Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

ПНИПУ

УДК 004

**Творческая работа**

**по теме:**

**автоматизация рабочего места психолога**

Выполнил:   
студент группы РИС-23-1б   
Сингур Иван Сергеевич

Молодых Никита Андреевич

Фаезов Александр Рамисович

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь, 2024 г.

**«ТЕМПЕРАМЕНТ ЛИЧНОСТИ»**

**ПОСТАНОВКА ЗАДАЧИ**

1. Разделившись на группы, выбрать тему для творческого проекта.
2. Распределить обязанности и создать план работы.
3. Осуществить визуализацию и интерфейс.
4. Создать описание творческого проекта при помощи видео

**АНАЛИЗ ЗАДАЧИ**

Void start\_Click(System::Object^ sender, System::EventArgs^ e)-Открывает 2-ю форму и хайдит(скрывает) первую

Void next(int n)-Переход к следующему вопросу относительно блоков вопросов.

Void da\_Box\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)- Отвечают за ответы.

Void net\_Box\_CheckedChanged(System::Object^ sender, System::EventArgs^ e)-Отвечают за ответы

Void timer1\_Tick(System::Object^ sender, System::EventArgs^ e)-Функция таймер.

Void dalee\_Click(System::Object^ sender, System::EventArgs^ e)-Взаимодействуя с выбранным ответом прибавляет счетчик и меняет флаг.

Void nazad\_Click(System::Object^ sender, System::EventArgs^ e)-Выполняет переход к предыдущему вопросу, меняя счетчик.

Void konec\_Click(System::Object^ sender, System::EventArgs^ e)-Отвечает за результат тестирования.

**ВВЕДЕНИЕ**

НИР на тему "Автоматизация рабочего места психолога" имеет огромное значение в современном мире, где стремительно развиваются технологии и цифровизация. Психологическая работа требует не только специальных знаний и навыков, но и эффективных инструментов для проведения исследований, диагностики и консультирования клиентов. В процессе создания творческой работы были рассмотрены основные методы автоматизации рабочего места психолога, их преимущества и возможные проблемы при внедрении. Анализ этих аспектов помог понять, какие решения могут быть наиболее эффективными для оптимизации работы и повышения качества услуг, предоставляемых клиентам психологическими специалистами.

Объект исследования: процесс автоматизации рутинных задач на рабочем месте специалиста с использованием программных средств.

Предмет исследования: программа-приложение, автоматизирующая автоматическую проверку работ и получение конечного результата для рабочего специалиста (в данном случае психолога).

Цель: разработка программы на языке программирования С++ для автоматизации рутинных задач, повышения эффективности и производительности труда специалиста.

Задачи:

1. Провести анализ трудоемких задач, подлежащих автоматизации: выявить наиболее часто выполняемые рутинные операции, занимающие значительное время специалиста.

2. Разработать алгоритм автоматизации выбранных задач: определить последовательность действий программы для реализации автоматизации.

3. Разработать код на языке программирования С++: реализовать алгоритм автоматизации в виде программного кода.

4. Разработать пользовательский интерфейс: обеспечить удобство и простоту использования программы специалистом.

**КОД НА ЯЗЫКЕ С++**

Для того, чтобы написать код следуя выбранному алгоритму, нужно выбрать платформу, на которой будет происходить реализация. Для своей лабораторной работы я буду использовать Visual Studio.

Интегрированная среда разработки Visual Studio является творческой стартовой площадкой, которую можно использовать для редактирования, отладки и сборки кода, а также для публикации приложения. В дополнение к стандартному редактору и отладчику, предоставляемых большинством интегрированных сред разработки, Visual Studio включает компиляторы, средства завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки программного обеспечения. Visual Studio является самой быстрой интегрированной средой разработки для повышения производительности.[4]

**Вывод:**

Таким образом, выбрав Visual Studio в качестве платформы для своей лабораторной работы, вы получаете доступ к мощному инструменту с широким набором функций, упрощающих процесс разработки программного обеспечения. Visual Studio позволит вам эффективно редактировать, отлаживать и собирать код, реализующий выбранный вами алгоритм оптимизации. Богатый функционал среды, включая компиляторы, средства завершения кода и графические конструкторы, поможет вам сосредоточиться на решении задачи и создании качественного кода.

