

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» (УрФУ)

Физико-технологический институт

Кафедра «Технической физики»

Оценка

Руководитель курсового

проектирования

Члены комиссии

Дата защиты

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по теме: Разработка приложения "Тест: Таблица чисел"

по дисциплине: Прикладное программирование

Студент: Мусатова Снежана Александровна

(ФИО) (Подпись)

Группа: ФТ - 230008

Екатеринбург

2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc186227836)

[1 Постановка задачи 4](#_Toc186227837)

[2 Описание алгоритмов программы 5](#_Toc186227838)

[2.1 Словесное описание алгоритмов программы 5](#_Toc186227839)

[2.2 Блок-схемы алгоритмов основных функций 6](#_Toc186227840)

[3 Реализация программы 10](#_Toc186227841)

[3.1 Листинг программы 10](#_Toc186227842)

[3.2 Тестирование функциональных и нефункциональных характеристик программы 15](#_Toc186227843)

[4 Руководство пользователя 19](#_Toc186227844)

[4.1 Назначение программы 19](#_Toc186227845)

[4.2 Запуск программы 19](#_Toc186227846)

[4.3 Выполнение основных функций 19](#_Toc186227847)

[4.4 Завершение программы 20](#_Toc186227848)

[4.5 Пример работы с программой 20](#_Toc186227849)

[Заключение 22](#_Toc186227850)

[Библиографический список 23](#_Toc186227851)

# ВВЕДЕНИЕ

В условиях современного информационного общества, развитие умственных способностей становится важным аспектом личной эффективности. Особое внимание уделяется развитию памяти, внимательности и способности к быстрой реакции. В этом контексте методики, направленные на тренировки мозга, приобретают все большую популярность. Одним из наиболее узнаваемых упражнений для тренировки когнитивных функций является игра "Таблица Шульте".

Таблицы Шульте представляют собой сетку с случайно размещенными числами, которые игрок должен находить и отмечать в порядке возрастания. Благодаря простоте правил и возможности адаптировать игру под разные уровни сложности, она подходит как для детей, так и для взрослых. В данной курсовой работе рассматривается процесс разработки интерактивного приложения, которое позволит пользователям играть в таблицу Шульте. Целью разработки является создание удобного и функционального интерфейса, который способствовал бы эффективной тренировке памяти и внимательности.

# **1 Постановка задачи**

Основной задачей данной курсовой работы является разработка приложения, реализующего игру "Таблица Шульте". Для достижения поставленной цели необходимо решить несколько более конкретных задач:

а) Генерация таблицы;

* Реализовать алгоритм для создания таблицы размером n\*m, хаотично заполненной числами от 1 до n\*m.

б) Создание графического интерфейса;

* Разработать интуитивно понятный графический интерфейс;
* Интерфейс должен включать элементы управления для начала и завершения игры, а также отображение затраченного времени.

в) Обработка взаимодействия с игроком;

* Написать функционал, который будет обрабатывать нажатия на ячейки таблицы;
* Обеспечить отображение следующей ячейки.

г) Реализация учета времени;

* Разработать секундомер, который будет отсчитывать время, затраченное игроком на нахождение всех чисел;
* Время должно отображаться в реальном времени и показываться по окончании игры.

д) Тестирование приложения;

* Провести тестирование программы для проверки функциональных и нефункциональных характеристик;
* Убедиться в корректности работы программы и устойчивости к ошибкам.

е) Создание руководства пользователя.

* Подготовить подробное руководство для пользователей, где будут описаны правила игры, интерфейс приложения и инструкции по его использованию.

# **2 Описание алгоритмов программы**

# **2.1 Словесное описание алгоритмов программы**

Алгоритм программы состоит из нескольких этапов:

Сначала программа получает размеры таблицы. Вызывается функция get\_two\_variables(), которая открывает окно для ввода ширины и высоты таблицы и проверяет, введены ли положительные числовые значения. Если введены некорректные данные, отображается сообщение об ошибке. После нажатия кнопки подтверждения программа возвращает заданные размеры таблицы.

