МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

по учебной вычислительной (ознакомительной) практике

Коробко Николая Андреевича

студента 1 курса, 7 группы

специальность «Прикладная

математика»

Руководитель практики:

старший преподаватель

О.О.Колб

Минск, 2024

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc171280092)

[Введение 3](#_Toc171280093)

[Глава 1 Ознакомительная часть 5](#_Toc171280094)

[1.1 Общая характеристика курса «Контроль версий с помощью Git» 5](#_Toc171280095)

[1.1.1 Ознакомление с системами версионного контроля 5](#_Toc171280096)

[1.1.2 Установка Git 6](#_Toc171280097)

[1.1.3 Изучение базовых операций Git 6](#_Toc171280098)

[1.1.4 Разрешение конфликтов 6](#_Toc171280099)

[1.1.5 Итоги прохождения курса 7](#_Toc171280100)

[1.2 Конференция SoftClub и профориентация от EPAM 7](#_Toc171280101)

[1.2.1 Искусственный интеллект 7](#_Toc171280102)

[1.2.2 Кредитная система 8](#_Toc171280103)

[1.2.3 Полезные материалы от EPAM 8](#_Toc171280104)

[Выводы к главе 8](#_Toc171280105)

[Глава 2 Практическая часть курса 9](#_Toc171280106)

[2.1 Решение задач 9](#_Toc171280107)

[2.1.1 Юнит-тестирование 9](#_Toc171280108)

[2.1.1.1 Задача 1 9](#_Toc171280109)

[2.1.1.2 Задача 2 10](#_Toc171280110)

[2.1.1.3 Задача 3 10](#_Toc171280111)

[2.1.2 Работа с табличными данными 11](#_Toc171280112)

[2.1.3 Разработка простейшего SDI приложения 12](#_Toc171280113)

[2.1.4 Векторный графический редактор 14](#_Toc171280114)

[2.1.5 Разработка простейшей игры 16](#_Toc171280115)

[2.3 Выводы к главе 2 17](#_Toc171280116)

[Заключение 18](#_Toc171280117)

[Список использованных источников 19](#_Toc171280118)

# Введение

Содержание вычислительной части учебной практики:

24-26 июня. Знакомство с системой контроля версий Git. Тренинг на платформе EPAM. (основные вопросы для изучения)

1. Основы системы контроля версий:

- Что такое система контроля версий и почему она необходима для разработки программного обеспечения.

- Преимущества использования системы контроля версий Git.

2. Настройка Git:

- Установка Git на локальную машину.

- Конфигурация имени пользователя и электронной почты в Git с помощью команд git config --global user.name "Your Name" и git config --global user.email "your.email@example.com".

3. Базовые операции:

- Создание и коммит изменений с использованием команд git add <file> для добавления файлов в индекс и git commit -m "Commit message" для фиксации изменений.

- Просмотр истории коммитов с помощью команды git log.

- Работа с ветками: создание новой ветки с помощью команды git branch <branchname>, переключение на другую ветку с помощью команды git checkout <branchname> и удаление ветки с помощью команды git branch -d <branchname>.

- Восстановление предыдущих версий файлов с использованием команды git checkout <commit> -- <file>.

4. Слияние изменений (Merge):

- Понимание процесса слияния веток.

- Создание слияний (merge) веток в Git с помощью команды git merge <branchname>.

- Разрешение конфликтов при слиянии путем ручного редактирования файлов.

5. Разрешение конфликтов:

- Понимание возможных конфликтов при слиянии.

- Ручное разрешение конфликтов в Git путем редактирования конфликтных файлов.

- Использование инструментов для автоматического разрешения конфликтов.

Дополнение ранее разработанных классов системой Unit-тестов (для консольных приложений).

26 июня. Разработка простейшего GUI-приложения в QT (Диалоговое окно с виджетами, обеспечивающими ввод информации, вызов расчетной части приложения и вывод результатов)

Примечание: перевод 4-5 заданий лабораторных работ, сделанных ранее, как консольные приложения.

27 июня. Разработка приложения, отражающего табличные данные с возможностью корректировки (на основе шаблона Модель-Вид), вывода и ввода данных.

28-29 июня. Разработка простейшего SDI-приложения, иллюстрирующего использование меню, корректировку, сохранение и восстановление текстовой, числовой информации в текстовом (QTextStream) и бинарном(QDataStream) формате. Оформление промежуточного отчета по первой половине учебной практики.

1-2 июля. Разработка приложения, аналога Векторного графического редактора, с обработкой событий от мыши, сериализацией набора графических объектов в файлах.

3-5 июля. Разработка проекта с графической анимацией объектов на экране (простейшей игры) (допускается доработка проекта до сентября).

6 июля. Оформление окончательного отчета (допускается дополнение отчета, если сделана доработка проекта с анимацией.

Целью учебной вычислительной (ознакомительной) практики является закрепление полученных знаний за первый год обучения посредством выполнения специальных учебных заданий, участия в работе над командным проектом.

Задачи практики:

* Знакомство с IT компанией, возможностями обучения и повышения квалификации, знакомство с разными направлениями профессиональной подготовки IT специалистов.
* Изучение инструментов для совместной разработки проектов и контроля версий.
* Подготовка и настройка программных сред и средств тестирования для прохождения тренинга по выбранному направлению.
* Выполнение стандартных заданий.
* Выполнение заданий повышенного уровня.
* Оформление отчета.

# Глава 1 Ознакомительная часть

## 1.1 Общая характеристика курса «Контроль версий с помощью Git»

Указать, какие курсы изучены в рамках практики. Их описание, предназначение.

В рамках первых дней учебной практики (24-25 июня) мною был пройден обучающий онлайн курс «Version Control with Git», созданный компанией EPAM. Ссылка на курс: <https://learn.epam.com/detailsPage?id=601f195a-d408-4439-a16d-0630ed2a412e>.