С кодом можно ознакомиться в приложении А

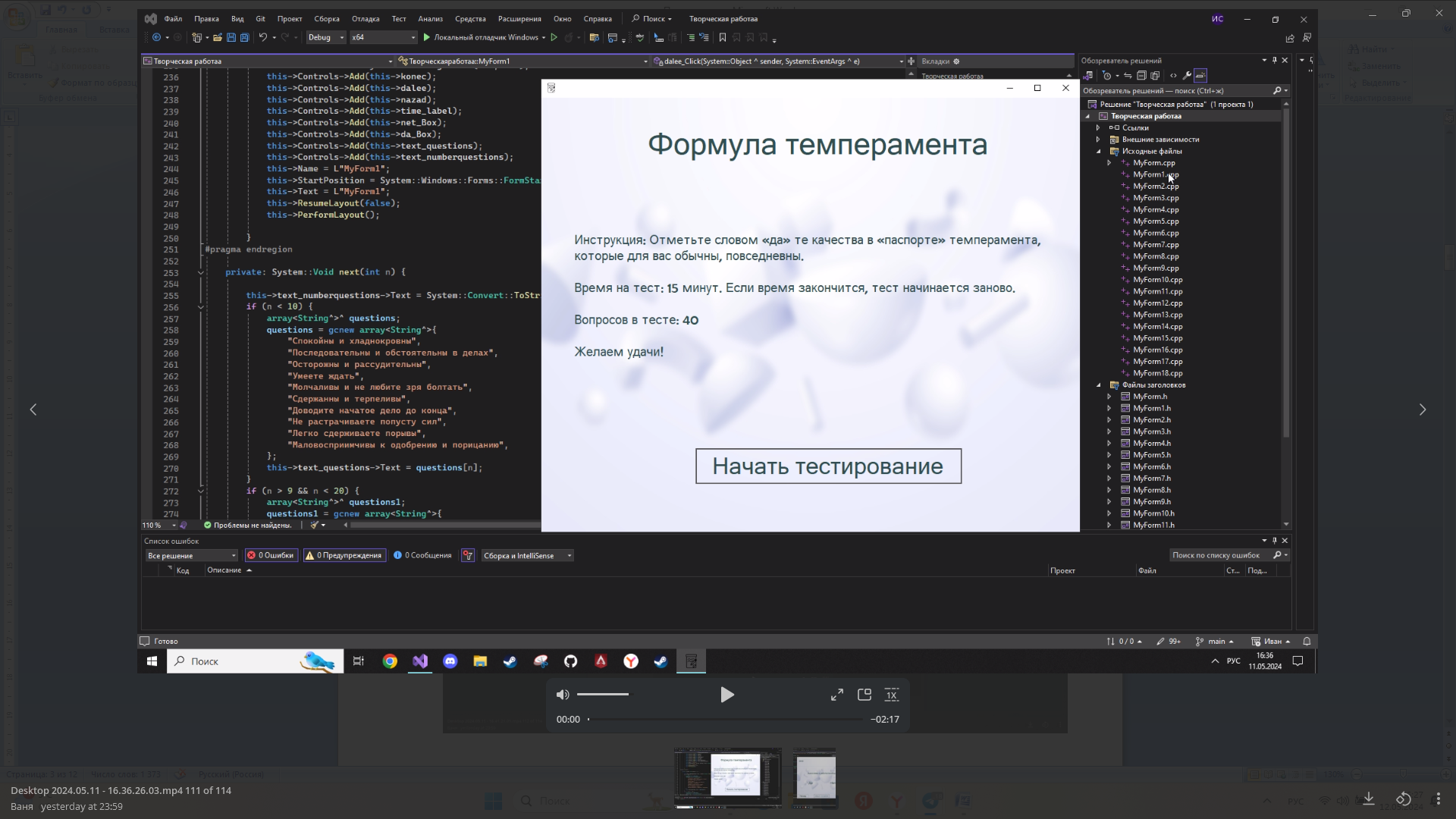
