Шаймарданов Д.А.

Зверева Н.Н.

ИСТ-216Б

Геоинформационных систем

1307.531600.000 ПЗ

Технологии программирования

Уфа – 2022

Регрессионное тестирование программного обеспечение

Содержание

[1. Цель и задачи работы 3](#_Toc122669964)

[2. Задание 1 4](#_Toc122669965)

[4. Вывод 7](#_Toc122669966)

[Список литературы 8](#_Toc122669967)

**1. Цель и задачи работы**

Цель работы: изучение принципов регрессионного тестирования программного обеспечения.

Задачи работы:

– изучить принципы регрессионного тестирования;

– разработать систему тестов программ;

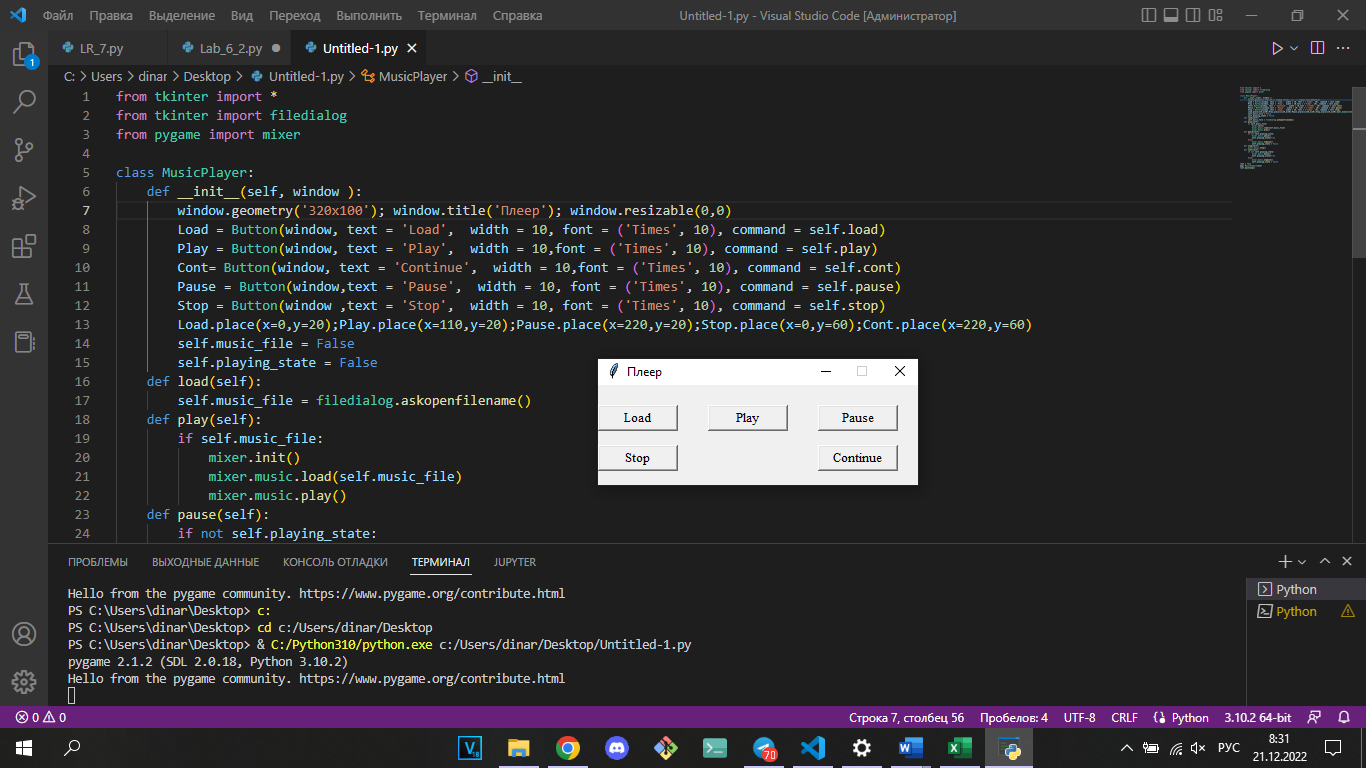
– составить и защитить отчет.

**2. Задание 1**

Создать плеер с возможностью старт/стоп. Добавить новую функцию пауза/продолжение. Проверить на наличие ошибок, если есть исправить. Добавить любую функцию, так же проверить на наличие ошибок.

Результат работы программы представлен на рисунке 1.

Код программы представлен ниже.