Основная цель данного курса - подготовить фундамент для изучения программирования, эффективного использования современных инструментов разработчика.

Необходимость прохождения курса заключалась в том, что практически каждый современный проект использует системы версионного контроля. Git - самый популярный инструмент для контроля версий. Этот курс помог научиться использовать git для решения ежедневных задач. Курс также был полезен для понимания различных стратегий организации ветвления в проекте и выборе оптимальной.

## Ознакомление с системами версионного контроля

В самом начале курса было осуществлено общее ознакомление с понятием систем версионного контроля. Появилось понимание необходимости использования подобных систем для дальнейшего развития в сфере разработки программного обеспечения, а также комфортной, продуктивной и безопасной коллективной работы по созданию различных проектов.

Кроме того, была определена наиболее оптимальная система версионного контроля под названием Git. С преимуществами данной системы можно ознакомиться на скриншоте, расположенным ниже:



Рис. 1

## Установка Git

Дальнейшим этапом прохождения курса стала установка Git на локальную машину, настройка, а также регистрация на интернет-платформе Github. Успешно освоены основные команды, необходимые для корректного использования Git Bash и Git Gui. После чего был создан аккаунт на платформе Github. Далее с целью дальнейшего применения полученных теоретических навыков был создан репозиторий. Ссылка на репозиторий: <https://github.com/itkorobko/MyFirstRep?tab=readme-ov-file#myfirstrep>.

## Изучение базовых операций Git

В ходе дальнейшего изучения Git были изучены основные команды, сфера их применения и правила использования. Примеры базовых инструкций: git add <file>, git commit -m "Commit message", git log. Кроме того, были изучены команды, позволяющие использовать разветвления и слияние веток, такие как: git branch <branchname>, git branch -d <branchname”, git checkout <commit> -- <file>, git merge <branchname>.

## Разрешение конфликтов

Иногда, при коллективной разработке проекта возникает конфликт, т.е. невозможность автоматического слияния файлов, которые были редактированы разными разработчиками. Пример конфликта:

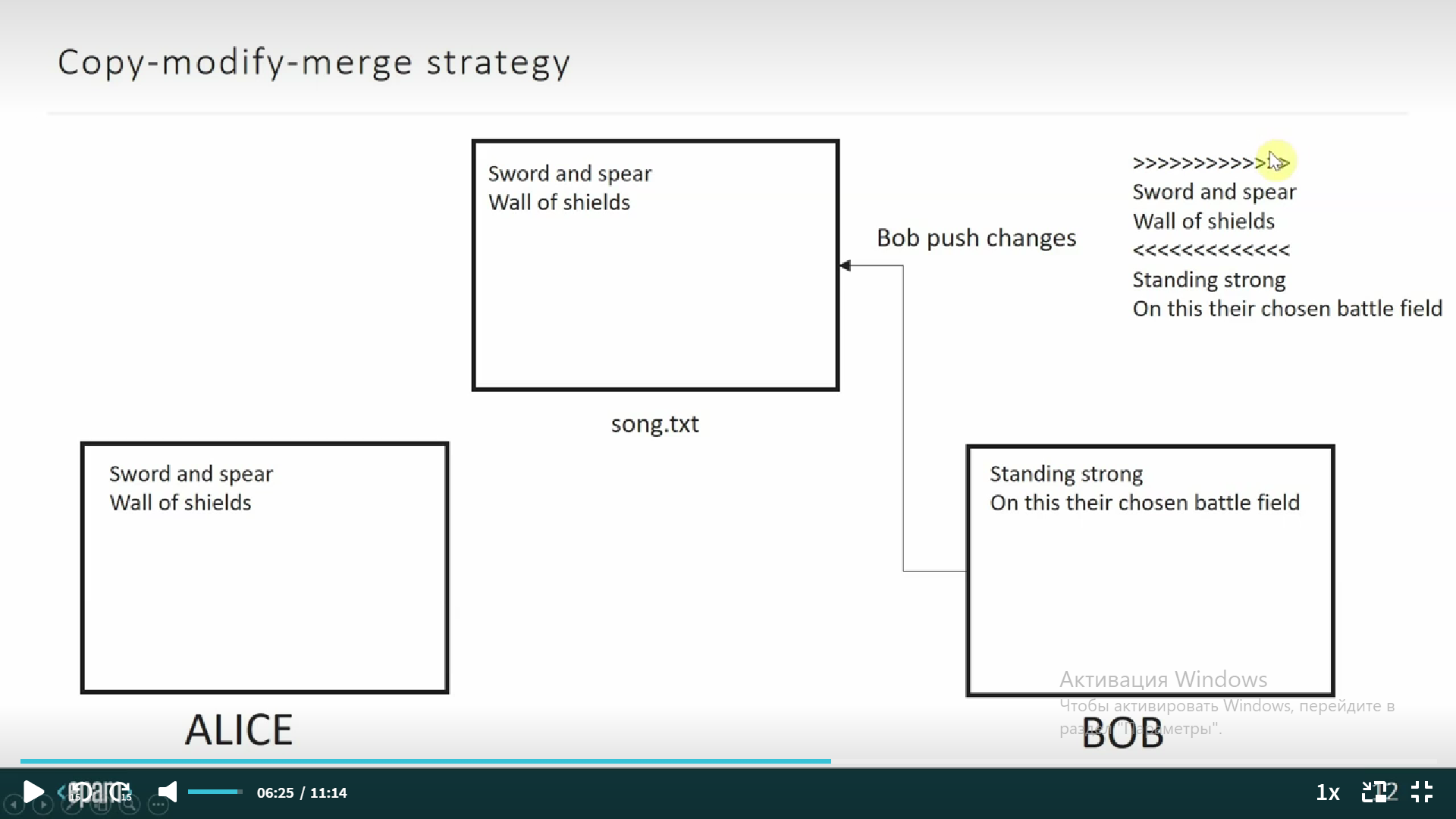


Рис.2

При возникновении подобных конфликтов возникает необходимость либо в использовании ручного редактирования файлов, либо с помощью средств для автоматического разрешения конфликтов.

