Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №5  
По дисциплине: «Проектирование баз знаний»  
Тема: Создание приложения для работы с БД и организация пользовательского интерфейса.

Выполнил:  
студент 3 курса  
группы ИИ-21(I)  
Пучинский А.А.

Проверил:  
Савонюк В. А.

Брест 2023

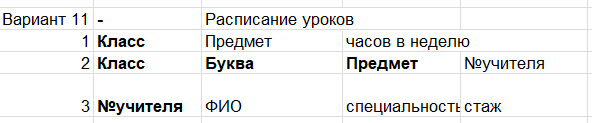
Цель: получить навыки разработки приложений БД и организации пользовательского интерфейса.

**Вариант 11**

Задание.

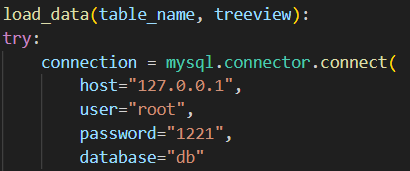
1. Изучить материал, приведенный в “К лабораторной работе 5.doc”.

2. На основании логической модели (в соответствии с вариантом Лабораторной работы №1) создать приложение и организовать пользовательский интерфейс, используя соответствующие элементы и методы.



**Ход выполнения лабораторной работы:**

**Интегрированные данные с MSQL Server:**



**Код программы:**

import tkinter as tk

from tkinter import ttk

import mysql.connector

def load\_data(table\_name, treeview):

try:

connection = mysql.connector.connect(

host="127.0.0.1",

user="root",

password="1221",

database="db"

)

cursor = connection.cursor()

query = f"SELECT \* FROM {table\_name}"

cursor.execute(query)

data = cursor.fetchall()

connection.close()

for i in treeview.get\_children():

treeview.delete(i)

for row in data:

treeview.insert("", "end", values=row)

except mysql.connector.Error as err:

print(f"Ошибка при загрузке данных: {err}")

def delete\_data(table\_name, tree):

selected\_item = tree.selection()

if selected\_item:

values = [tree.item(selected\_item)['values'][0]]

if table\_name == "Class":

query = f"DELETE FROM {table\_name} WHERE ClassID = %s"

elif table\_name == "ClassSubject":

query = f"DELETE FROM {table\_name} WHERE ClassID = %s"

elif table\_name == "Teacher":

query = f"DELETE FROM {table\_name} WHERE TeacherID = %s"

execute\_query(query, values)

load\_data(table\_name, tree)

def edit\_data(table\_name, tree, entry\_vars, labels):

selected\_item = tree.selection()

if selected\_item:

values = [entry\_vars[i].get() for i in range(len(entry\_vars))]

values.append(tree.item(selected\_item)['values'][0])

query = f"UPDATE {table\_name} SET {', '.join([f'{col} = %s' for col in labels])} WHERE {table\_name}ID = %s"

execute\_query(query, values)

load\_data(table\_name, tree)

def add\_data(table\_name, tree, entry\_vars, labels):

values = [entry\_vars[i].get() for i in range(len(entry\_vars))]

query = f"INSERT INTO {table\_name} ({', '.join(labels)}) VALUES ({', '.join(['%s' for \_ in range(len(entry\_vars))])})"

execute\_query(query, values)

load\_data(table\_name, tree)

def save\_data(table\_name, treeview):

load\_data(table\_name, treeview)

def execute\_query(query, values=()):

connection = mysql.connector.connect(

host="127.0.0.1",

user="root",

password="1221",

database="db"

)

cursor = connection.cursor()

cursor.execute(query, values)

connection.commit()

connection.close()

root = tk.Tk()

root.title("Расписание уроков")

notebook = ttk.Notebook(root)

notebook.pack(fill="both", expand=True)

tab1 = ttk.Frame(notebook)

notebook.add(tab1, text="Class")

tab2 = ttk.Frame(notebook)

notebook.add(tab2, text="ClassSubject")

tab3 = ttk.Frame(notebook)

notebook.add(tab3, text="Teacher")

labels\_class = ["ClassID", "Subject", "HoursPerWeek"]

entries\_class = []

entry\_var\_class = []

for label in labels\_class:

i = labels\_class.index(label)

label\_widget = tk.Label(tab1, text=label)

label\_widget.grid(row=i, column=0, padx=10, pady=5)

entry\_var\_class.append(tk.StringVar())