Затем создается окно с информацией об игре. Функция info() вызывается после введения размеров таблицы. Она открывает новое окно, отображающее текущее время игры и следующее число, на которое нужно щелкнуть. Программа обновляет таймер каждую 1/10 секунды, показывая время, прошедшее с начала игры.

Вместе с предыдущим окном появляется окно игровой таблицы. После установки размеров таблицы создается экземпляр класса ShulteTable. В конструкторе \_\_init\_\_ класс перемешивает набор чисел для выбранных размеров и отображает их в сетке Tkinter в виде виджетов (label). Каждое число имеет обработчик событий на нажатие (лямбда-функция), который вызывает метод on\_click.

Каждое нажатие поддается обработке с помощью метода on\_click. Проверяется, совпадает ли нажатое число с текущим числом (curr\_num), которое нужно найти. Если число совпадает, виджет изменяет свой цвет, и программа обновляет следующее число для поиска. Если игрок нашел все числа, вызывается функция end\_game, которая завершает игру и выводит время, затраченное на поиск чисел.

В итоге, когда все числа найдены, отображается итоговое время и программа предлагает кнопку "Выйти", завершающую работу приложения.

# **2.2 Блок-схемы алгоритмов основных функций**

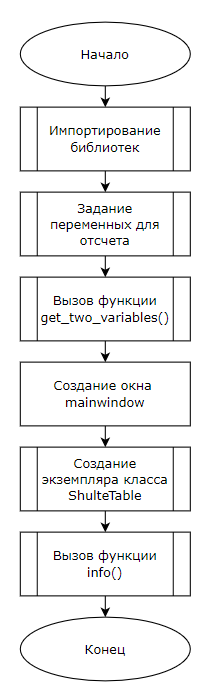


Рисунок 1 – Общая блок схема приложения

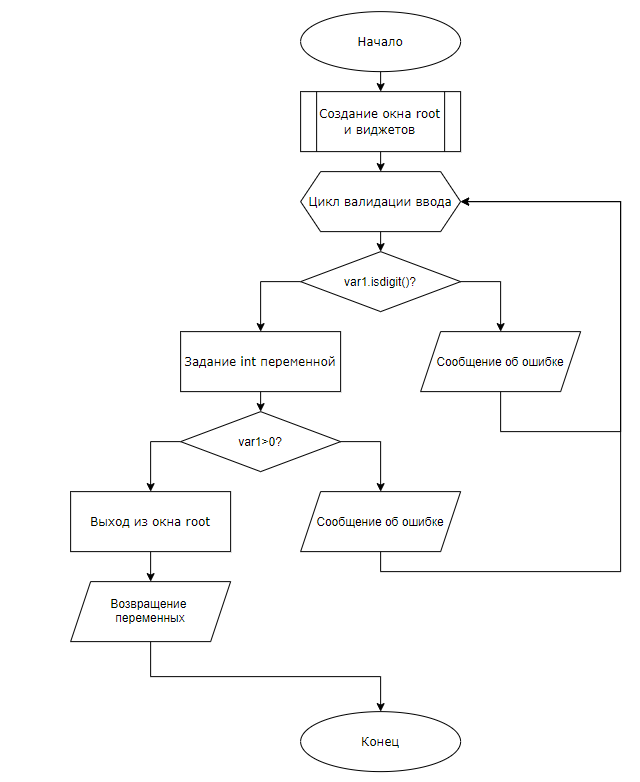


Рисунок 2 – Блок-схема функции get\_two\_variables()

