Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: «Проектирование баз знаний»

Тема: «Создание приложения для работы с БД и организация пользовательского интерфейса»

Выполнил:

Студент 3 курса

Группы ИИ-21(1)

Пучинский А.А.

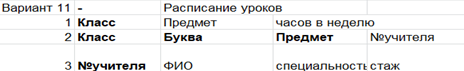
Проверил:

Савонюк В.А.

Брест 2023

**Цель работы:** создание приложения для работы с БД организация пользовательского интерфейса: табличное представление и отчеты.

**Вариант 11**



**Ход работы:**

**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()

if table\_name == "Teacher":

query = f"SELECT \* FROM {table\_name} ORDER BY TeacherName ASC"

else:

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 search\_data(table\_name, treeview, search\_var):

try:

connection = mysql.connector.connect(

host="127.0.0.1",

user="root",

password="1221",

database="db"

)

cursor = connection.cursor()

if table\_name == "Teacher":

query = f"SELECT \* FROM {table\_name} WHERE TeacherName LIKE %s"

else:

query = f"SELECT \* FROM {table\_name} WHERE Subject LIKE %s"

cursor.execute(query, (f"%{search\_var.get()}%",))

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 поиске данных: {err}")

def sort\_data(table\_name, treeview, sort\_var):

try:

connection = mysql.connector.connect(

host="127.0.0.1",

user="root",

password="1221",

database="db"

)

cursor = connection.cursor()

if table\_name == "Teacher":

query = f"SELECT \* FROM {table\_name} ORDER BY TeacherName {sort\_var.get()}"

else:

query = f"SELECT \* FROM {table\_name} ORDER BY Subject {sort\_var.get()}"

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 сортировке данных: {err}")

def generate\_report(table\_name, search\_var, sort\_var):

search\_criteria = search\_var.get()

sort\_criteria = sort\_var.get()

report = f"Отчет\nПоиск: {search\_criteria}\nСортировка: {sort\_criteria}\n"

treeview = None

columns = [] # Добавим переменную для хранения столбцов отчета

if table\_name == "Class":

report += "Таблица: Class\n\n"

treeview = treeview1

columns = ["ClassID", "Subject", "HoursPerWeek"] # Столбцы для отчета

elif table\_name == "ClassSubject":

report += "Таблица: ClassSubject\n\n"

treeview = treeview2

columns = ["ClassID", "ClassLetter", "Subject", "TeacherID"] # Столбцы для отчета

elif table\_name == "Teacher":

report += "Таблица: Teacher\n\n"

treeview = treeview3

columns = ["TeacherID", "TeacherName", "Specialization", "Experience"] # Столбцы для отчет

# Добавляем заголовки столбцов

report += "\t".join(columns) + "\n"

for i in treeview.get\_children():

item = treeview.item(i)['values']

report += "\t".join(map(str, item)) + "\n"

# Создание окна для отчета

report\_window = tk.Toplevel(root)

report\_window.title("Отчет")

# Создание текстового виджета для отображения отчета

report\_text = tk.Text(report\_window, wrap=tk.WORD, height=25, width=55)

report\_text.pack()

report\_text.insert("1.0", report)

report\_text.config(state=tk.DISABLED)

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")

# Элементы интерфейса и функции для вкладки "Class"

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)

search\_var1 = tk.StringVar()

search\_label1 = tk.Label(tab1, text="Поиск:")

search\_label1.grid(row=len(labels\_class) + 3, column=0, padx=10, pady=5)

search\_entry1 = tk.Entry(tab1, textvariable=search\_var1)

search\_entry1.grid(row=len(labels\_class) + 3, column=1, padx=10, pady=5)

search\_button1 = tk.Button(tab1, text="Искать", command=lambda: search\_data("Class", treeview1, search\_var1))

search\_button1.grid(row=len(labels\_class) + 4, column=0, padx=10, pady=5)

sort\_var1 = tk.StringVar()

sort\_var1.set("ASC")

sort\_label1 = tk.Label(tab1, text="Сортировка:")

sort\_label1.grid(row=len(labels\_class) + 5, column=0, padx=10, pady=5)

sort\_option1 = tk.OptionMenu(tab1, sort\_var1, "ASC", "DESC")

sort\_option1.grid(row=len(labels\_class) + 5, column=1, padx=10, pady=5)

sort\_button1 = tk.Button(tab1, text="Сортировать", command=lambda: sort\_data("Class", treeview1, sort\_var1))

sort\_button1.grid(row=len(labels\_class) + 6, column=0, padx=10, pady=5)

report\_button1 = tk.Button(tab1, text="Создать отчет", command=lambda: generate\_report("Class", search\_var1, sort\_var1))

report\_button1.grid(row=len(labels\_class) + 6, column=1, padx=10, pady=5)

