

Министерство образования Республики Беларусь  
Учреждение образования  
“Брестский государственный технический университет”

Лабораторная работа №5

По дисциплине ООТПиСП за 5 семестр  
Тема: “Создание приложений баз данных”

**Выполнил:**

Студент группы ПО-6(1)  
3-го курса

Мартынович Даниил

**Проверил:**

Хацкевич М. В.

Брест 2022

**Цель:** отработка умений и навыков создания интерфейса базы данных.

## Задание

Напишите программу «Учет выплаты заработной платы». В программе использовать две таблицы: fio (поля id, family, name, otch) и vid (поля id, vid, month, summa). Создать запросы: вывод зар. платы за месяц по каждому работнику; вывод вида работ работника за месяц.

## Код программы

### Mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent)
, ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->pushButton_8->setFixedWidth(140);
    ui->pushButton_5->setFixedWidth(140);
    ui->pushButton_6->setFixedWidth(110);
    ui->pushButton_7->setFixedWidth(110);
    ui->pushButton_3->setFixedWidth(110);
    ui->pushButton_4->setFixedWidth(110);
    ui->tableView->setFixedWidth(530);
    ui->tableView_2->setFixedWidth(530);
    ui->tableWidget_4->setFixedWidth(400);
    ui->tableWidget_5->setFixedWidth(400);
    db=QSqlDatabase::addDatabase("SQLITE");

    db.setDatabaseName("./lab5.db"); if(!db.open()){
        QMessageBox::warning(this, "Внимание", "База данных не открыта!");
    }

    query1=new QSqlQuery(db);
    query1->exec("CREATE TABLE fio(id INT, family TEXT, name TEXT, otch TEXT);");
    model=new QSqlTableModel(this, db); model->setTable("fio"); model->select();
    ui->tableView->setModel(model);
    query2=new QSqlQuery(db);
    query2->exec("CREATE TABLE vid(id INT, vid TEXT, monht INT, summa INT);");
    model2=new QSqlTableModel(this, db);
    model2->setTable("vid");
    model2->select();
    ui->tableView->setModel(model);
    ui->tableView_2->setModel(model2);
}

MainWindow::~MainWindow() {
    delete ui;
}

void MainWindow::on_pushButton_3_clicked() {
    model->insertRow(model->rowCount());
}

void MainWindow::on_tableView_clicked(const QModelIndex &index) {
    row=index.row();
}

void MainWindow::on_pushButton_4_clicked() {
    model->removeRow(row);
    query2=new QSqlQuery(db);
    query2->exec("CREATE TABLE fio(id INT, day INT, month TEXT, year INT);");
    model=new QSqlTableModel(this, db);
    model->setTable("fio");
    model->select();
    ui->tableView->setModel(model);
}
```

```

void MainWindow::on_pushButton_5_clicked() {
    ui->tableWidget_4->setRowCount(0);
    if(ui->lineEdit->text().toInt() > 0 && ui->lineEdit->text().toInt() <= 12) {
        int countt = 0;
        for (int a = 0; a < model->rowCount(); ++a) {
            int money = 0;
            for(int i = 0; i < model2->rowCount(); ++i) {
                if(model2->index(i,2).data().toInt() == ui->lineEdit->text().toInt() &&
model->index(a,0).data().toInt() == model2->index(i,0).data().toInt()) {
                    money += model2->index(i,3).data().toInt();
                }
            }
            if(money!=0){
                ui->tableWidget_4->setRowCount(ui->tableWidget_4->rowCount() + 1);
                for (int t = countt; t < ui->tableWidget_4->rowCount(); ++t) {
                    QTableWidgetItem * item;
                    for (int j = 0; j < ui->tableWidget_4->columnCount(); ++j) {
                        item = new QTableWidgetItem;
                        if(j==0)
                            item->setText(model->index(a,1).data().toString() + "\n" + model-
>index(a,2).data().toString());
                        if(j==1) item->setText(QString::number(money));
                        ui->tableWidget_4->setItem(t, j, item);
                    }
                    countt++;
                }
            }
        }
    }
    else{
        QString f;
        int e=ui->lineEdit->text().toInt();
        if(ui->lineEdit->text()==""){
            return;
        }
        if(e%10==1){
            f="месяц";
        }
        else{
            if(e%10==2||e%10==3||e%10==4){
                f="месяца";
            }
            else{
                f="месяцев";
            }
        }
    }
    QMessageBox::about(this,"Ошибка","В году не может быть "+ui->lineEdit->text()+" "+
f);
    return;
}