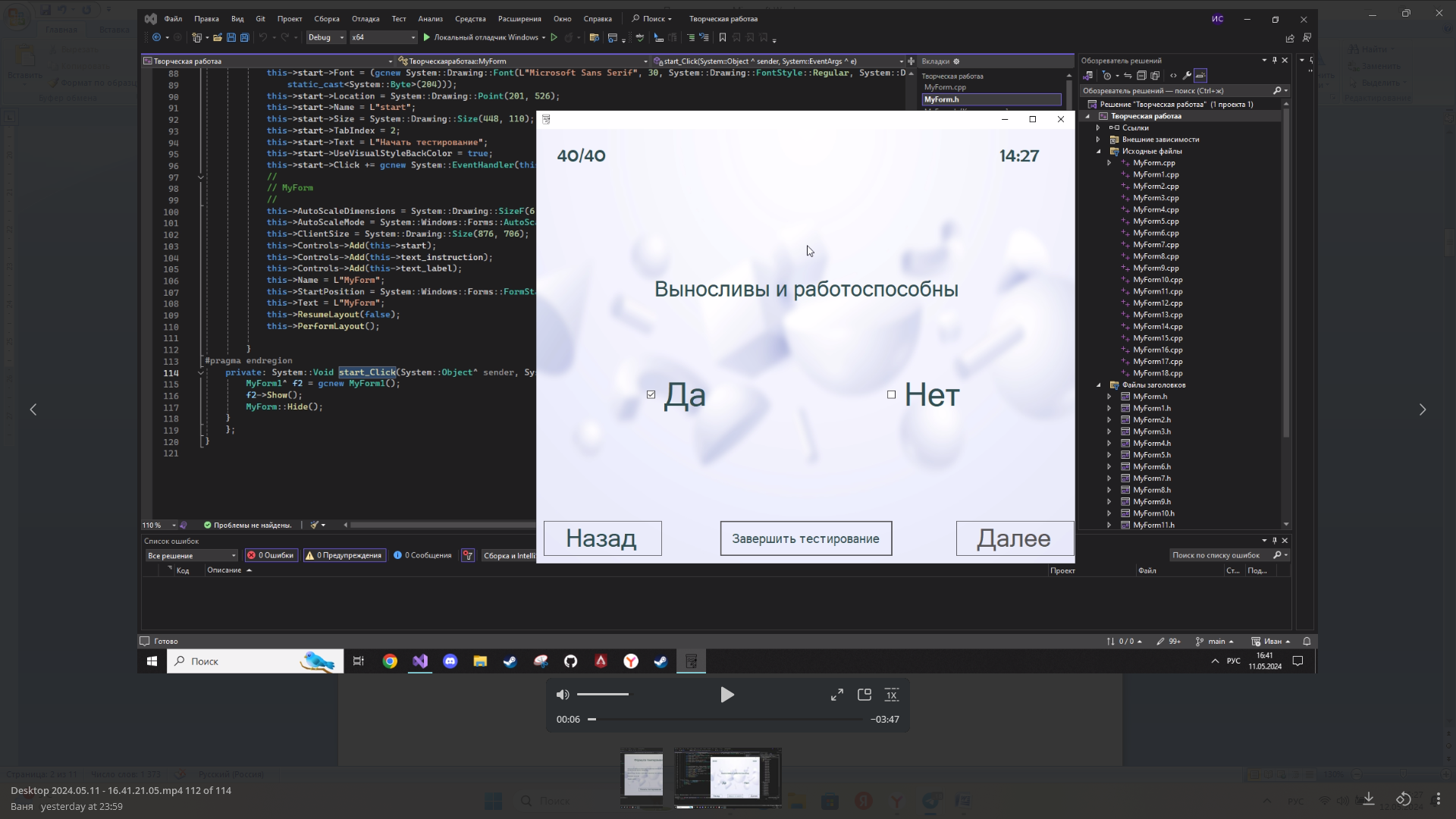
**РАБОТА ПРОГРАММЫ**

