

## 1. Цель работы

научиться использовать виджет QCheckBox

## 2. Задание

Разработать форму, подобную приведенной на рисунке (студент может по желанию обрабатывать любую информацию)

The diagram shows a Qt widget layout on a grid background. At the top right is the text "Наименование основного предмета + картинка". Below it, on the left, is the label "Комплектующие:" followed by a list of checkboxes: "Шины", "Диски", "Каско", "ОСАГО", "Кондиционер", and "Сигнализация". To the right of this list is a "PushButton" widget labeled "PushButton -кнопка для расчета результата". Below the button is a "Label" widget labeled "Label - для вывода результата".

## 3. Форма программы

The screenshot shows a Qt application window titled "Пишите здесь". Inside, there is a form for configuring a computer. On the left, under the heading "Комплектующие", there is a list of checkboxes: "Материнская плата", "Видеокарта", "Процессор", "Оперативная память", and "Кулер". To the right of this list is a "Компьютер" widget containing an image of a computer case. Below the image, the following text is displayed: "Цена корпуса: 4000", "Цена комплектующих: 0", and "Итого: 4000".

Рисунок 1 – Форма

## 4. Описание формы

- QGroupBox – CompBox – Группа название-картинка
  - QLabel – name – Название
  - QLabel – picture – изображение
- QVBoxLayout – compListBase – Список комплектующих+название
  - QLabel - complectName – Заголовок
  - QVBoxLayout – compList – Список комплектующих
    - QCheckBox – cooler – Чекбокс Кулера
    - QCheckBox – motherBoard – Чекбокс материнской платы
    - QCheckBox – processor – Чекбокс процессора
    - QCheckBox – ram – Чекбокс оперативной памяти
    - QCheckBox – videocard - Чекбокс видеокарты
- QLabel – resultName – Цена корпуса и комплектующих

## 5. Текст программы

### main.cpp

```
#include "mainwindow.h"
```

```
#include <QApplication>
```

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

### mainwindow.cpp

```
#include "mainwindow.h"
```

```
#include "../ui_mainwindow.h"
```

```
#include "main_functions.h"
```

```
MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
```

```
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}
```

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

```
void MainWindow::on_motherBoard_checkStateChanged(const Qt::CheckState &arg1)
{
    ui->motherBoard->setProperty("Checked", isChecked(arg1));
    updateOutput(ui->compList, ui->resultName, ui->sale);
}
```

```
void MainWindow::on_videocard_checkStateChanged(const Qt::CheckState &arg1)
{
    ui->videocard->setProperty("Checked", isChecked(arg1));
    updateOutput(ui->compList, ui->resultName, ui->sale);
}
```

```
void MainWindow::on_processor_checkStateChanged(const Qt::CheckState &arg1)
{
    ui->processor->setProperty("Checked", isChecked(arg1));
    updateOutput(ui->compList, ui->resultName, ui->sale);
}
```

```
void MainWindow::on_ram_checkStateChanged(const Qt::CheckState &arg1)
{
    ui->ram->setProperty("Checked", isChecked(arg1));
    updateOutput(ui->compList, ui->resultName, ui->sale);
}
```

```
}
```

```
void MainWindow::on_cooler_checkStateChanged(const Qt::CheckState &arg1)
{
    ui->cooler->setProperty("Checked", isChecked(arg1));
    updateOutput(ui->compList, ui->resultName, ui->sale);
}
```

### **mainwindow.h**

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
```

```
#include <QMainWindow>
```

```
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_motherBoard_checkStateChanged(const Qt::CheckState &arg1);
```

```
    void on_videocard_checkStateChanged(const Qt::CheckState &arg1);
```

```
void on_processor_checkStateChanged(const Qt::CheckState &arg1);
```

```
void on_ram_checkStateChanged(const Qt::CheckState &arg1);
```

```
void on_cooler_checkStateChanged(const Qt::CheckState &arg1);
```

```
private:
```

```
    Ui::MainWindow *ui;
```

```
};
```

```
#endif // MAINWINDOW_H
```

### **main\_functions.h**

```
#ifndef MAIN_FUNCTIONS_H
```

```
#define MAIN_FUNCTIONS_H
```

```
#include <qboxlayout.h>
```

```
#include <qcheckbox.h>
```

```
#include <qlabel.h>
```

```
#include <qnamespace.h>
```

```
bool isChecked(const Qt::CheckState &arg1)
```

```
{
```

```
    if (arg1 == Qt::Checked)
```

```
    {
```

```
        return true;
```

```
    }
```

```
    else
```

```
    {
```

```
        return false;
```

```
    }
```

```
}
```

```
bool isChecked(const QWidget *arg1)
```

```
{
```

```
    if (arg1->property("Checked").toBool())
```

```

    {
        return true;
    }
    else
    {
        return false;
    }
}

```

```

void addSum(QString &arg1, const QVBoxLayout *arg2)

```

```

{
    int sum{0};
    for (int i = 0;i<arg2->count();i++)
    {

        if (isChecked(arg2->itemAt(i)->widget()))
        {
            sum += arg2->itemAt(i)->widget()->property("Price").toInt();
        }
    }
    arg1 += QString::number(sum);
}

```