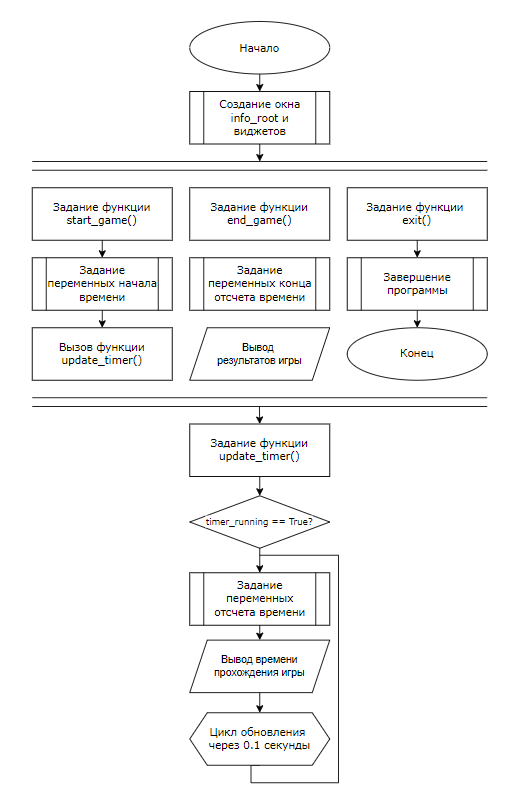


Рисунок 3 – Блок схема окна информации

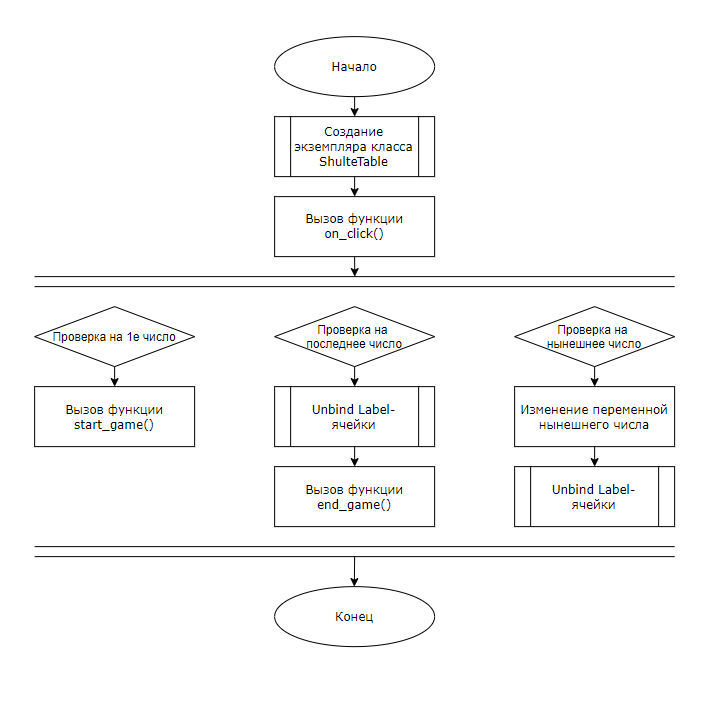


Рисунок 4 – Блок схема класса ShulteTable

# **3 Реализация программы**

# **3.1 Листинг программы**

1. import tkinter as tk
2. import random
3. import time
4. import sys
5. start\_time = 0
6. timer\_running = False
7. curr\_num = 1
8. def get\_two\_variables():
9. def submit():
10. while True:
11. var1 = entry\_var1.get()
12. var2 = entry\_var2.get()
13. if (var1.strip("-").isdigit()) and (var2.strip("-").isdigit()):
14. var1 = int(var1)
15. var2 = int(var2)
16. if (var1 > 0) and (var2 > 0):
17. root.quit()
18. button\_submit.destroy()
19. return var1, var2
20. else:
21. label\_error1.pack(pady = 5)
22. label\_error1["text"] = "Введите положительные значения"
23. root.update()
24. pass
25. else:
26. label\_error1.pack(pady = 5)
27. label\_error1["text"] = "Введите числовые значения"
28. root.update()
29. pass
30. root = tk.Tk()
31. root.title("Введите размеры таблицы")
32. root.resizable(False,False)
33. root.geometry("200x225+100+100")
34. label\_var1 = tk.Label(root, text="Ширина таблицы:")
35. label\_var1.pack(pady=5)
36. entry\_var1 = tk.Entry(root)
37. entry\_var1.pack(pady=5)
38. label\_var2 = tk.Label(root, text="Высота таблицы:")
39. label\_var2.pack(pady=5)
40. entry\_var2 = tk.Entry(root)
41. entry\_var2.pack(pady=5)
42. button\_submit = tk.Button(root, text="Подтвердить", command=submit)
43. button\_submit.pack(pady=20)
44. label\_error1 = tk.Label(root, text="")
45. root.mainloop()
46. return int(entry\_var1.get()), int(entry\_var2.get())
47. def info():
48. global k, timer\_running, start\_game, end\_game, main, curr\_num
50. info\_root = tk.Tk()
51. info\_root.title("Информация")
52. info\_root.resizable(False, False)
53. info\_root.geometry("310x150+800+100")
54. def start\_game():
55. global start\_time, timer\_running
56. start\_time = time.time()
57. timer\_running = True
58. update\_timer()
60. def end\_game():
61. global timer\_running
62. timer\_running = False
63. elapsed\_time = time.time() - start\_time
64. timer\_label.config(text = f"Вы закончили! Время: {elapsed\_time:.2f} секунд")
65. error\_label.config(text = f"")
66. def update\_timer():
67. if timer\_running:
68. elapsed\_time = time.time() - start\_time
