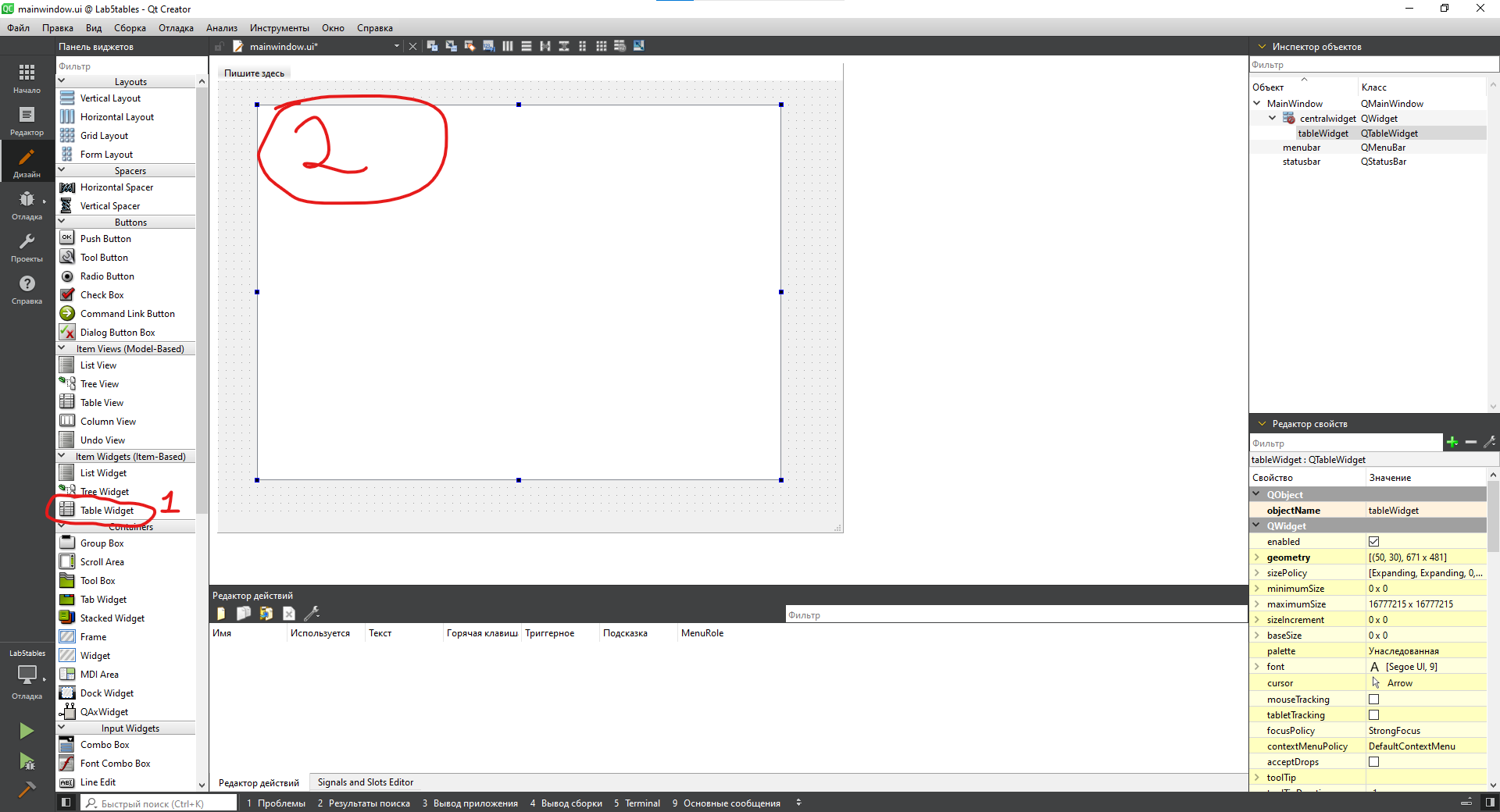
|  |  |
| --- | --- |
| ДИСЦИПЛИНА | Технологии индустриального программирования |
| ИНСТИТУТ | ИПТИП |
| КАФЕДРА | Индустриального программирования |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Методические указания к практическим занятиям по дисциплине |
| ПРЕПОДАВАТЕЛЬ | Александров Алексей Сергеевич |
| СЕМЕСТР | 3 семестр, 2024/2025 уч. год |