entry = tk.Entry(tab1, textvariable=entry\_var\_class[i])

entry.grid(row=i, column=1, padx=10, pady=5)

entries\_class.append(entry)

treeview1 = ttk.Treeview(tab1, columns=labels\_class, show="headings")

for label in labels\_class:

treeview1.heading(label, text=label)

treeview1.column(label, width=100)

treeview1.grid(row=len(labels\_class), column=0, columnspan=2, padx=10, pady=5)

load\_data("Class", treeview1)

labels\_class\_subject = ["ClassID", "ClassLetter", "Subject", "TeacherID"]

entries\_class\_subject = []

entry\_var\_class\_subject = []

for label in labels\_class\_subject:

i = labels\_class\_subject.index(label)

label\_widget = tk.Label(tab2, text=label)

label\_widget.grid(row=i, column=0, padx=10, pady=5)

entry\_var\_class\_subject.append(tk.StringVar())

entry = tk.Entry(tab2, textvariable=entry\_var\_class\_subject[i])

entry.grid(row=i, column=1, padx=10, pady=5)

entries\_class\_subject.append(entry)

treeview2 = ttk.Treeview(tab2, columns=labels\_class\_subject, show="headings")

for label in labels\_class\_subject:

treeview2.heading(label, text=label)

treeview2.column(label, width=100)

treeview2.grid(row=len(labels\_class\_subject), column=0, columnspan=2, padx=10, pady=5)

load\_data("ClassSubject", treeview2)

labels\_teacher = ["TeacherID", "TeacherName", "Specialization", "Experience"]

entries\_teacher = []

entry\_var\_teacher = []

for label in labels\_teacher:

i = labels\_teacher.index(label)

label\_widget = tk.Label(tab3, text=label)

label\_widget.grid(row=i, column=0, padx=10, pady=5)

entry\_var\_teacher.append(tk.StringVar())

entry = tk.Entry(tab3, textvariable=entry\_var\_teacher[i])

entry.grid(row=i, column=1, padx=10, pady=5)

entries\_teacher.append(entry)

treeview3 = ttk.Treeview(tab3, columns=labels\_teacher, show="headings")

for label in labels\_teacher:

treeview3.heading(label, text=label)

treeview3.column(label, width=100)

treeview3.grid(row=len(labels\_teacher), column=0, columnspan=2, padx=10, pady=5)

load\_data("Teacher", treeview3)

delete\_button1 = tk.Button(tab1, text="Удалить", command=lambda: delete\_data("Class", treeview1))

delete\_button1.grid(row=len(labels\_class) + 1, column=0, padx=10, pady=5)

edit\_button1 = tk.Button(tab1, text="Изменить", command=lambda: edit\_data("Class", treeview1, entry\_var\_class, labels\_class))

edit\_button1.grid(row=len(labels\_class) + 1, column=1, padx=10, pady=5)

add\_button1 = tk.Button(tab1, text="Добавить", command=lambda: add\_data("Class", treeview1, entry\_var\_class, labels\_class))

add\_button1.grid(row=len(labels\_class) + 2, column=0, padx=10, pady=5)

save\_button1 = tk.Button(tab1, text="Сохранить", command=lambda: save\_data("Class", treeview1))

save\_button1.grid(row=len(labels\_class) + 2, column=1, padx=10, pady=5)

delete\_button2 = tk.Button(tab2, text="Удалить", command=lambda: delete\_data("ClassSubject", treeview2))

delete\_button2.grid(row=len(labels\_class\_subject) + 1, column=0, padx=10, pady=5)

edit\_button2 = tk.Button(tab2, text="Изменить", command=lambda: edit\_data("ClassSubject", treeview2, entry\_var\_class\_subject, labels\_class\_subject))

edit\_button2.grid(row=len(labels\_class\_subject) + 1, column=1, padx=10, pady=5)

add\_button2 = tk.Button(tab2, text="Добавить", command=lambda: add\_data("ClassSubject", treeview2, entry\_var\_class\_subject, labels\_class\_subject))

add\_button2.grid(row=len(labels\_class\_subject) + 2, column=0, padx=10, pady=5)

save\_button2 = tk.Button(tab2, text="Сохранить", command=lambda: save\_data("ClassSubject", treeview2))

save\_button2.grid(row=len(labels\_class\_subject) + 2, column=1, padx=10, pady=5)