69. timer\_label.config(text=f"Время: {elapsed\_time:.2f} секунд")
70. error\_label.config(text=f"Следующее число: {curr\_num}")
71. info\_root.after(100, update\_timer)
72. def exit():
73. sys.exit()
74. timer\_label = tk.Label(info\_root, text="Время: 0.00 секунд", font=("Times New Roman", 14, "bold"))
75. timer\_label.pack(pady = 10)
76. error\_label = tk.Label(info\_root, text=f"Следующее число: {curr\_num}")
77. error\_label.pack(pady = 10)
79. button\_exit = tk.Button(info\_root, text="Выйти", command=exit)
80. button\_exit.pack(pady = 5)
81. class ShulteTable:
82. def \_\_init\_\_(self, master):
83. self.master = master
84. self.master.title("Таблица Шульте")
85. self.master.resizable(False,False)
86. self.master.geometry("+400+100")
88. self.rows = n
89. self.columns = m
90. self.numbers = list(range(1, n \* m + 1))
91. random.shuffle(self.numbers)
92. self.start\_time = None
93. self.current\_number = 1
94. self.last\_number = self.rows \* self.columns
95. self.labels = {}
96. k = 0
97. for i in range(self.rows):
98. for j in range(self.columns):
99. number = self.numbers[k]
100. label = tk.Label(master, text=str(number), width=5, height=2, borderwidth=2, relief='groove')
101. label.grid(row=i, column=j, padx=5, pady=5)
102. label.bind("<Button-1>", lambda event, num=number: self.on\_click(num))
103. self.labels[number] = label
104. k += 1
105. def on\_click(self, number):
106. global start\_game, end\_game, curr\_num
107. if number == 1 and self.start\_time is None:
108. start\_game()
110. if number == self.last\_number and self.current\_number == self.last\_number:
111. label = self.labels[number]
112. label.config(fg = 'white')
113. label.unbind("<Button-1>")
114. end\_game()
115. elif number == self.current\_number:
116. self.current\_number += 1
117. curr\_num = self.current\_number
118. label = self.labels[number]
119. label.config(fg = 'white')
120. label.unbind("<Button-1>")
121. n, m = get\_two\_variables()
122. mainwindow = tk.Tk()
123. shulte\_app = ShulteTable(mainwindow)
124. info\_w = info()
125. mainwindow.mainloop()

# **3.2 Тестирование функциональных и нефункциональных характеристик программы**

Функциональное тестирование направлено на проверку того, выполняет ли приложение заявленные функции. В рамках этого тестирования можно выделить следующие аспекты.