## Итоги прохождения курса

Пройденный курс позволил успешно освоить основы систем версионного контроля, а также дал все необходимые навыки для применения подобных систем на практике. Курс пройден на 100%.

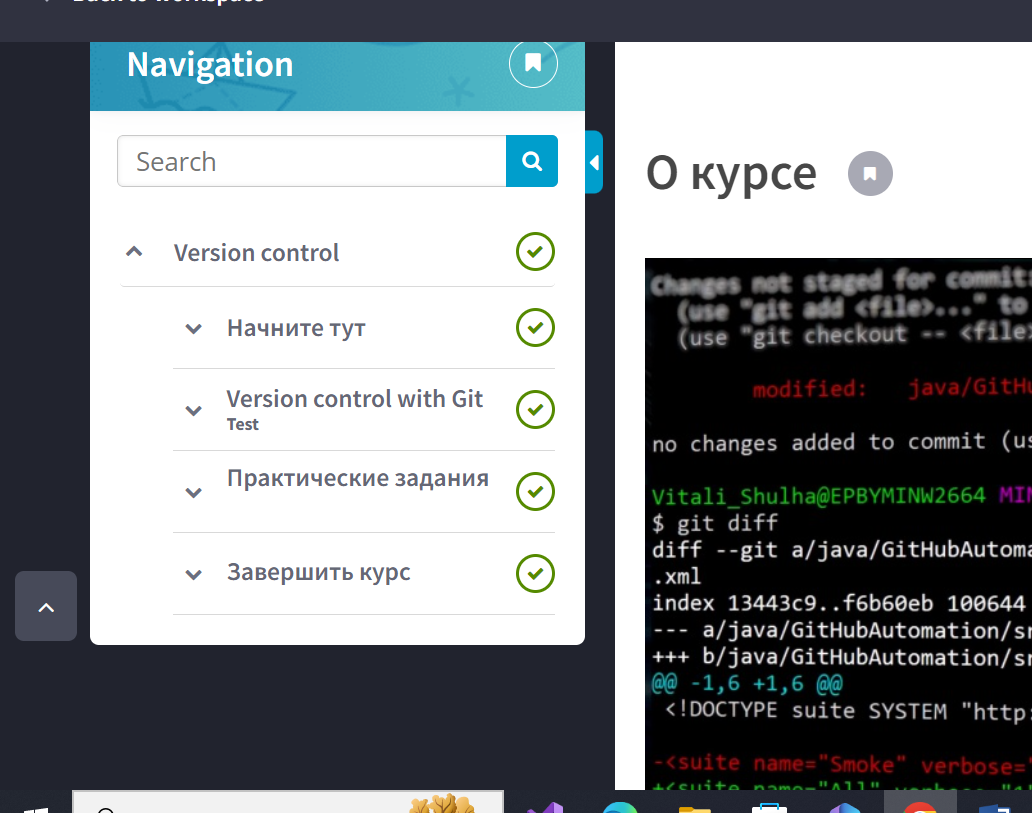


Рис. 3

## Конференция SoftClub и профориентация от EPAM

В рамках ознакомительной части учебной практики была посещена конференция, организованная известной международной IT компанией SoftClub.

Целью данной конференции было узнать подробнее о востребованных сегодня направлений IT сферы, а также современных методах решения разных проблем и задач.

Кроме того, было просмотрены полезные видео от компании EPAM, целью которого была найти свое место в глобальной IT сфере.

## Искусственный интеллект

После выступления первых спикеров конференции SoftClub, стало очевидно, что центральное место в современной IT сфере занимает развитие и управление искусственным интеллектом, который в свою очередь позволяет значительно модифицировать и ускорить решение различных задач и проблем. Современный искусственный интеллект — это система, которая способна воспринимать свою среду и принимать меры, чтобы максимизировать шансы на успешное достижение своих целей, а также интерпретировать и анализировать данные таким образом, чтобы они обучались и адаптировались по мере развития.

## Кредитная система

Поскольку значительная часть ресурсов команды компании SoftClub уходит на разработку приложений для различных банковских систем, немалая часть выступления была посвящена основным принципам работы кредитной системы, а также основам финансовой грамотности.

Особый интерес был проявлен к роли разработчиков программного обеспечения в данной сфере, а также количеству задействованных лиц.

На конкретных примерах, таких как: Единое расчетное информационное пространство, банковские приложения и многое другое.

## Полезные материалы от EPAM

В рамках ознакомительной части практики, также были изучены полезные материалы от компании EPAM. А именно: прочитаны интересные статьи, просмотрено видео и пройден тест, посвященные определению своего места в IT сфере. Ссылка: https://wearecommunity.io/events/vfhMV7eINF

## Выводы к главе

В ходе теоретической части учебной практики я ознакомился с понятием систем версионного контроля. Узнал о важности подобных систем для создания коллективных проектов, а также для удобного хранения и безопасной корректировки собственных репозиториев. Для дальнейшего использования была установлена программа Git, так как она является самой удобной в использовании. Кроме того, я прослушал конференцию компании SoftClub, в ходе которой получил большой объём необходимых знаний для дальнейшего устройства на работу по специальности.

# Глава 2 Практическая часть курса

## 2.1 Решение задач

## 2.1.1 Юнит-тестирование

## 2.1.1.1 Задача 1

Условие: Заданы два целых числа. Найти наибольший общий делитель этих двух чисел.

