

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 6

з дисципліни “ Основи програмування ”

Тема “**Віконні програми** ”

|  |  |  |
| --- | --- | --- |
| Виконав  студент I курсу  групи КП-62  Лук’янець Михайло Олександрович  (*прізвище, ім’я, по батькові*)  варіант №15 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 2017 р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі** | **Оформлення звіту** | |  |  | | Нараховані бали:   |  |  |  | | --- | --- | --- | | **Корект. програм (2 бала)** | **Відп. на теор. питання (1 бал)** | **Відп. на прогр. питання (2 бала)** | |  |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2017

**Мета роботи**

Навчитися створювати віконні програми в ОС Linux за допомогою фреймворку Qt та редактора Qt Creator.

**Постановка завдання**

#### **Завдання №1. Типи даних**

Описати С++ клас *Динозавр*. Розділити інтерфейс і реалізацію класy між файлами TypeName.h та typename.cpp (де замість typename - *Динозавр* англійською мовою).

Всі поля даних класів мають бути приватними, для доступу до них використати методи геттери і сеттери. Створити декілька конструкторів класу.

#### **Завдання №2. Головне вікно**

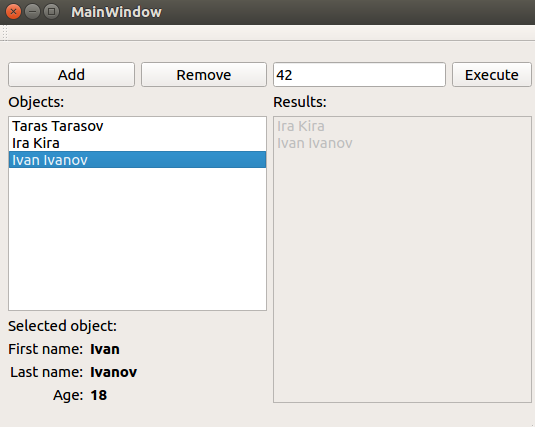
За допомогою стандартних колекцій мови С++ ініціалізувати список екземплярів класу *Динозавр*. На головному вікні розмістити список елементів, що відображає створений список екземплярів користувачу (у рядках списку відображати не всю інформацію, а лише назви екземплярів) і форму (пари текстовий опис поля і текстове значення значення поля) у якій відображати значення обраного елемента списку.

Поруч зі списком розмістити на вікні кнопки Add, Remove та Execute, поле для вводу тексту та ще один список для результатів:

* Нажаття на Add викликає додаткове вікно для створення нового елемента.
* Нажаття на Remove видаляє обраний елемент зі списку (тому дана кнопка має бути активною тільки коли якийсь елемент списку обраний).
* Нажання та Compute отримує із поля для вводу значення X та виконує дію: *Знайти всіх динозаврів із середньою масою тіла меншою за Х кг*.

Результатуючі елементи занести у список результатів.

Приблизний вид головного вікна:



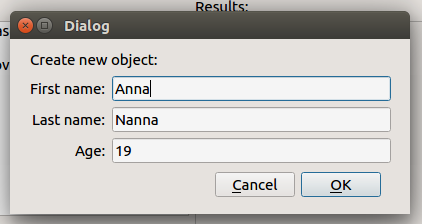
#### **Завдання №3. Діалог створення нових елементів**

Реалізувати діалог для створення нових екземплярів класу *Динозавр*, яке викликається за допомогою кнопки Addіз головного вікна.

Діалог має містити форму (пари текстовий опис поля і поле для вводу значення) для вводу значень полів нового екземпляра та дві кнопки OK та Cancel:

* Cancel відміняє створення нового елемента та закриває діалог.
* OK створює новий екземпляр класу із введеними користувачем значеннями, закриває діалог і додає новий елемент у список головного вікна.