```

bool isSaleActive(const QVBoxLayout *arg1)

```

```

{
    for (int i = 0;i<arg1->count();i++)
    {
        if (not(isChecked(arg1->itemAt(i)->widget())))
        {
            return false;
        }
    }
    return true;
}

```

```

void updateOutput(const QVBoxLayout *arg1, QLabel *arg2)
{
    arg2->clear();
    QString str_base = "Цена корпуса: 4000\nЦена комплектующих: ";
    QString str_price;

    addSum(str_price, arg1);

    str_base += str_price;

    if (isSaleActive(arg1))
    {
        QString final_price = QString::number(str_price.toInt() - str_price.toInt() * 0.1 + 4000);
        str_base += "\nСкидка на комплектующие: ";
        str_base += QString::number(str_price.toInt() * 0.1);
        str_base += "\nИтого: ";
        str_base += final_price;
    }
    else
    {
        str_price = QString::number(str_price.toInt()+4000);
        str_base += "\nИтого: ";
        str_base += str_price;
    }
    arg2->setText(str_base);
    arg2->adjustSize();
    qDebug() << str_base;
}

#endif // MAIN_FUNCTIONS_H

```

## **6. Пример работы программы**

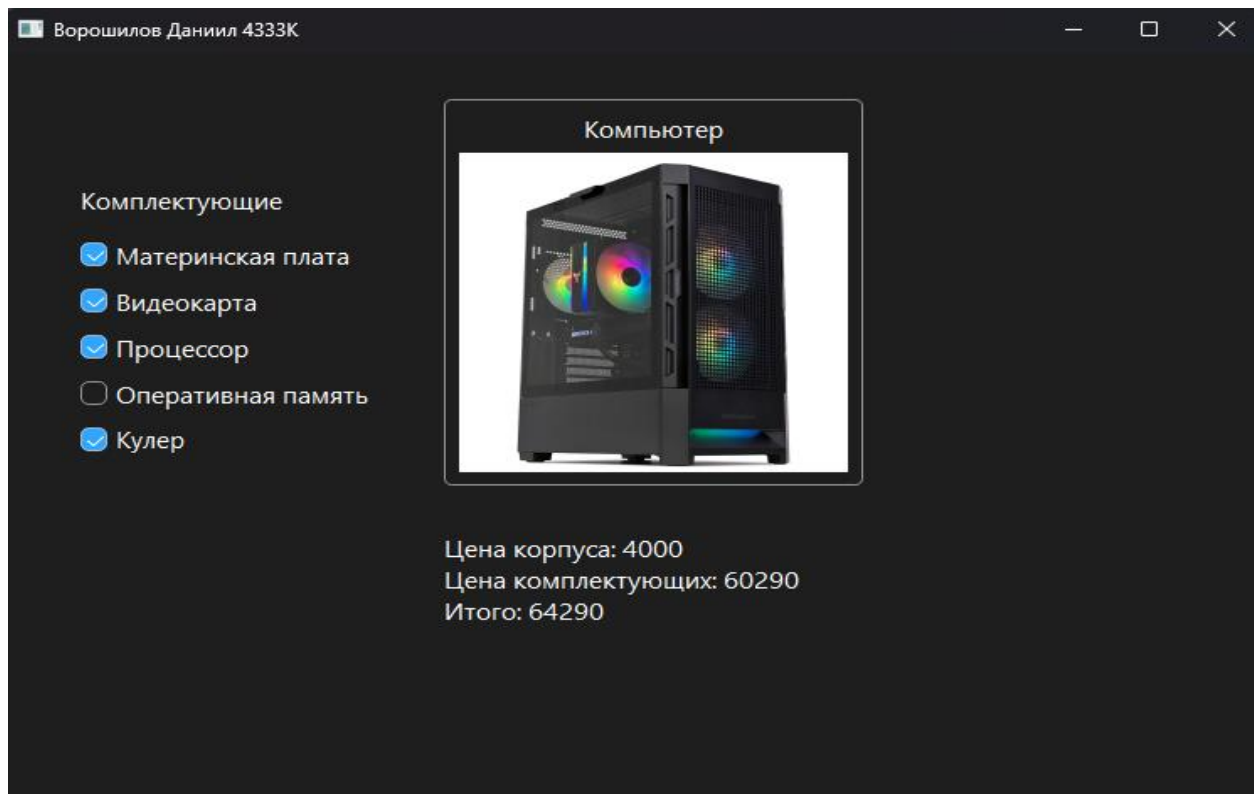


Рисунок 2 – Пример базовой работы

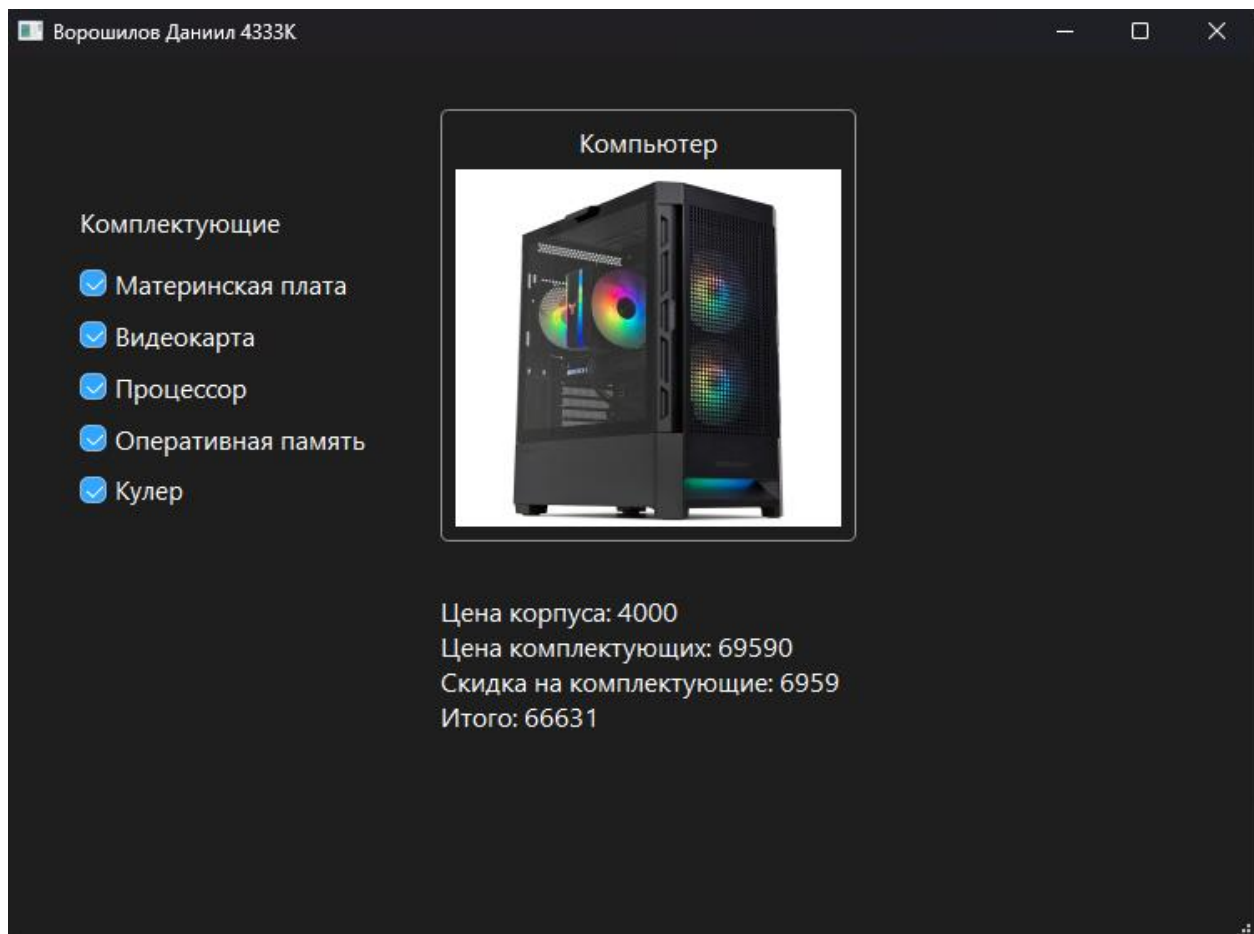


Рисунок 3 – Пример работы скидки



## **7. Выводы**

В процессе лабораторной работы был изучен принцип использования виджета QCheckBox