# Практическое занятие №15

## Динамические виджеты в Qt.

Применение контейнеров особенно эффективно при использовании динамических виджетов – в предыдущих практических работах все виджеты, с которыми пользователь может взаимодействовать были заложены в интерфейс заранее. Однако, при работе, например, с базами данных, данные могут добавляться, удаляться и меняться в процессе работы программы. Для работы с такими данными обычно используются таблицы. В Qt таблицы представлены виджетом TableWidget:



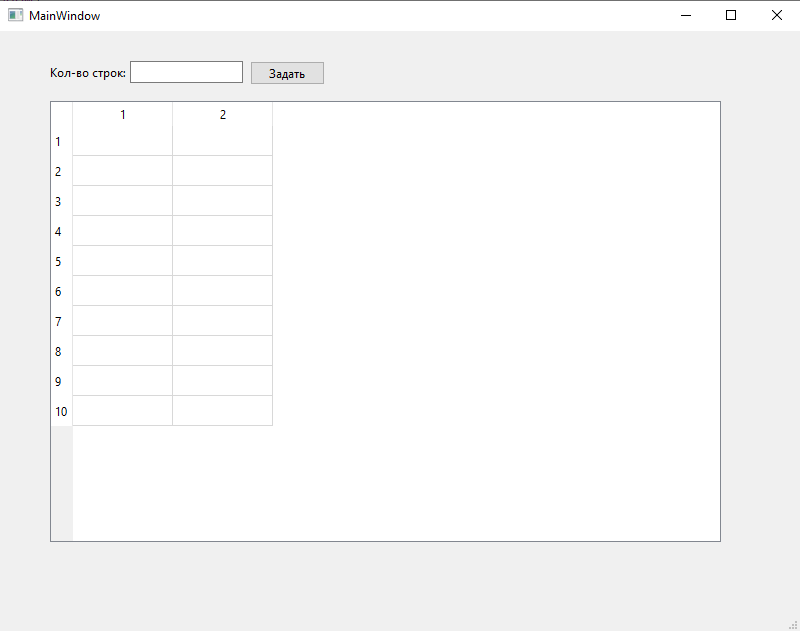
TableWidget на форме

Для задания количества строк и столбцов необходимо использовать методы setRowCount() и setColumnCount() соответственно:

ui->tableWidget\_data->setRowCount(10); // задать 10 строк в таблице

ui->tableWidget\_data->setColumnCount(2); // задать 2 столбца в таблице

Результат:



Результат установки 10 строк и 2 столбцов в таблице

Важно отметить, что при задании количества строк или столбцов, память под ячейки не выделяется, поэтому, если ячейка пустая, её значение будет NULL по умолчанию. Поэтому, прежде чем прочитать данные из ячейки, необходимо проверить, выделена ли под неё память.

Выделить память под ячейки можно сразу при создании строк и столбцов, для этого необходимо в цикле выделить память под каждую из ячеек, однако, стоит учитывать, что, если ячеек будет много, процесс выделения памяти может занять длительное время.

Ячейки таблицы в Qt представлены классом QTableWidgetItem, поэтому для инициализации ячейки необходимо создать объект этого класса и с помощью метода setItem(номер строки, номер столбца, указатель на ячейку) поместить его в таблицу в нужную ячейку:

// задаём количество строк и столбцов

ui->tableWidget\_data->setRowCount(10); // задать 10 строк в таблице

ui->tableWidget\_data->setColumnCount(2); // задать 2 столбца в таблице

// получаем кол-во строк в таблице

int rows = ui->tableWidget\_data->rowCount();

// получаем кол-во столбцов в таблице

int cols = ui->tableWidget\_data->columnCount();

// инициализируем ячейки

for(int i=0; i<rows; i++) // проходим по всем строкам

{

for(int j=0; j<cols; j++) // проходим по всем столбцам

{

// получаем указатель на ячейку (i, j)

QTableWidgetItem \*item = ui->tableWidget\_data->item(i, j);

if(item == nullptr) // если память под ячейку не выделена

{

item = new QTableWidgetItem; // создаём объект ячейки

// устанавливаем объект ячейки в таблицу

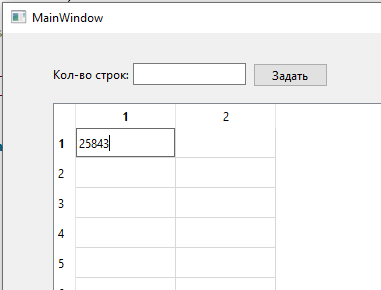
ui->tableWidget\_data->setItem(i, j, item);

}

}

}

Данные в таблицу можно вводить после двойного клика по выбранной ячейке:



