## 4.3. Листинги

|  |
| --- |
| Загружаем проект бронирования мест  def book\_seat(seats):  seat\_name = input("Введите место для бронирования (от Б1 до Б9): ")  try:  # Находим индекс места, которое пользователь хочет забронировать  i = seats.index((seat\_name, 'свободно'))  # Отмечаем место как забронированное  seats[i] = (seat\_name, 'забронировано')  print(f"Место '{seat\_name}' успешно забронировано.")  except ValueError:  print(f"Место '{seat\_name}' уже забронировано или не существует.")  # Инициализация списка мест с использованием спискового включения  seats = [(f"Б{i}", 'свободно') for i in range(1, 10)]  print(seats)  # Запуск функции бронирования с возможностью повторного ввода  while True:  book\_seat(seats)  booking = input("Хотите забронировать еще одно место? (да/нет): ")  if booking.lower() != 'да':  break  # Отображение итогового состояния бронирования  print("Итоговое состояние бронирования мест:")  for seat in seats:  print(f"{seat[0]}: {seat[1]}") |
| Знакомство с enumerate() Используем не кортеж а словарь  def book\_seat(seats):  s = input("Введите место для бронирования (от Б1 до Б9): ")  if seats[s] == 'свободно': #Выдаст ошибку если ввести неправильное название места Надо добавить обработку исключения  seats[s] = 'забронировано'  print(f"Место '{s}' успешно забронировано.")  else:  print(f"Место '{s}' уже забронировано или не существует.")  # Инициализация словаря мест  seats = {f"Б{i}": 'свободно' for i in range(1, 10)}  # Запуск функции бронирования с возможностью повторного ввода  while True:  book\_seat(seats)  booking = input("Хотите забр. еще одно место? (да/нет): ")  if booking.lower() != 'да':  break  # Отображение итогового состояния бронирования с использованием enumerate  print("Итоговое состояние бронирования мест:")  for i, (seat, status) in enumerate(seats.items()):  print(f"{i + 1}. {seat}: {status}") |
| Обработка исключений  def book\_seat(seats):  seat\_name = input("Введите место (от Б1 до Б9): ")  try:  if seats[seat\_name] == 'свободно':  seats[seat\_name] = 'забронировано'  print(f"Место '{seat\_name}' успешно забронировано.")  else:  print(f"Место '{seat\_name}' уже забронировано или не существует.")  except KeyError:  print(f"Место '{seat\_name}' не существует.") |
| Программа с интерфейсом  from tkinter import \*  from tkinter import messagebox as mb  def book\_seat():  s = seat\_entry.get().upper()  try:  if seats[s] == 'свободно':  seats[s] = 'забронировано'  update\_canvas()  mb.showinfo("Успех", f"Место '{s}' успешно забронировано.")  else:  mb.showerror("Ошибка", f"Место '{s}' уже забронировано.")  except KeyError:  mb.showerror("Ошибка", f"Место '{s}' не существует.")  def update\_canvas():  canvas.delete("all")  for i, (seat, status) in enumerate(seats.items()):  x = i \* 40 + 20 # Уменьшаем шаг и начальное смещение  y = 20 # Фиксированное положение по оси Y для одного ряда  color = "green" if status == 'свободно' else "red"  canvas.create\_rectangle(x, y, x + 30, y + 30, fill=color) # Уменьшаем размер квадратов  canvas.create\_text(x + 15, y + 15, text=seat) # Центрируем текст в квадрате  window = Tk()  window.title("Бронирование мест")  window.geometry("400x200")  canvas = Canvas(width=400, height=80)  canvas.pack(pady=10)  # Инициализация мест  seats = {f"Б{i}": 'свободно' for i in range(1, 10)}  update\_canvas()  seat\_entry = Entry()  seat\_entry.pack(pady=10)  Button(text="Забронировать место", command=book\_seat).pack(pady=10)  window.mainloop() |
| Бронируем нажатием Enter на клавиатуре.  def book\_seat(event=None):  . . .  seat\_entry = Entry()  seat\_entry.pack(pady=10)  seat\_entry.focus() # Устанавливаем фокус на поле ввода  # Привязываем событие нажатия Enter к функции book\_seat  seat\_entry.bind("<Return>", book\_seat) |
| Код проекта  #Проект “Бронирование мест”.  from tkinter import \*  from tkinter import messagebox as mb  def book\_seat(event=None):  s = seat\_entry.get()  try:  if seats[s] == "свободно":  seats[s] = "забронировано"  update\_canvas()  mb.showinfo("Успех", f"Место {s} успешно забронировано.")  else:  mb.showinfo("Ошибка", f"Место {s} уже забронировано или не существует.")  except KeyError:  mb.showinfo("Ошибка", f"Место {s} не существует.")  def update\_canvas():  canvas.delete("all")  for i, (seat, status) in enumerate(seats.items()):  x = i \* 40 + 20  y = 20  color = "green" if status == "свободно" else "red"  canvas.create\_rectangle(x, y, x+30, y+30, fill=color)  canvas.create\_text(x+15, y+15, text=seat)  window = Tk()  window.title("Бронирование мест")  window.geometry("400x200")  canvas = Canvas(width=400, height=60)  canvas.pack(pady=10)  seats = {f"Б{i}": "свободно" for i in range(1, 10)}  update\_canvas()  seat\_entry = Entry()  seat\_entry.pack(pady=10)  seat\_entry.focus()  seat\_entry.bind("<Return>", book\_seat)  Button(text="Забронировать место", command=book\_seat).pack(pady=10)  window.mainloop() |
| Добавляем отмену бронирования  from tkinter import \*  from tkinter import messagebox as mb  def book\_seat(event=None):  s = seat\_entry.get()  try:  if seats[s] == "свободно":  seats[s] = "забронировано"  update\_canvas()  mb.showinfo("Успех", f"Место {s} успешно забронировано.")  else:  mb.showerror("Ошибка", f"Место {s} уже забронировано.")  except KeyError:  mb.showerror("Ошибка", f"Место {s} не существует.")  def cancel\_booking(event=None):  s = cancel\_seat\_entry.get()  try:  if seats[s] == "забронировано":  seats[s] = "свободно"  update\_canvas()  mb.showinfo("Успех", f"Бронь на месте {s} успешно отменена.")  else:  mb.showerror("Ошибка", f"Место {s} не забронировано или не существует.")  except KeyError:  mb.showerror("Ошибка", f"Место {s} не существует.")  def update\_canvas():  canvas.delete("all")  for i, (seat, status) in enumerate(seats.items()):  x = i \* 40 + 20  y = 20  color = "green" if status == "свободно" else "red"  canvas.create\_rectangle(x, y, x+30, y+30, fill=color)  canvas.create\_text(x+15, y+15, text=seat)  window = Tk()  window.title("Бронирование мест")  window.geometry("400x250")  canvas = Canvas(width=400, height=60)  canvas.pack(pady=10)  seats = {f"Б{i}": "свободно" for i in range(1, 10)}  update\_canvas()  seat\_entry = Entry()  seat\_entry.pack(pady=5)  seat\_entry.focus()  seat\_entry.bind("<Return>", book\_seat)  Button(text="Забронировать место", command=book\_seat).pack(pady=5)  # Добавление поля ввода и кнопки для отмены бронирования  cancel\_seat\_entry = Entry()  cancel\_seat\_entry.pack(pady=5)  cancel\_seat\_entry.bind("<Return>", cancel\_booking)  Button(text="Отменить бронь", command=cancel\_booking).pack(pady=5)  window.mainloop() |