Фрагмент из myclass.cpp

int MyClass::NOD(int a, int b){

int res=1;

if(a>b)

swap(a,b);

for(int i=1;i<=a;i++){

if(a%i==0 && b%i==0)

res=i; }

return res;}

Фрагмент из tst\_mytest.cpp

void MyTest::test\_case1(){

MyClass mc;

QCOMPARE(mc.NOD(3,8),1);

QCOMPARE(mc.NOD(24,4),4);

QCOMPARE(mc.NOD(12,16),4);

}

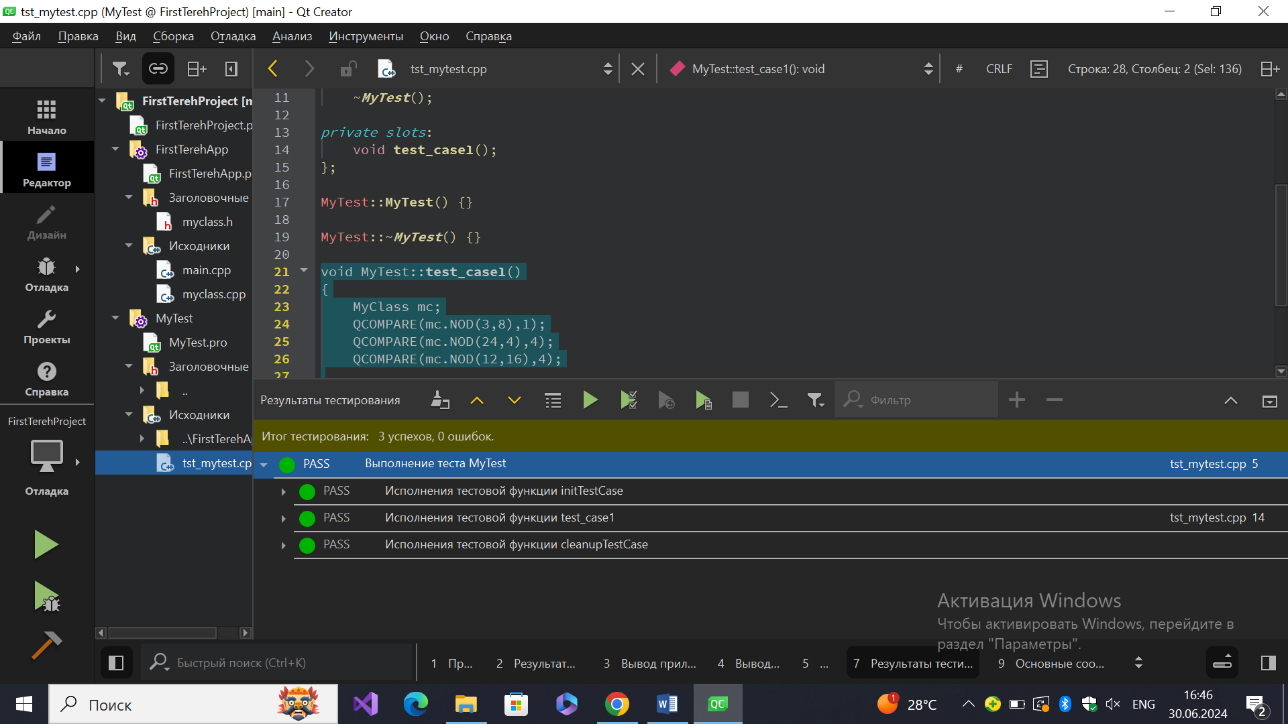


Рис. 4

## 2.1.1.2 Задача 2

Условие задачи: Задано целое число. Определить является ли оно простым.

Фрагмент из myclass.cpp

bool MyClass::prime\_num(int a){

if(a<2)

return false;

else{

for(int i=2;i<a;i++){

if(a%i==0)

return false;}

return true;}}

Фрагмент из tst\_mytest.cpp

void MyTest::test\_case1(){

MyClass mc;

QCOMPARE(mc.prime\_num(-1),false);

QCOMPARE(mc.prime\_num(0),false);

QCOMPARE(mc.prime\_num(1),false);