delete\_button3 = tk.Button(tab3, text="Удалить", command=lambda: delete\_data("Teacher", treeview3))

delete\_button3.grid(row=len(labels\_teacher) + 1, column=0, padx=10, pady=5)

edit\_button3 = tk.Button(tab3, text="Изменить", command=lambda: edit\_data("Teacher", treeview3, entry\_var\_teacher, labels\_teacher))

edit\_button3.grid(row=len(labels\_teacher) + 1, column=1, padx=10, pady=5)

add\_button3 = tk.Button(tab3, text="Добавить", command=lambda: add\_data("Teacher", treeview3, entry\_var\_teacher, labels\_teacher))

add\_button3.grid(row=len(labels\_teacher) + 2, column=0, padx=10, pady=5)

save\_button3 = tk.Button(tab3, text="Сохранить", command=lambda: save\_data("Teacher", treeview3))

save\_button3.grid(row=len(labels\_teacher) + 2, column=1, padx=10, pady=5)

root.mainloop()

**Результат программы**:

Таблица 1:

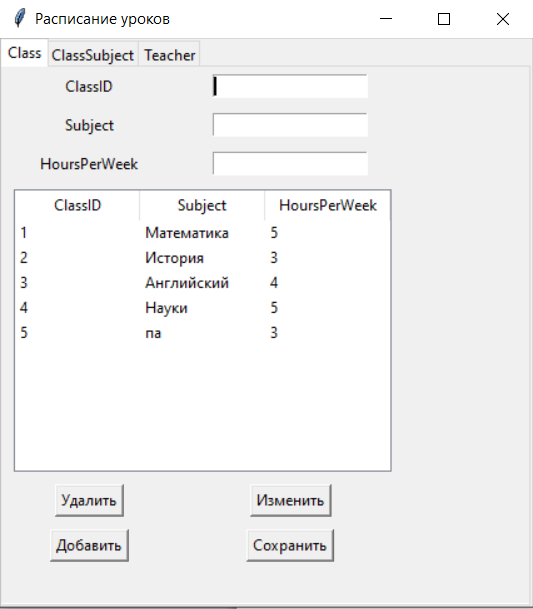


Таблица 2:

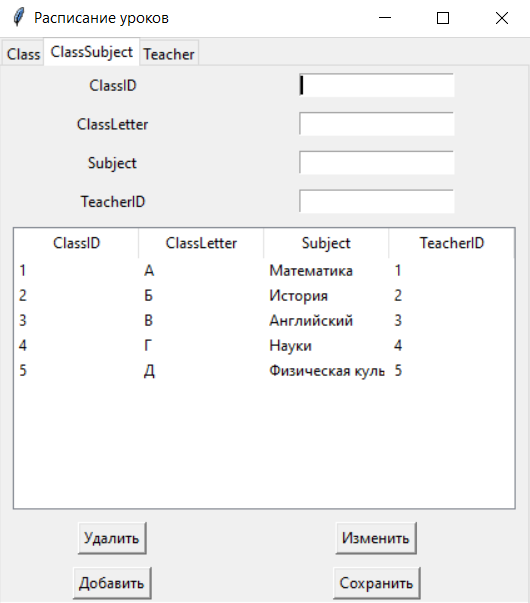
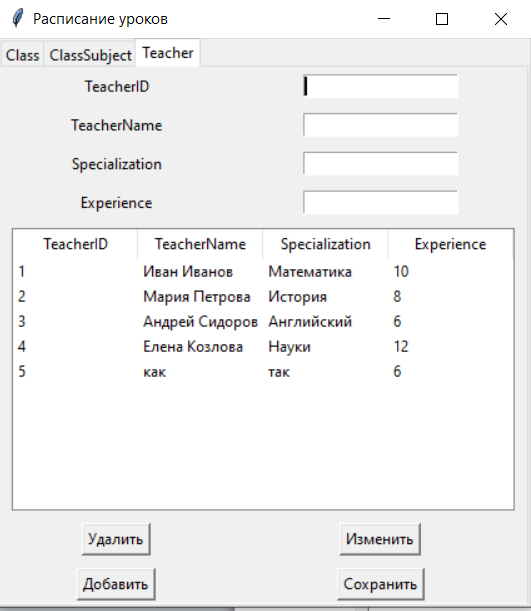


Таблица 3:



Вывод: получил навыки разработки приложений БД и организации пользовательского интерфейса.