1)Ознакомительная форма.В ней представлена вся необходимая инфармация перед началом теста.(Рисунок 1)

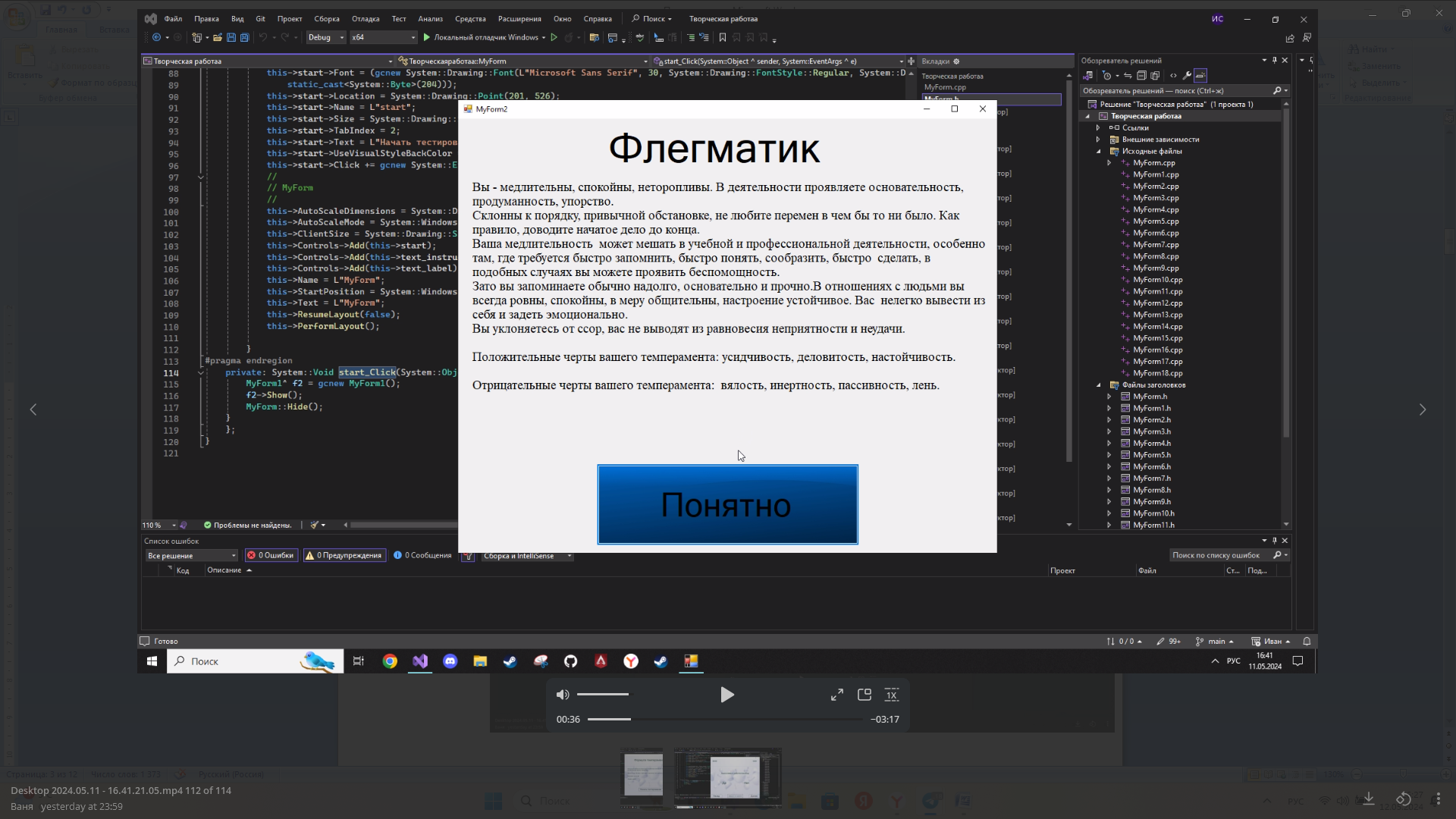
2)Форма теста. Содержит вопросы, варианты ответа, время и т.д.

(Рисунок 2)

3)Заключительная форма. Содержит вид темперамента и его обьяснение.

****(Рисунок 3)

**Рисунок 1 Рисунок 2**

** Рисунок 3**

**ССЫЛКА НА ФОРМУ** [**https://drive.google.com/drive/folders/1yHvcbcli0wNho78JYUx-0DZNxH4uqa\_w**](https://drive.google.com/drive/folders/1yHvcbcli0wNho78JYUx-0DZNxH4uqa_w)

**ЗАКЛЮЧЕНИЕ**

В ходе проведенного исследования было установлено, что автоматизация рабочего места психолога является актуальной и востребованной темой в современном обществе. Внедрение специализированных программ и технологий позволяет оптимизировать процессы работы психолога, увеличивая его эффективность и результативность. Автоматизация позволяет улучшить качество предоставляемых услуг, упростить доступ к информации и ускорить процесс принятия решений. Таким образом, данное исследование подтверждает необходимость внедрения автоматизации на рабочих местах психологов для повышения их профессиональной деятельности и улучшения услуг, оказываемых клиентам.