1. - результат работы программы

from tkinter import \*

from tkinter import filedialog

from pygame import mixer

class MusicPlayer:

    def \_\_init\_\_(self, window ):

        window.geometry('320x100'); window.title('Iris Player'); window.resizable(0,0)

        Load = Button(window, text = 'Load',  width = 10, font = ('Times', 10), command = self.load)

        Play = Button(window, text = 'Play',  width = 10,font = ('Times', 10), command = self.play)

        Cont= Button(window, text = 'Continue',  width = 10,font = ('Times', 10), command = self.cont)

        Pause = Button(window,text = 'Pause',  width = 10, font = ('Times', 10), command = self.pause)

        Stop = Button(window ,text = 'Stop',  width = 10, font = ('Times', 10), command = self.stop)

        Load.place(x=0,y=20);Play.place(x=110,y=20);Pause.place(x=220,y=20);Stop.place(x=0,y=60);Cont.place(x=220,y=60)

        self.music\_file = False

        self.playing\_state = False

    def load(self):

        self.music\_file = filedialog.askopenfilename()

    def play(self):

        if self.music\_file:

            mixer.init()

            mixer.music.load(self.music\_file)

            mixer.music.play()

    def pause(self):

        if not self.playing\_state:

            mixer.music.pause()

            self.playing\_state=True

        else:

            mixer.music.unpause()

            self.playing\_state = False

    def stop(self):

        mixer.music.stop()

    def cont(self):

        if not self.playing\_state:

            mixer.music.pause()

            self.playing\_state=True

        else:

            mixer.music.unpause()

            self.playing\_state = False

root = Tk()

app= MusicPlayer(root)

root.mainloop()

В первой версии программы были добавлены стандартные функции, была выявлена ошибка с остановкой программы на кнопку «Стоп», не происходила остановка песни.

Результаты представлены в таблице 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Плеер v1 | | | | |
| № п/п | Тест | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка кнопки "play" | Начало проигрывания песни | Начало проигрывания песни | Успешно |
| 2 | Проверка кнопки "pause" | Остановка проигрывания песни | Остановка проигрывания песни | Успешно |
| 3 | Проверка кнопки "stop" | Прекращение проигрывания песни | Прекращение проигрывание песни не происходит | Не успешно |

Таблица 1 - первая версия плеера

Во второй версии были исправлены предыдущие ошибки, была добавлена новая функция - продолжение на кнопку «continue», была выявлена ошибка с продолжением песни на эту кнопку.

Результаты представлены в таблице 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Плеер с функцией продолжение v2 | | | | |
| № п/п | Тест | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка кнопки "play" | Начало проигрывания песни | Начало проигрывания песни | Успешно |
| 2 | Проверка кнопки "pause" | Остановка проигрывания песни | Остановка проигрывания песни | Успешно |
| 3 | Проверка кнопки "stop" | Прекращение проигрывания песни | Прекращение проигрывание песни | Успешно |
| 4 | Проверка кнопки "continue" | Продолжение проигрывание песни с того же места | Проигрывание песни с самого начала | Не успешно |

Таблица 2 - вторая версия плеера

В третьей версии были исправлены предыдущие ошибки, была добавлена новая функция - загрузка определенных песен из папки, была выявлена ошибка с загрузкой песен на кнопку «Load».

Результаты представлены в таблице 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Плеер с новой функцией v3 | | | | |
| № п/п | Тест | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка кнопки "play" | Начало проигрывания песни | Начало проигрывания песни | Успешно |
| 2 | Проверка кнопки "pause" | Остановка проигрывания песни | Остановка проигрывания песни | Успешно |
| 3 | Проверка кнопки "stop" | Прекращение проигрывания песни | Прекращение проигрывание песни | Успешно |
| 4 | Проверка кнопки "continue" | Продолжение проигрывание песни с того же места | Продолжение проигрывание песни с того же места | Успешно |
| 5 | Проверка кнопки "Load" | Загрузка определенного трека из папки | Ошибка: не удалось найти файл | Не успешно |

Таблица 3 - третья версия плеера

В четвертой версии были исправлены предыдущие ошибки, больше ошибок выявлено не было.

Результаты представлены в таблице 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Плеер без ошибок v4 | | | | |
| № п/п | Тест | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка кнопки "play" | Начало проигрывания песни | Начало проигрывания песни | Успешно |
| 2 | Проверка кнопки "pause" | Остановка проигрывания песни | Остановка проигрывания песни | Успешно |
| 3 | Проверка кнопки "stop" | Прекращение проигрывания песни | Прекращение проигрывание песни | Успешно |
| 4 | Проверка кнопки "continue" | Продолжение проигрывание песни с того же места | Продолжение проигрывание песни с того же места | Успешно |
| 5 | Проверка кнопки "Load" | Загрузка определенного трека из папки | Загрузка определенного трека из папки | Успешно |

Таблица 4 - четвертая версия плеера

**4. Вывод**

Были изучены и применены на практике методы регрессионного тестирования. Тестирование выявило ошибки в программах, которые были исправлены.

Список литературы

1. Варфоломеева, Т. Н. Структуры данных и основные алгоритмы их обработки: учебное пособие / Т. Н. Варфоломеева. – Москва: ФЛИНТА, 2017. – 159 с. – ISBN 978-5-9765-3691-3. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/104903 (дата обращения: 04.05.2020). – Режим доступа: для авториз. пользователей.

2. Вычислительная геометрия. Алгоритмы и приложения / д. Б. Марк, Ч. Отфрид, в. К. Марк, О. Марк; перевод с английского А. А. Слинкин. – 3-е изд. – Москва: ДМК Пресс, 2017. – 438 с. – ISBN 978-5-97060-406-9. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/105833 (дата обращения: 04.05.2020). – Режим доступа: для авториз. пользователей.

3. Васильев А. Н. Python на примерах. Практический курс по программированию. 2-е издание. – СПб.: Наука и Техника, 2017. – 432 е.: ил