QCOMPARE(mc.prime\_num(3),true);}

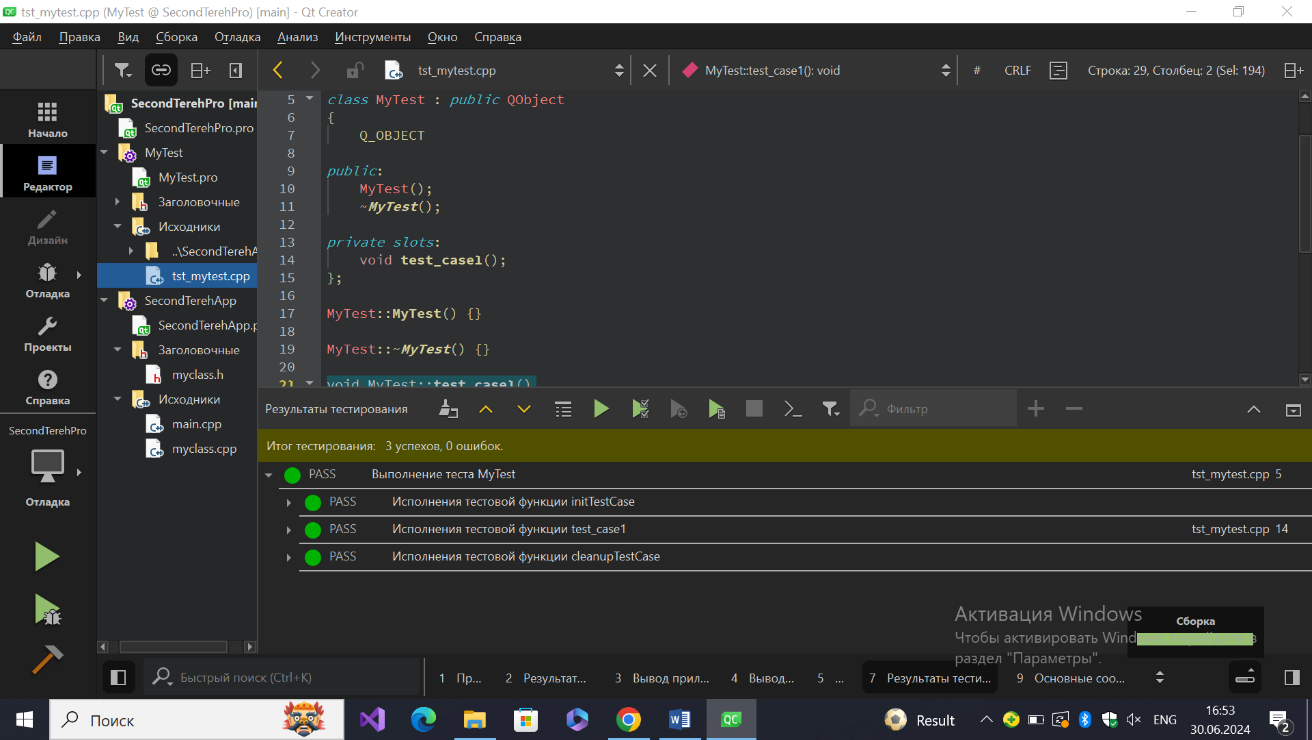


Рис. 5

## 2.1.1.3 Задача 3

Условие задачи: Задано целое число. Определить сумму его цифр.

Фрагмент из myclass.cpp

int MyClass::sum\_of\_numbers(int a){

int res=0;

if(a<0)

a\*=-1;

while(a>0){

res+=a%10;

a/=10; }

return res;}

Фрагмент из tst\_mytest.cpp

void MyTest::test\_case1(){

MyClass mc;

QCOMPARE(mc.sum\_of\_numbers(0),0);

QCOMPARE(mc.sum\_of\_numbers(4),4);

QCOMPARE(mc.sum\_of\_numbers(-4),4);

QCOMPARE(mc.sum\_of\_numbers(145),10);}

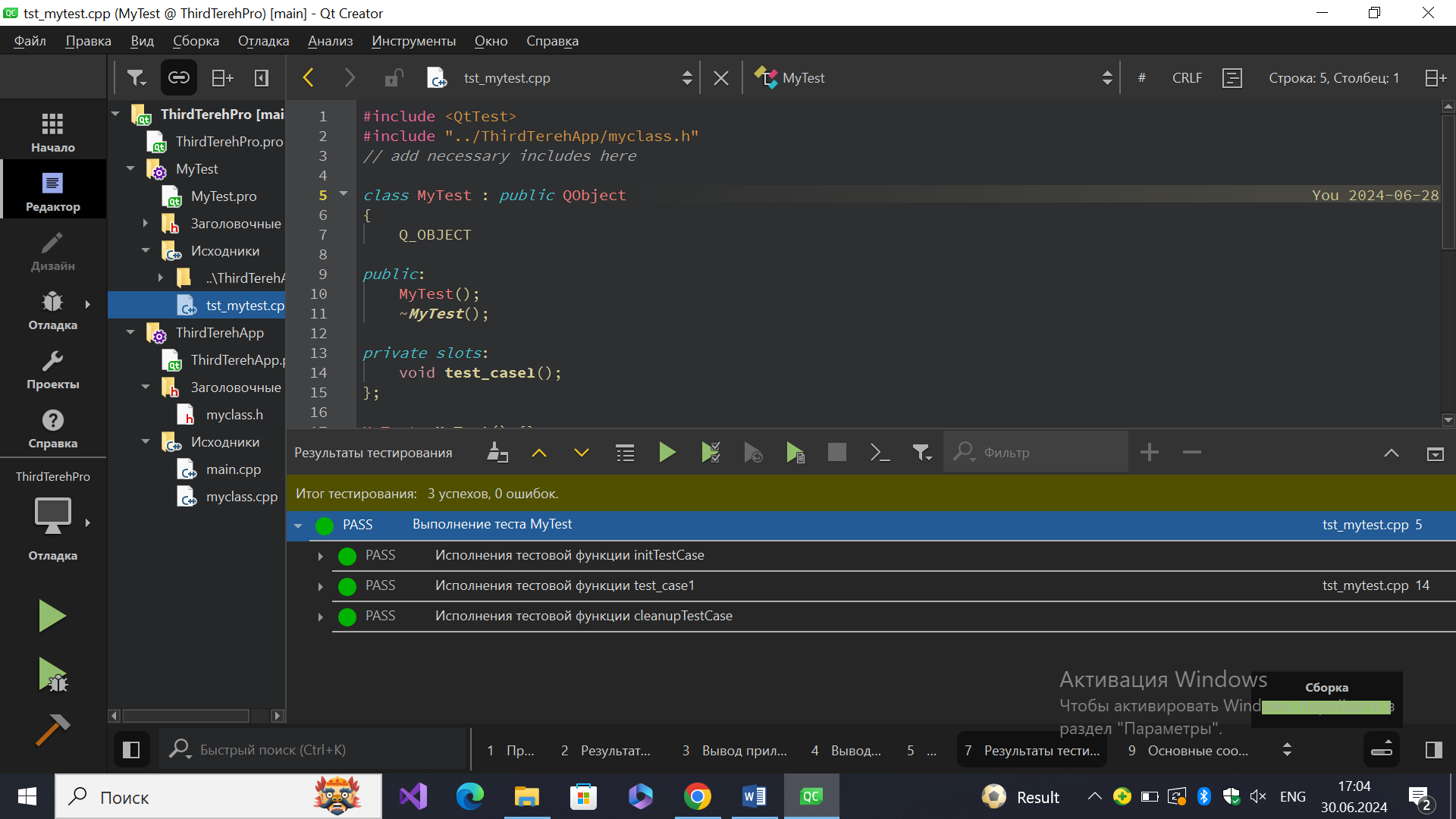


Рис. 6

## 2.1.2 Работа с табличными данными

Условие: разработать программу, которая может читать, редактировать и записывать табличные данные, используя TableView. Табличные данные брать на основе Excel.