В первую очередь следует проверить корректность отображения таблицы, т.е. убедиться, что таблица генерируется с правильным набором чисел, а также что числа располагаются в случайном порядке (Рисунок 5).

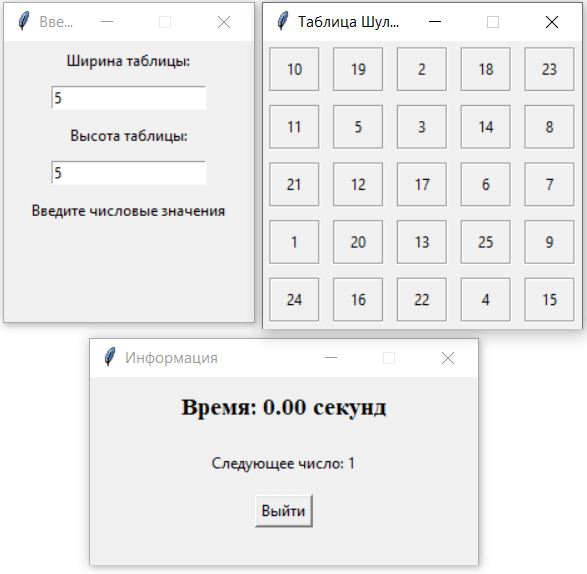


Рис 5 – Создание окон при правильном вводе

Далее следует протестировать взаимодействия с пользователем. В этот пункт входит проверка возможности начинать и останавливать игру, а также тестирование реакции приложения на нажатие пользователем на числа (Рисунок 6,7).

Также требуется провести тестирование функционала подсчета времени, в том числе корректности отображения времени во время игры и по её завершению (Рисунок 6,8).