void MainWindow::on_tableView_2_clicked(const QModelIndex &index) {
    row2=index.row();
}

void MainWindow::on_pushButton_6_clicked() {
    model2->insertRow(model2->rowCount());
}

void MainWindow::on_pushButton_7_clicked() {
    model2->removeRow(row2);
    query2=new QSqlQuery(db);
    query2->exec("CREATE TABLE vid(id INT, vid TEXT, month INT, summa INT);");
    model2=new QSqlTableModel(this, db); model2->setTable("vid"); model2->select();
    ui->tableView_2->setModel(model2);
}

```

```

void MainWindow::on_pushButton_8_clicked() {
    ui->tableWidget_5->setRowCount(0);
    if(ui->lineEdit_2->text().toInt() > 0 && ui->lineEdit_2->text().toInt() <= 12){
        int countt = 0;
        for(int a = 0; a < model->rowCount(); ++a){
            QString vid = "";
            for(int i = 0; i < model2->rowCount(); ++i){
                if(model2->index(i,2).data().toInt() == ui->lineEdit_2->text().toInt()
                    && model->index(a,0).data().toInt() == model2-
>index(i,0).data().toInt()){
                    vid += model2->index(i,1).data().toString()+" ";
                }
            }
            if(vid!=""){
                ui->tableWidget_5->setRowCount(ui->tableWidget_5->rowCount()+1);
                for (int t = countt; t < ui->tableWidget_5->rowCount(); ++t) {
                    QTableWidgetItem * item;
                    for(int j = 0; j < ui->tableWidget_5->columnCount(); ++j){
                        item = new QTableWidgetItem;
                        if(j == 0){
                            item->setText(model->index(a,1).data().toString() + " " + model-
>index(a,2).data().toString());
                        }
                        if(j == 1){
                            item->setText(vid);
                        }
                        ui->tableWidget_5->setItem(t, j, item);
                    }
                }
                countt++;
            }
        }
    }
    else{
        QString f;
        int e=ui->lineEdit_2->text().toInt();
        if(ui->lineEdit->text()==""){
            return;
        }
        if(e%10==1){
            f="месяц"; }
        else{
            if(e%10==2||e%10==3||e%10==4){
                f="месяца";
            }
            else{
                f="месяцев";
            }
        }
        QMessageBox::about(this,"Ошибка","В году не может быть "+ui->lineEdit_2->text()+"
"+ f);
        return;
    }
}

```

## Mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include <QSqlDatabase>
#include <QDebug>
#include <QSqlQuery>
#include <QSqlTableModel>
#include <QCloseEvent>

```

```

#include <QSqlRecord>
#include <string>
#include <iostream>
#include <string.h>
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; } QT_END_NAMESPACE
class MainWindow : public QMainWindow {
    Q_OBJECT
public:
    MainWindow(QWidget *parent = nullptr); ~MainWindow();
private slots:
    void on_pushButton_3_clicked();
    void on_tableView_clicked(const QModelIndex &index);
    void on_pushButton_4_clicked();
    void on_pushButton_5_clicked();
    void on_tableView_2_clicked(const QModelIndex &index);
    void on_pushButton_6_clicked();
    void on_pushButton_7_clicked();
    void on_pushButton_8_clicked();
private:
    Ui::MainWindow *ui;
    QSqlDatabase db;
    QSqlQuery *query1;
    QSqlQuery *query2;
    QSqlTableModel *model;
    QSqlTableModel *model2;
    int row, row2;
};
#endif // MAINWINDOW_H

```