Фрагмент из файла mainwindow.cpp (чтение из файла.xlsx и запись в файл.xlsx)

void MainWindow::on\_pushButton\_clicked(){

QXlsx::Document xlsx;

int row = 1; int col = 1;

Document xlsxR("Table\_APL\_1.xlsx");

if (xlsxR.load()){

Cell\* cell = xlsxR.cellAt(row, col);

while(cell!=nullptr)

cell=xlsxR.cellAt(++row,1);

row--;

cell=xlsxR.cellAt(1,1);

while(cell!=nullptr)

cell=xlsxR.cellAt(1,++col);

col--;

model=new QStandardItemModel(row,col);

ui->tableView->setModel(model);

for(int i=0; i<row;i++){

for(int j=0;j<col;j++){

cell=xlsxR.cellAt(i+1,j+1);

QVariant var = cell->readValue();

model->setItem(i,j, new QStandardItem(var.toString()));}}}}

void MainWindow::on\_pushButton\_2\_clicked(){

QXlsx::Document xlsx;

int row,col;

row=ui->tableView->model()->rowCount();

col=ui->tableView->model()->columnCount();

for(int i=0; i<row;i++){

for(int j=0;j<col;j++){

QString temp;

QModelIndex index = model->index(i,j,QModelIndex());

temp = model->data(index,0).toString();

xlsx.write(i+1,j+1, temp); }}

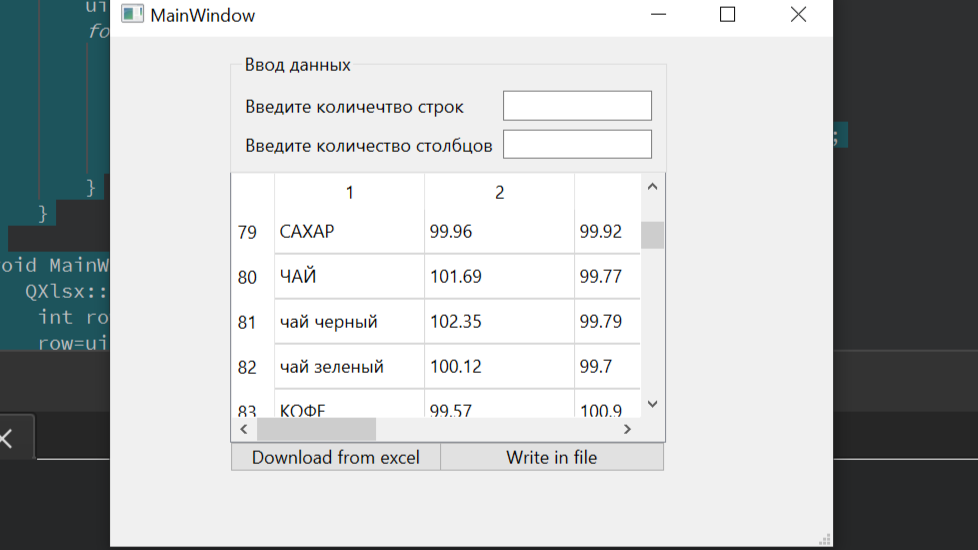
xlsx.saveAs("Test.xlsx");}

Рис. 7

## 2.1.3 Разработка простейшего SDI приложения

Условие: Разработать приложение, которое реализует работу простейшего текстового редактора, включая обработку текстовых и бинарных файлов.

Фрагмент из файла mainwindow.cpp

void MainWindow::on\_action\_triggered(){ //Открытие файла

fileName = QFileDialog::getOpenFileName(this,"Выбор файла", "C:/", "\*.txt ;; \*.bin");

QFile myFile(fileName);

if (!myFile.open(QIODevice::ReadOnly | QIODevice::Text))

return;

QByteArray a;

if(fileName[fileName.size()-1]=='t')

a=myFile.readAll();

else{

QDataStream potok(&myFile);

while(!potok.atEnd())

potok>>a; }

ui->textEdit->setText(QString(a));

myFile.close();}

void MainWindow::on\_action\_3\_triggered(){//Сохранить файл

fileName = QFileDialog::getSaveFileName(this, tr("Save File"), "/newDoc.txt", tr("myDoc (\*.txt);;myDoc (\*.bin)"));

if(!fileName.isEmpty()) {

QFile myFile(fileName);

if(!myFile.exists()) {

if (!myFile.open(QIODevice::WriteOnly | QIODevice::Text))

return;