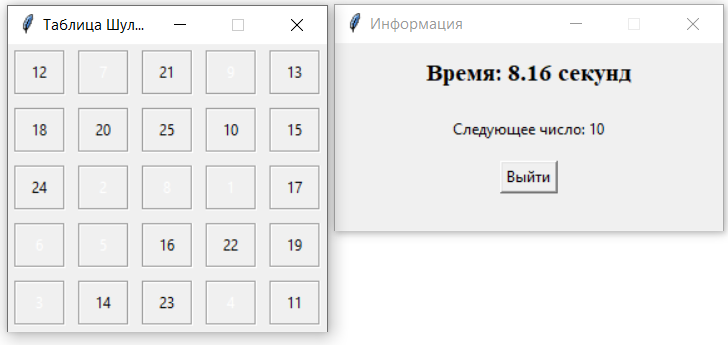


Рис 6 – Обработка нажатия ячеек

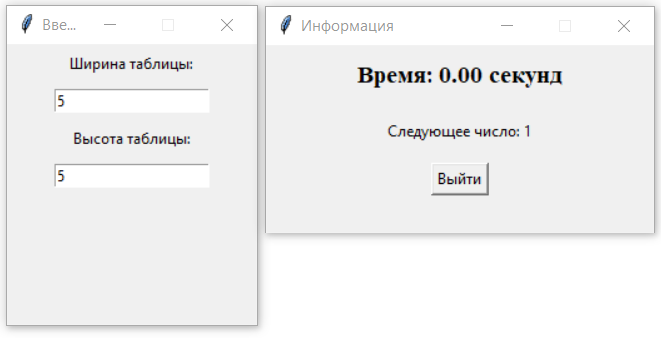


Рис 7 – Взаимодействие с пользователем после подтверждения размера таблицы

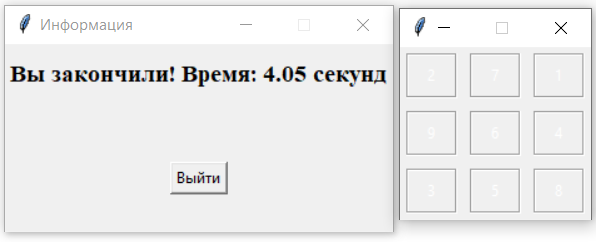
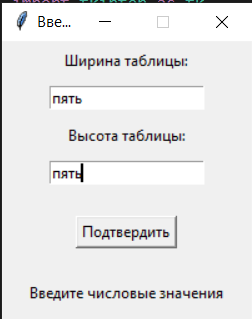
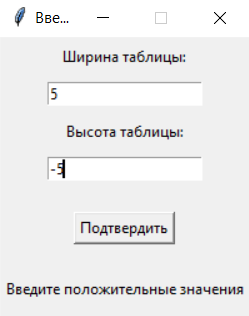


Рис 8 – Завершение программы

И наконец, тестирование ввода и валидации данных, необходимое для корректного задания размеров и создания таблицы чисел. Производится проверка на числовое и положительное значение (Рисунок 9).

а) б)

Рис 9 – Обработка ввода числовых (а) и положительных (б) значений

Нефункциональное тестирование оценивает характеристики приложения, не связанные непосредственно с его функциональностью. Это может включать производительность, удобство использования и совместимость.

Благодаря написанной валидации данных программа работает корректно и без задержек. Время запуска и завершения программы не превышает 1 секунды (при вводе стандартных размеров таблицы, до 10x10), что подтверждает её производительность.

Интерфейс приложения прост и понятен. Простота навигации и логичное размещение элементов интерфейса обеспечивают комфортное взаимодействие пользователя с приложением. Использование базовых, контрастных цветов улучшает восприятие информации на экране.

Программа подразумевает использование на различных устройствах, поэтому для удобного использования в коде прописано расположение окон относительно друг друга. Это обеспечивает качественную адаптацию под разные размеры экрана.

# **4 Руководство пользователя**

# **4.1 Назначение программы**

Программа "Таблица Шульте" предназначена для тренировки внимания и быстроты реакции. Она предоставляет пользователям возможность улучшить свои когнитивные навыки с помощью увлекательных игр, где необходимо находить заданные числа в определенном порядке в ограниченное время.

# **4.2 Запуск программы**

1. Убедитесь, что у вас установлен Python и откройте терминал в соответствующей программе.

2. Сохраните код программы в файл с расширением «.py», например, shulte.py. Обратите внимание, что для выполнения файлов зачастую недопустимо появление кириллицы в пути к расположению файла;

3. Откройте терминал или командную строку;

4. Перейдите в каталог, где сохранен файл программы;

5. Запустите программу с помощью команды «python (имя файла)».

# **4.3 Выполнение основных функций**

Программа запрашивает размеры таблицы Шульте, затем создает окно с игрой, а также информационное окно, где отображается номер ячейки, которую нужно нажать и время, затраченное на прохождение игры.

При нажатии последовательных ячеек, начиная с 1, программа ведет отсчет времени и обновляет номер нынешней ячейки. После нажатия ячейки её номер пропадает, и она перестает быть кликабельной, чтобы не отвлекать пользователя от поиска новых чисел.

# **4.4 Завершение программы**

После нажатия последней ячейки время останавливается, и программа предлагает выйти. Для повторного использования программы требуется перезапустить.

# **4.5 Пример работы с программой**

При запуске программы пользователю открывается окно ввода для указания размеров таблицы. Например, вводятся два числа: ширина 5; высота 5. После ввода данных и нажатия на кнопку "Подтвердить", окно закрывается.

После получения размеров, программа генерирует таблицу 5x5 с числами от 1 до 25 в случайном порядке. На экране также будет отображаться таймер, который начнет отсчет (Рисунок 11).

