

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Лабораторная работа №13

по дисциплине: ООП

тема: **«Знакомство с библиотеками языка Python. PyQt.»**

Выполнил: студент группы ПВ-233
Мороз Роман Алексеевич

Проверили:
Морозов Данила Александрович

Белгород 2025

Знакомство с библиотеками языка Python. PyQt.

Цель работы: приобретение практических навыков создания приложений на языке Python, QT приложения.

Задание:

Для выполнения лабораторной работы требуется установить интерпретатор Python версии 3.6+. Выполнить написание программы-сценария в соответствии с вариантом задания (табл. 1). Провести тестирование. Оформить отчет.

Варианты задания:

1	QT-paint
2	QT-судоку
3	QT-TaskManager
4	QT-Библиотека (с поиском по тексту)
5	QT-Плагиат
6	QT-Электронные кошелек
7	QT-Адресная книга (контакты, с поиском)
8	QT-Дневник спортсмена
9	QT-Книга рецептов
10	QT-Фитнес инструктор (составление тренировок)

```
class Recipe:
    def __init__(self, title, ingredients, instructions, notes):
        self.title = title
        self.ingredients = ingredients
        self.instructions = instructions
        self.notes = notes

class RecipeBook:
    def __init__(self):
        self.recipes = []

    def add(self, recipe):
        self.recipes.append(recipe)

    def update(self, index, title, ingredients, instructions, notes):
        if 0 <= index < len(self.recipes):
            recipe = self.recipes[index]
            recipe.title = title
            recipe.ingredients = ingredients
```

```

        recipe.instructions = instructions
        recipe.notes = notes

    def delete(self, index):
        if 0 <= index < len(self.recipes):
            del self.recipes[index]

    def all(self):
        return self.recipes

```

```

from kivy.uix.boxlayout import BoxLayout
from kivy.uix.label import Label
from kivy.uix.button import Button
from kivy.uix.textinput import TextInput
from kivy.uix.popup import Popup
from kivy.uix.scrollview import ScrollView
from kivy.uix.gridlayout import GridLayout
from models import Recipe, RecipeBook

class RecipeAppUI(BoxLayout):
    def __init__(self, **kwargs):
        super().__init__(orientation='vertical', spacing=10, padding=20,
**kwargs)

        self.book = RecipeBook()

        self.add_widget(Label(text='Recipe Book', font_size=28, color=[0,
0, 0, 1], size_hint=(1, 0.1)))

        self.title_input = TextInput(hint_text='Dish name',
size_hint_y=None, height=40)
        self.ingredients_input = TextInput(hint_text='Ingredients
(comma-separated)', size_hint_y=None, height=80)
        self.instructions_input = TextInput(hint_text='Preparation method',
size_hint_y=None, height=100)
        self.notes_input = TextInput(hint_text='Additional notes',
size_hint_y=None, height=60)

        self.add_widget(self.title_input)
        self.add_widget(self.ingredients_input)

```

```

        self.add_widget(self.instructions_input)
        self.add_widget(self.notes_input)

        self.add_button = Button(text='Add Recipe', background_color=[0.2,
0.6, 0.86, 1], size_hint_y=None, height=50)
        self.add_button.bind(on_press=self.add_recipe)
        self.add_widget(self.add_button)

        self.recipe_layout = GridLayout(cols=1, spacing=10,
size_hint_y=None)

self.recipe_layout.bind(minimum_height=self.recipe_layout.setter('height')
)

        scroll = ScrollView()
        scroll.add_widget(self.recipe_layout)
        self.add_widget(scroll)

        self.refresh()

    def add_recipe(self, _):
        title = self.title_input.text.strip()
        ingredients = self.ingredients_input.text.strip()
        instructions = self.instructions_input.text.strip()
        notes = self.notes_input.text.strip()

        if title and ingredients and instructions:
            self.book.add(Recipe(title, ingredients, instructions, notes))
            self.title_input.text = ''
            self.ingredients_input.text = ''
            self.instructions_input.text = ''
            self.notes_input.text = ''
            self.refresh()

    def refresh(self):
        self.recipe_layout.clear_widgets()
        for i, recipe in enumerate(self.book.all()):
            card = BoxLayout(orientation='vertical', padding=10, spacing=5,
size_hint_y=None, height=120)

```

```

        label = Label(text=f"[b]{recipe.title}[/b]", markup=True,
color=[0, 0, 0, 1], size_hint_y=None, height=30)
        btns = BoxLayout(size_hint_y=None, height=30)

        view_btn = Button(text='View')
        edit_btn = Button(text='Edit')
        delete_btn = Button(text='Delete', background_color=[0.9, 0.3,
0.3, 1])

        view_btn.bind(on_press=lambda inst, idx=i:
self.view_recipe(idx))
        edit_btn.bind(on_press=lambda inst, idx=i:
self.edit_recipe(idx))
        delete_btn.bind(on_press=lambda inst, idx=i:
self.delete_recipe(idx))

        btns.add_widget(view_btn)
        btns.add_widget(edit_btn)
        btns.add_widget(delete_btn)

        card.add_widget(label)
        card.add_widget(btns)
        self.recipe_layout.add_widget(card)

    def view_recipe(self, index):
        r = self.book.all()[index]
        content = BoxLayout(orientation='vertical', padding=10)
        content.add_widget(Label(text=f"[b]Ingredients: [/b]
{r.ingredients}", markup=True))
        content.add_widget(Label(text=f"[b]Instructions: [/b]
{r.instructions}", markup=True))
        content.add_widget(Label(text=f"[b]Notes: [/b] {r.notes}",
markup=True))

        popup = Popup(title=r.title, content=content, size_hint=(0.9, 0.7))
        popup.open()

    def edit_recipe(self, index):
        r = self.book.all()[index]
        layout = BoxLayout(orientation='vertical', spacing=10, padding=10)

```

```

        title_input = TextInput(text=r.title)
        ingredients_input = TextInput(text=r.ingredients)
        instructions_input = TextInput(text=r.instructions)
        notes_input = TextInput(text=r.notes)
        save_btn = Button(text='Save', background_color=[0.2, 0.6, 0.86,
1]))

        layout.add_widget(title_input)
        layout.add_widget(ingredients_input)
        layout.add_widget(instructions_input)
        layout.add_widget(notes_input)
        layout.add_widget(save_btn)

        popup = Popup(title='Edit Recipe', content=layout, size_hint=(0.9,
0.8))

        def save(_):
            self.book.update(index, title_input.text,
ingredients_input.text, instructions_input.text, notes_input.text)
            popup.dismiss()
            self.refresh()

        save_btn.bind(on_press=save)
        popup.open()

    def delete_recipe(self, index):
        self.book.delete(index)
        self.refresh()

```

```

from kivy.app import App
from kivy.core.window import Window
from ui.recipe_ui import RecipeAppUI

Window.clearcolor = (1, 1, 1, 1)

class RecipeApp(App):
    def build(self):

```

```
        return RecipeAppUI()

if __name__ == '__main__':
    RecipeApp().run()
```

Вывод: приобрели практические навыки создания приложений на языке Python, QT приложения.