?')***

***if result == 'yes':***

***for i in self.frame.master.winfo\_children():***

***i.destroy()***

***self.end\_test()***

***def next\_test(self):***

***self.q\_numb += 1***

***self.generate\_widgets()***

***def prev\_test(self):***

***self.q\_numb -= 1***

***self.generate\_widgets()***

***def generate\_widgets(self):***

***question = ttk.Label(self.frame.master, text=f'Вопрос номер { self.q\_numb + 1}')***

***title = ttk.Label(self.frame.master, text=self.test[self.q\_numb]['title'])***

***end\_btn = ttk.Button(self.frame.master, text='Завершить тест', command=self.on\_click)***

***next\_btn = ttk.Button(self.frame.master, name='next', text='Следующий вопрос', command=self.next\_test)***

***prev\_btn = ttk.Button(self.frame.master, name='back', text='Предыдущий вопрос', command=self.prev\_test)***

***answer1 = ttk.Radiobutton(self.frame.master, text=self.test[self.q\_numb]['answers'][0],***

***value=self.test[self.q\_numb]['answers'][0], variable=self.vars[self.q\_numb])***

***answer2 = ttk.Radiobutton(self.frame.master, text=self.test[self.q\_numb]['answers'][1],***

***value=self.test[self.q\_numb]['answers'][1], variable=self.vars[self.q\_numb])***

***answer3 = ttk.Radiobutton(self.frame.master, text=self.test[self.q\_numb]['answers'][2],***

***value=self.test[self.q\_numb]['answers'][2], variable=self.vars[self.q\_numb])***

***answer4 = ttk.Radiobutton(self.frame.master, text=self.test[self.q\_numb]['answers'][3],***

***value=self.test[self.q\_numb]['answers'][3], variable=self.vars[self.q\_numb])***

***if self.q\_numb != 0:***

***prev\_btn.grid(row=6, column=1, padx=10, pady=10, sticky='nsew')***

***else:***

***prev\_btn.destroy()***

***if self.q\_numb != 9:***

***next\_btn.grid(row=6, column=2, padx=10, pady=10, sticky='nsew')***

***else:***

***next\_btn.destroy()***

***prev\_btn.grid(row=6, column=2, padx=10, pady=10, sticky='nsew')***

***question.grid(row=0, column=0, padx=10, pady=10, sticky='nsew')***

***end\_btn.grid(row=0, column=2, padx=10, pady=10, sticky='nsew')***

***title.grid(row=1, column=0, padx=10, pady=10, sticky='nsew')***

***answer1.grid(row=2, column=0, padx=10, pady=10, sticky='nsew')***

***answer2.grid(row=3, column=0, padx=10, pady=10, sticky='nsew')***

***answer3.grid(row=4, column=0, padx=10, pady=10, sticky='nsew')***

***answer4.grid(row=5, column=0, padx=10, pady=10, sticky='nsew')***

***if \_\_name\_\_ == "\_\_main\_\_":***

***App()***

***Текст тестового файла:***

***Вопросы:***

***Какой оператор в Python проверяет равенство?