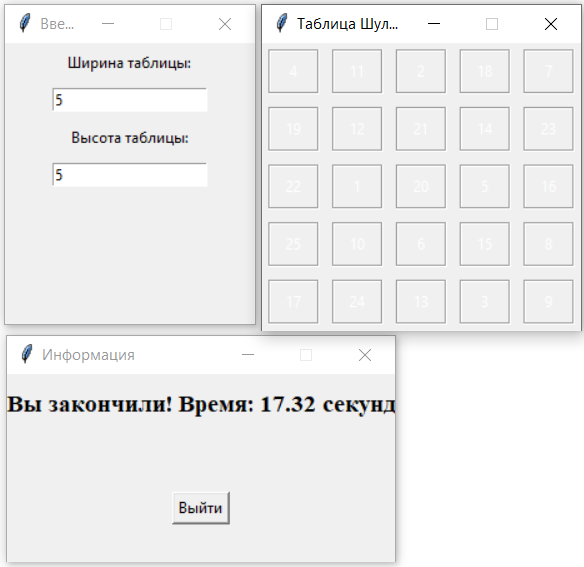


Рисунок 10 – Пример выполнения программы

Пользователь начинает игру, кликнув по числу 1. Если нажатие верное, число пропадает, и появляется следующее ожидаемое число в информационном окне. Дальше пользователь нажимает 2, затем 3 и так далее, заканчивая на числе 25. После нахождения всех чисел, программа запишет время, за которое пользователь прошел игру (Рисунок 10). Например, если пользователь затратил 1 минуту и 30 секунд, появится сообщение: "Вы закончили! Время 17,32 секунд". После этого программа предложит пользователю закрыть окно с результатами, и он сможет выйти из приложения.

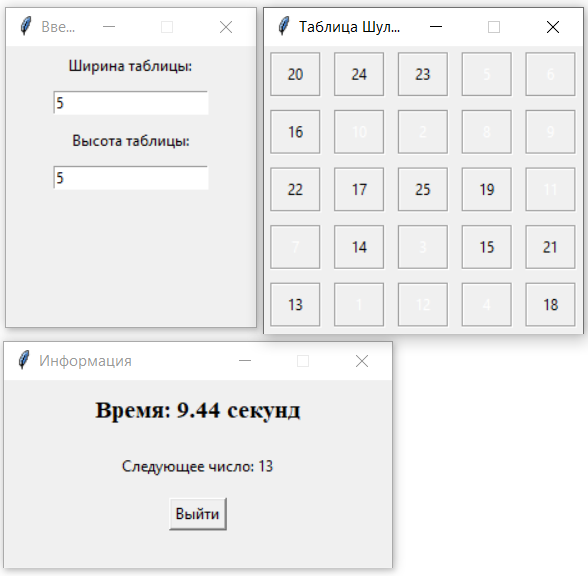


Рисунок 11 – Отображение времени и следующего числа ячейки

# Заключение

В данной курсовой работе представлена разработка приложения "Таблица Шульте", что включает в себя описание алгоритмов, реализацию и тестирование программы, а также инструкции для пользователей. Результаты проделанной работы могут использоваться для дальнейшего изучения техник тренировки памяти и развития когнитивных навыков. Также у продукта имеется потенциал к улучшению и увеличению функций.

# Библиографический список

1. Mathcha [Электронный ресурс] // Онлайн WYSIWYG редактор: https://www.mathcha.io/− Дата обращения 20.12.2024

2. Каталог статей: Tkinter // FullStacker [Электронный ресурс] // Онлайн блог о программировании: https://fullstacker.ru/search/?s=tkinter − Дата обращения 5.11.2023

3. Руководство по Tkinter // Metanit [Электронный ресурс] // Онлайн блог о программировании: https://metanit.com/python/tkinter/− Дата обращения 6.12.2024

4. Draw.io [Электронный ресурс] // Облачная платформа для создания визуального контента: https://www.drawio.com/ − Дата обращения 25.12.2024