Приблизний вид діалогу створення нового елемента:



**Тексти коду програм**

|  |
| --- |
| mainwindow.cpp |
| #include "mainwindow.h"  #include "ui\_mainwindow.h"  #include "dino.h"  #include <QString>  #include <QDebug>  #include "adddialog.h"  MainWindow::MainWindow(QWidget \*parent) :  QMainWindow(parent),  ui(new Ui::MainWindow)  {  ui->setupUi(this);  QString name("Ori");  QString fr("Sein");  Dino din(name, fr, 15.6, 48);  QVariant qVariant;  qVariant.setValue(din);  QListWidgetItem \*qDinoListItem = new QListWidgetItem();  qDinoListItem->setText(din.getName());  qDinoListItem->*setData*(Qt::*UserRole*, qVariant);  ui->Dino\_listWidget->addItem(qDinoListItem);  ui->pushButton\_2->setEnabled(false);  }  MainWindow::~*MainWindow*()  {  delete ui;  }  void MainWindow::on\_Dino\_listWidget\_itemSelectionChanged()  {  auto items = ui->Dino\_listWidget->selectedItems();  if (items.size() == 0) {  ui->pushButton\_2->setEnabled(false);  qDebug() << "no selected items";  return;  }  ui->pushButton\_2->setEnabled(true);  QListWidgetItem \* selectedItem = items[0];  Dino din = selectedItem->*data*(Qt::*UserRole*).value<Dino>();  ui->Age\_label->setText(QString::number(din.getAge()));  ui->Name\_label->setText(din.getName());  ui->Friend\_label->setText(din.getFriend());  ui->Mass\_label->setText(QString::number(din.getMass()));  }  void MainWindow::on\_pushButton\_2\_pressed()  {  QList<QListWidgetItem\*> items = ui->Dino\_listWidget->selectedItems();  foreach(QListWidgetItem \* item, items)  {  delete ui->Dino\_listWidget->takeItem(ui->Dino\_listWidget->row(item));  }  }  void MainWindow::on\_FindByMass\_pushButton\_pressed()  {  float critMass = ui->Mass\_doubleSpinBox->value();  QString res = "";  for(int index = 0; index < ui->Dino\_listWidget->count(); index++)  {  QListWidgetItem \* item = ui->Dino\_listWidget->item(index);  Dino din = item->*data*(Qt::*UserRole*).value<Dino>();  if(din.getMass() < critMass) {  res += din.getName();  res += "\n";  }  }  ui->Result\_textBrowser->setText(res);  }  void MainWindow::on\_pushButton\_pressed()  {  AddDialog dialog;  dialog.setWindowTitle("Enter data");  if (dialog.*exec*()) {  QString name;  QString fr;  int age;  float mass;  dialog.GetOptions(name, fr, age, mass);  Dino din(name, fr, mass, age);  QVariant qVariant;  qVariant.setValue(din);  QListWidgetItem \*qDinoListItem = new QListWidgetItem();  qDinoListItem->setText(din.getName());  qDinoListItem->*setData*(Qt::*UserRole*, qVariant);  ui->Dino\_listWidget->addItem(qDinoListItem);  qDebug() << "Accepted!";  } else {  qDebug() << "Rejected!";  }  } |

|  |
| --- |
| dino.cpp |
| #include "dino.h"  Dino::Dino (void){}  Dino::Dino (QString name, QString fr, float mass, int age){  this->name = name;  this->age = age;  this->mass = mass;  this->fr = fr;  }  QString Dino::getName(){  return name;  }  QString Dino::getFriend(){  return fr;  }  int Dino::getAge(){  return age;  }  float Dino::getMass(){  return mass;  } |

|  |
| --- |
| dino.h |
| #ifndef DINO\_H  #define DINO\_H  #include <string>  #include <QString>  #include <QMetaType>  using namespace std;  class Dino {  private:  QString name;  int age;  float mass;  QString fr;  public:  Dino();  Dino(QString name, QString fr, float mass, int age);  QString getName();  QString getFriend();  int getAge();  float getMass();  };  Q\_DECLARE\_METATYPE(Dino )  #endif // DINO\_H |