***

***=***

***==***

***===***

***Нет правильного ответа***

***Как начинается условный оператор в Python?***

***if***

***when***

***switch***

***Нет правильного ответа***

***Для чего предназначен условный оператор if?***

***После создания выводит противоположные значения***

***Для выбора к использованию одного из возможных действий в зависимости от некоторого условия***

***Создаёт дополнительное условие***

***Нет правильного ответа***

***Составные условия строятся из простых с помощью?***

***логических операций and, or, not***

***арифметических операций +, - , \****

***с помощью операций and, or, not, +, - ,***

***Нет правильного ответа***

***Логическая операция and - это?***

***логическое И***

***логическое ИЛИ***

***логическое отрицание***

***Нет правильного ответа***

***Слово if переводится как?***

***если***

***то***

***иначе***

***Нет правильного ответа***

***Слово else переводится как?***

***если***

***то***

***иначе***

***Нет правильного ответа***

***Логическая операция or - это?***

***логическое И***

***логическое ИЛИ***

***логическое отрицание***

***Нет правильного ответа***

***Логическая операция not- это?***

***логическое И***

***логическое ИЛИ***

***логическое отрицание***

***Нет правильного ответа***

***В какой записи есть ошибка?***

***a=int(input())***

***print (c)***

***print("u", u)***

***Нет правильного ответа***

***Ответы:***

***==***

***if***

***Для выбора к использованию одного из возможных действий в зависимости от некоторого условия***

***Нет правильного ответа***

***логическое И***

***если***

***иначе***

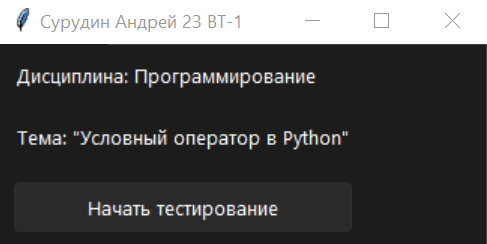
***логическое ИЛИ***

***логическое отрицание***

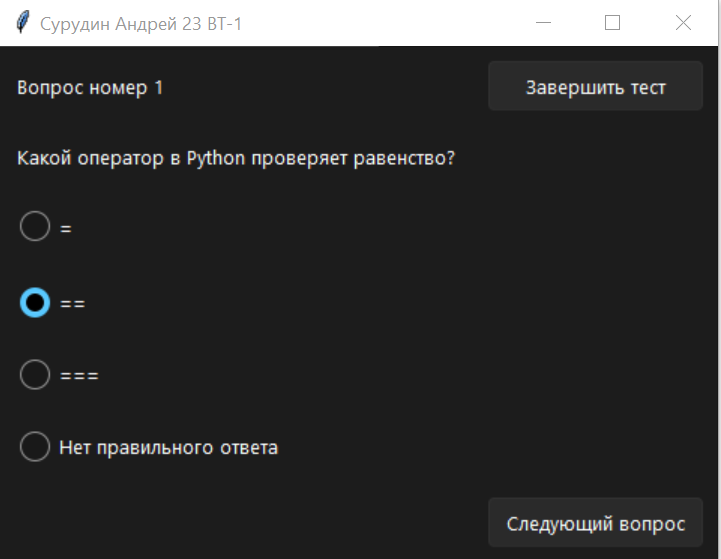
***Нет правильного ответа***

***Скриншоты выполнения программы:***

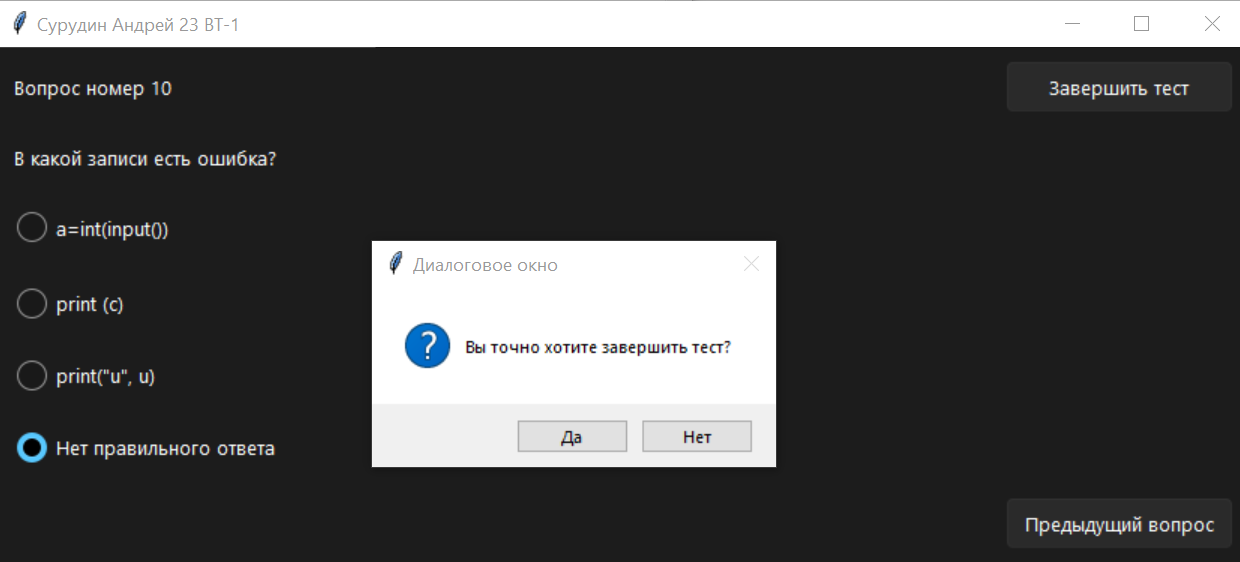
***Рис. 4 – Начальное окно***

******

***Рис. 5 - Тестирование***

******

***Рис. 6 – Подтверждение завершения тестирования***

******

***Рис. 7 – Результаты тестирования***

******

***Заключение:***

При выполнении ознакомительной практики я закрепил и расширил свои теоретические знания и практические навыки, полученные мной в ходе учебного процесса: работа с файлами, разработка программ на основе графического интерфейса и методов и принципов объектно-ориентированного программирования.

***Список используемых источников:***

https://videouroki.net/tests/8-3-3-python-uslovnyi-opierator.html