# Элементы интерфейса и функции для вкладки "ClassSubject"

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)

search\_var2 = tk.StringVar()

search\_label2 = tk.Label(tab2, text="Поиск:")

search\_label2.grid(row=len(labels\_class\_subject) + 3, column=0, padx=10, pady=5)

search\_entry2 = tk.Entry(tab2, textvariable=search\_var2)

search\_entry2.grid(row=len(labels\_class\_subject) + 3, column=1, padx=10, pady=5)

search\_button2 = tk.Button(tab2, text="Искать", command=lambda: search\_data("ClassSubject", treeview2, search\_var2))

search\_button2.grid(row=len(labels\_class\_subject) + 4, column=0, padx=10, pady=5)

sort\_var2 = tk.StringVar()

sort\_var2.set("ASC")

sort\_label2 = tk.Label(tab2, text="Сортировка:")

sort\_label2.grid(row=len(labels\_class\_subject) + 5, column=0, padx=10, pady=5)

sort\_option2 = tk.OptionMenu(tab2, sort\_var2, "ASC", "DESC")

sort\_option2.grid(row=len(labels\_class\_subject) + 5, column=1, padx=10, pady=5)

sort\_button2 = tk.Button(tab2, text="Сортировать", command=lambda: sort\_data("ClassSubject", treeview2, sort\_var2))

sort\_button2.grid(row=len(labels\_class\_subject) + 6, column=0, padx=10, pady=5)

report\_button2 = tk.Button(tab2, text="Создать отчет", command=lambda: generate\_report("ClassSubject", search\_var2, sort\_var2))

report\_button2.grid(row=len(labels\_class\_subject) + 6, column=1, padx=10, pady=5)

# Элементы интерфейса и функции для вкладки "Teacher"

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)

enties\_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)

search\_var3 = tk.StringVar()

search\_label3 = tk.Label(tab3, text="Поиск:")

search\_label3.grid(row=len(labels\_teacher) + 3, column=0, padx=10, pady=5)

search\_entry3 = tk.Entry(tab3, textvariable=search\_var3)

search\_entry3.grid(row=len(labels\_teacher) + 3, column=1, padx=10, pady=5)

search\_button3 = tk.Button(tab3, text="Искать", command=lambda: search\_data("Teacher", treeview3, search\_var3))

search\_button3.grid(row=len(labels\_teacher) + 4, column=0, padx=10, pady=5)

sort\_var3 = tk.StringVar()

sort\_var3.set("ASC")

sort\_label3 = tk.Label(tab3, text="Сортировка:")

sort\_label3.grid(row=len(labels\_teacher) + 5, column=0, padx=10, pady=5)

sort\_option3 = tk.OptionMenu(tab3, sort\_var3, "ASC", "DESC")

sort\_option3.grid(row=len(labels\_teacher) + 5, column=1, padx=10, pady=5)

sort\_button3 = tk.Button(tab3, text="Сортировать", command=lambda: sort\_data("Teacher", treeview3, sort\_var3))

sort\_button3.grid(row=len(labels\_teacher) + 6, column=0, padx=10, pady=5)

report\_button3 = tk.Button(tab3, text="Создать отчет", command=lambda: generate\_report("Teacher", search\_var3, sort\_var3))

report\_button3.grid(row=len(labels\_teacher) + 6, column=1, padx=10, pady=5)

root.mainloop()

**Результат программы**:

Таблица 1:

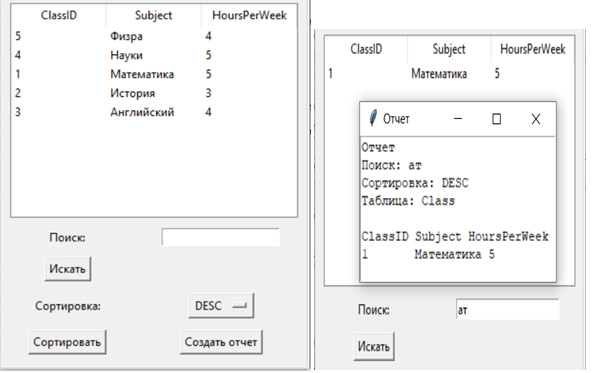


Таблица 2:

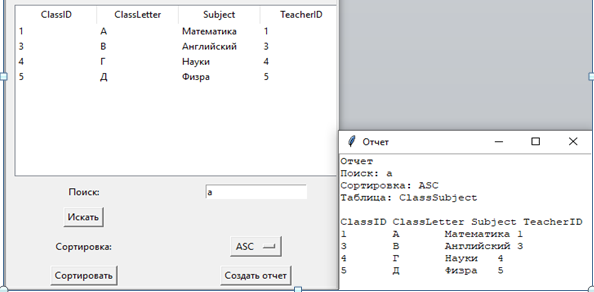
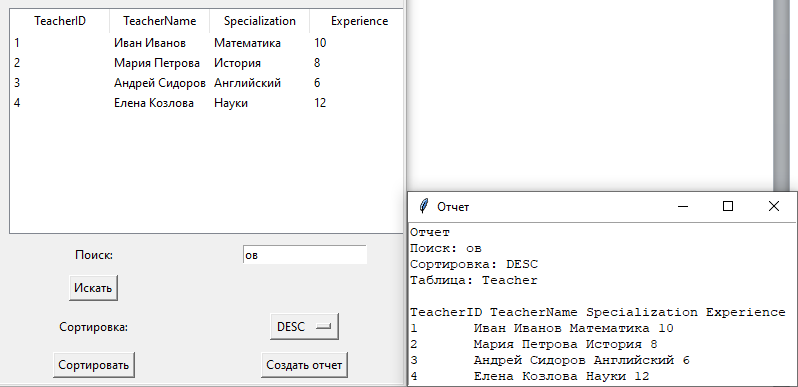


Таблица 3:



**Вывод:** в ходе данной лабораторной работы мной были получены навыки разработки приложений БД и навыки организации пользовательского интерфейса (табличное представление и отчеты).