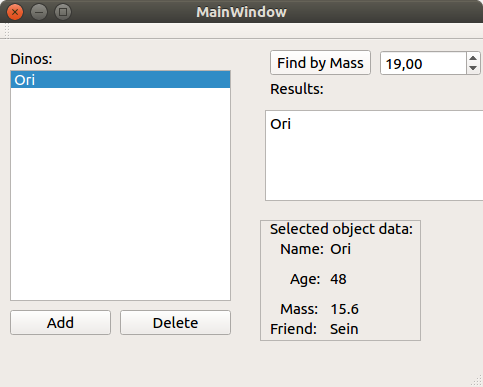
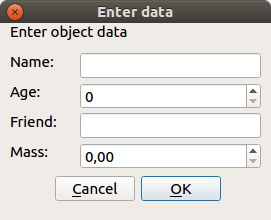
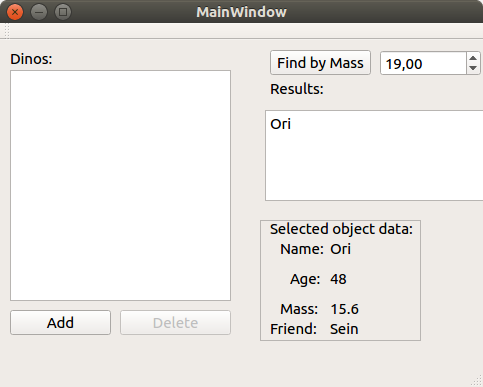
|  |
| --- |
| dino.cpp |
| #include "dino.h"  Dino::Dino (void){}  Dino::Dino (QString name, QString fr, float mass, int age){  this->name = name;  this->age = age;  this->mass = mass;  this->fr = fr;  }  QString Dino::getName(){  return name;  }  QString Dino::getFriend(){  return fr;  }  int Dino::getAge(){  return age;  }  float Dino::getMass(){  return mass;  } |

|  |
| --- |
| adddialog.cpp |
| #include "adddialog.h"  #include "ui\_adddialog.h"  AddDialog::AddDialog(QWidget \*parent) :  QDialog(parent),  ui(new Ui::AddDialog)  {  ui->setupUi(this);  }  AddDialog::~*AddDialog*()  {  delete ui;  }  void AddDialog::on\_buttonBox\_accepted()  {  }  void AddDialog::GetOptions (QString & name, QString & fr, int & age, float & mass){  name = ui->Name\_lineEdit->text();  fr = ui->Friend\_lineEdit->text();  mass = ui->Mass\_doubleSpinBox->value();  age = ui->Age\_spinBox->value();  } |

|  |
| --- |
| **adddialog.h** |
| **#ifndef ADDDIALOG\_H**  **#define ADDDIALOG\_H**  **#include <QDialog>**  **namespace Ui {**  **class AddDialog;**  **}**  **class AddDialog : public QDialog**  **{**  **Q\_OBJECT**  **public:**  **explicit AddDialog(QWidget \*parent = 0);**  **void GetOptions (QString & name, QString & fr, int & age, float & mass);**  **~*AddDialog*();**  **private slots:**  **void on\_buttonBox\_accepted();**  **private:**  **Ui::AddDialog \*ui;**  **};**  **#endif // ADDDIALOG\_H** |

|  |
| --- |
| **adddialog.cpp** |
| **#include "adddialog.h"**  **#include "ui\_adddialog.h"**  **AddDialog::AddDialog(QWidget \*parent) :**  **QDialog(parent),**  **ui(new Ui::AddDialog)**  **{**  **ui->setupUi(this);**  **}**  **AddDialog::~*AddDialog*()**  **{**  **delete ui;**  **}**  **void AddDialog::on\_buttonBox\_accepted()**  **{**  **}**  **void AddDialog::GetOptions (QString & name, QString & fr, int & age, float & mass){**  **name = ui->Name\_lineEdit->text();**  **fr = ui->Friend\_lineEdit->text();**  **mass = ui->Mass\_doubleSpinBox->value();**  **age = ui->Age\_spinBox->value();**  **}** |

**Приклади результатів**



**Висновки**

Виконавши дану лабораторну роботу було створено віконну програму в ОС Linux за допомогою фреймворку Qt та редактора Qt Creator.