QByteArray strByteArray = ui->textEdit->toPlainText().toUtf8();

if(fileName[fileName.size()-1]=='t')

myFile.write(strByteArray);

else{

QDataStream potok(&myFile);

potok<<strByteArray; }

myFile.close();}}}

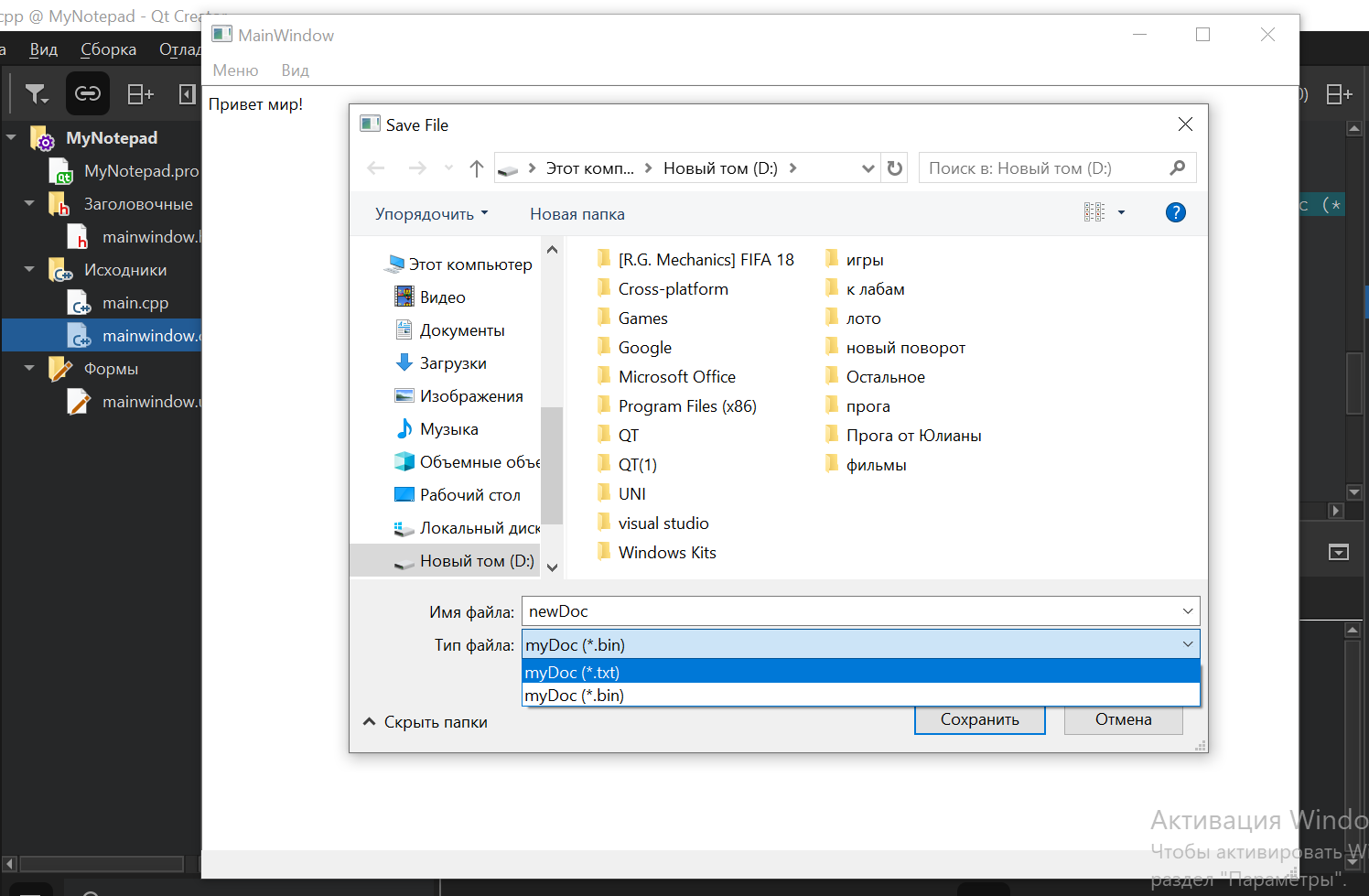


Рис. 8

## 2.1.4 Векторный графический редактор

Условие: Разработать приложение, реализующее простейший векторный графический редактор.

Отрывок из mainwindow.cpp

MainWindow::MainWindow(){ //Установка центрального виджета

scribbleArea = new ScribbleArea;

setCentralWidget(scribbleArea);

createActions();

createMenus();

setWindowTitle(tr("СуперРисовалка"));

resize(500, 500);}

Отрывок из scribblearea.cpp

void ScribbleArea::mouseReleaseEvent(QMouseEvent \*event){//Обработка отпускания мыши

if (event->button() == Qt::LeftButton && scribbling) {

switch(mode){

case LINE\_DRAWING:{

drawLineTo(event->pos());

scribbling = false;

QImage c\_image=image;

v\_i.push\_back(c\_image);

break; }

case RECT\_DRAWING:{

scribbling=false;

drawRectangle(event->pos());

QImage c\_image=image;

v\_i.push\_back(c\_image);

mode=LINE\_DRAWING;

break;}

case CIRCLE\_DRAWING:{

scribbling=false;

drawCircle(event->pos());

QImage c\_image=image;

v\_i.push\_back(c\_image);

mode=LINE\_DRAWING;

break;}}}}

void ScribbleArea::paintEvent(QPaintEvent \*event){// Рисование создающегося объекта

QPainter painter(this);

if(mode==LINE\_DRAWING)

painter.drawImage(QPoint(0, 0), image);

else

painter.drawImage(QPoint(0, 0), v\_i[v\_i.size()-1]);

if(mode==COME\_BACK\_DRAWING)

mode=LINE\_DRAWING;

if (scribbling && mode == RECT\_DRAWING) {

painter.setPen(QPen(myPenColor, myPenWidth, Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

QRect rect(startPoint, myEndPoint);

painter.drawRect(rect.normalized());}

if (scribbling && mode == CIRCLE\_DRAWING) {

painter.setPen(QPen(myPenColor, myPenWidth, Qt::SolidLine, Qt::RoundCap, Qt::RoundJoin));