Ввод данных в ячейку таблицы

Данные из ячейки можно получить с помощью метода text(), предварительно получив указатель на ячейку таблицы с помощью метода item(номер строки, номер столбца):

#include <QTableWidgetItem> // подключаем класс ячейки

…

// получаем указатель на ячейку 0-й строки, 1-го столбца

// нумерация строк и столбцов начинается с 0

QTableWidgetItem \*item = ui->tableWidget\_data->item(0, 1);

if(item != nullptr) // проверяем, что указатель не пустой

{

QString text = item->text(); // получаем текст ячейки

}

Аналогично данные в ячейки можно выводить, с помощью метода setText():

// получаем указатель на ячейку 2-й строки, 0-го столбца

// нумерация строк и столбцов начинается с 0

QTableWidgetItem \*item = ui->tableWidget\_data->item(2, 0);

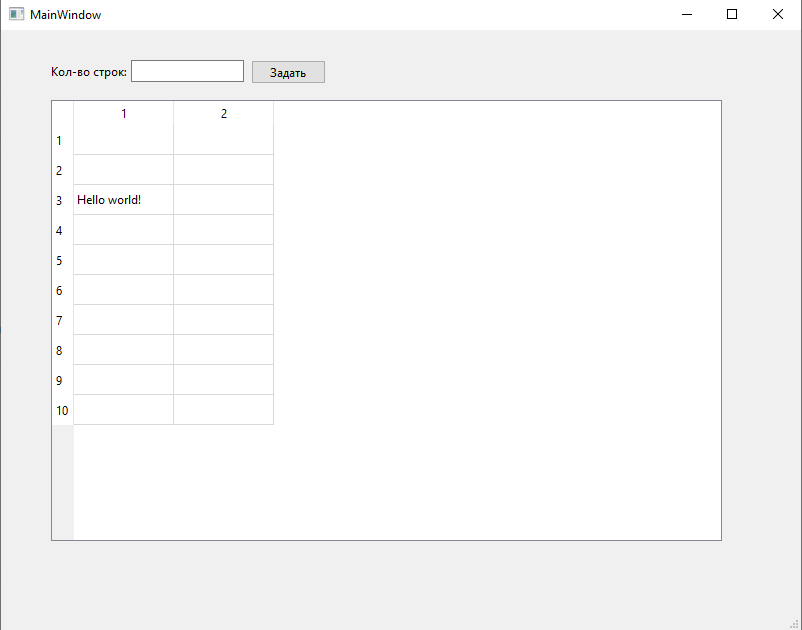
if(item != nullptr) // проверяем, что указатель не пустой

{

item->setText("Hello world!"); // выводим строку в ячейку

}

Результат:



Результат вывода строки в ячейку.

Чтобы считать данные из таблицы, будем использовать контейнер QList<QString>, на примере строк, при необходимости, можно использовать и другие типы данных:

// считывание данных

QList<QString> list; // создаём список для элементов

for(int i=0; i<rows; i++) // проходим по всем строкам

{

// получаем указатель на ячейку (i, j)

QTableWidgetItem \*item = ui->tableWidget\_data->item(i, 0);

if(item == nullptr) // если память под ячейку не выделена

{

list.append(""); // добавляем пустое значение

}

else // если же память выделена

{

list.append(item->text()); // считываем текст ячейки

}

}

// обработка данных

for(int i=0; i<list.size(); i++) // цикл по всем значениям

{

QString value = list[i]; // получаем значение из списка

}

В данном случае будут считаны значения только из первого столбца, т.к. при вызове метода item() параметр номера столбца равен константе – 0.

Вывод списка в таблицу производится аналогичным образом:

// вывод данных

// создаём список для элементов

QList<QString> list =

{"1", "6", "19", "2", "14", "5", "18", "4", "7", "9"};

// задаём необходимое количество строк и столбцов

ui->tableWidget\_data->setRowCount(list.size());

ui->tableWidget\_data->setColumnCount(1);

for(int i=0; i<list.size(); i++) // проходим по всем строкам

{

// получаем указатель на ячейку (i, j)

QTableWidgetItem \*item = ui->tableWidget\_data->item(i, 0);

// проверяем, что память выделена, если нет - выделяем

if(item == nullptr) // если память под ячейку не выделена

{

item = new QTableWidgetItem; // создаём объект ячейки

// устанавливаем объект ячейки в таблицу

ui->tableWidget\_data->setItem(i, 0, item);

}

// выводим значение в ячейку

item->setText(list[i]); // выодим значение из списка

}

Таким образом были рассмотрен базовый функционал работы с таблицами в Qt.