QRect rect(startPoint, myEndPoint);

painter.drawEllipse(rect.normalized());}}

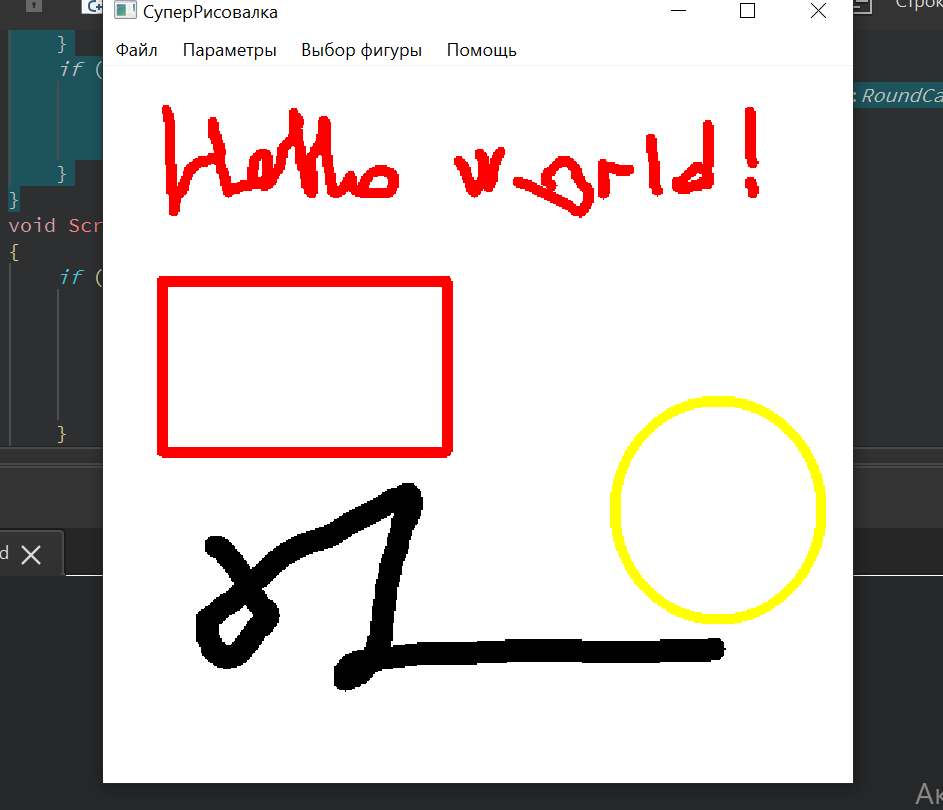


Рис. 9

## 2.1.5 Разработка простейшей игры

Условие работы:

«Необходимо получить программу, которая будет удовлетворять следующим условиям:

1.Основой принцип: На экране появляется фотография преподавателя, которая начинает убегать от курсора. В свою очередь курсор изображается в виде бейсбольной биты. Задача состоит в том, чтобы левым щелчком мыши «стукнуть» по фотографии. У фотографии есть 3 жизни, после третьего попадания за фотографией приезжает катафалк и увозит ее за пределы экрана.

2.Попадание по фотографии сопровождается соответствующим звуком.

3. Погоня сопровождается веселой музыкой.

4. Катафалк забирает фотографию под грустную музыку.

5. После попадания по фотографии на ней появляются следы удара (синяки и т.д.).

6. Наличие интерактивного меню.

7. В меню можно выбрать различных персонажей и количество жизней.

8. Засекается время прохождения игры.

9. Результаты заносятся в «Базу данных» в качестве рейтинга.

Т.к. программа содержит более 20 файлов с исходным текстом программы, вставка кода в отчет становится затруднительной. Поэтому, желании исходный код можно посмотреть по ссылке из пункта 1.1.2.

## 2.3 Выводы к главе 2

В ходе практической части практики, были реализованы такие необходимы навыки как: тестирование программ, простейший графический интерфейс, простейшая анимация, работа с файлами разного типа. Эти навыки были использованы для разработки вышеперечисленных проектов.

# Заключение

В ходе учебной практики, я ознакомился со многими навыками, необходимыми для быстрой, качественной, корректной и современной разработки программного обеспечения и создания коллективных проектов.

Основными вопросами, изученными во время теоретической части учебной практики, стали:

* основы систем версионного контроля
* установка Git для дальнейшего использования
* ознакомление с базовыми операциями Git
* участие в конференции компании SoftClub
* просмотр материалов, помогающих определится с выбором будущей профессии, от компании EPAM.

Основными задачами, решенными в ходе практической части учебной практики, стали:

* Модульное тестирование, ранее решенных задач
* Работа с табличными данными, с использованием QTableView
* Разработка простейшего SDI приложения, на примере текстового редактора
* Разработка простейшего векторного графического редактора
* Разработка простейшей игры

По итогам проделанной в ходе учебной практики работы, можно сделать вывод, что большинство задач, поставленных перед началом практики выполнены, цель учебной практики достигнута.

# Список использованных источников

1. Макс Шлее. QT 5.10. Профессиональное программирование на С++. – Санкт-Петербург, «БХВ-Петербург»
2. Вводные материалы по QT с портала edufpmi: https://edufpmi.bsu.by/course/view.php?id=81 (Курс: Программирование ПМ 1к летний семестр (bsu.by) )
3. https://radioprog.ru/post/987 Модульное тестирование кода на C++ с помощью Qt.
4. Книга, Бен Штрауб и Скотт Чакон. “Pro Git”.
5. Ознакомительные материалы от компании EPAM <https://wearecommunity.io/events/vfhMV7eINF>
6. Обучающий онлайн курс «Version Control with Git», созданный компанией EPAM. <https://learn.epam.com/detailsPage?id=601f195a-d408-4439-a16d-0630ed2a412e>.