**ПРИЛОЖЕНИЕ А**

1)Форма на ознакомительное окно

#pragma endregion

private: System::Void start\_Click(System::Object^ sender, System::EventArgs^ e) {

MyForm1^ f2 = gcnew MyForm1();

f2->Show();

MyForm::Hide();

}

};

}

2)Форма на тест

#pragma endregion

private: System::Void next(int n) {

this->text\_numberquestions->Text = System::Convert::ToString(n + 1) + "/40";

if (n < 10) {

array<String^>^ questions;

questions = gcnew array<String^>{

"Спокойны и хладнокровны",

"Последовательны и обстоятельны в делах",

"Осторожны и рассудительны",

"Умеете ждать",

"Молчаливы и не любите зря болтать",

"Сдержанны и терпеливы",

"Доводите начатое дело до конца",

"Не растрачиваете попусту сил",

"Легко сдерживаете порывы",

"Маловосприимчивы к одобрению и порицанию",

};

this->text\_questions->Text = questions[n];

}

if (n > 9 && n < 20) {

array<String^>^ questions1;

questions1 = gcnew array<String^>{

"Стеснительны и застенчивы",

"Теряетесь в новой обстановке",

"Затрудняетесь установить контакт с новыми людьми",

"Не верите в свои силы",

"Легко переносите одиночество",

"Чувствуете подавленность и растерянность при неудачах",

"Склонны уходить в себя",

"Быстро утомляетесь",

"Обладаете тихой речью, иногда снижающейся до шепота",

"Впечатлительны до слезливости",

};

this->text\_questions->Text = questions1[n-10];

}

if (n > 19 && n < 30) {

array<String^>^ questions2;

questions2 = gcnew array<String^>{

"Неусидчивы, суетливы",

"Невыдержанны, вспыльчивы",

"Нетерпеливы",

"Резки и прямолинейны в отношениях с людьми",

"Решительны и инициативны",

"Упрямы",

"Находчивы в споре",

"Работаете рывками",

"Склонны к риску",

"Незлопамятны и необидчивы",

};

this->text\_questions->Text = questions2[n - 20];

}

if (n > 29 && n < 40) {

array<String^>^ questions3;

questions3 = gcnew array<String^>{

"Веселы и жизнерадостны",

"Энергичны и деловиты",

"Часто не доводите начатое дело до конца",

"Склонны переоценивать себя",

"Способны быстро схватывать новое",

"Неустойчивы в интересах и склонностях",

"Легко переживаете неудачи и неприятности",

"Легко приспосабливаетесь к различным обстоятельствам",

"С увлечением беретесь за любое новое дело",

"Выносливы и работоспособны",

};

this->text\_questions->Text = questions3[n - 30];

}

}

private: System::Void da\_Box\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (da\_Box->Checked) {

net\_Box->Checked = false;

}

}

private: System::Void net\_Box\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (net\_Box->Checked) {

da\_Box->Checked = false;

}

}

private: System::Void timer1\_Tick(System::Object^ sender, System::EventArgs^ e) {

if (Second == 0) {

Minute = Minute - 1;

Second = 60;

}

Second--;

if (Minute == 0 && Second == 0) {

Second = 0;

Minute = 15;

this->n = 0;

this->count = 0;

this->j = 0;

this->k = 0;

this->i = 0;

next(this->n);

}

sec = Convert::ToString(Second);

Min = Convert::ToString(Minute);

if (Second < 10) {

time\_label->Text = Min + ":0" + sec;

}

else {

time\_label->Text = Min + ":" + sec;

}

}

private: System::Void dalee\_Click(System::Object^ sender, System::EventArgs^ e) {

if (this->n < 39) {

previousAnswerYes = da\_Box->Checked;

if (da\_Box->Checked == true) {

if (n < 10) {

count++;

}

if (n > 9 && n < 20) {

j++;

}

if (n > 19 && n < 30) {

k++;

}

if (n > 29 && n < 39) {

i++;

}

}

else {

count = count;

j = j;

k = k;

i = i;

}

next(this->n + 1);

net\_Box->Checked = false;

da\_Box->Checked = false;

this->n = n + 1;

}

else {

if (this->n == 39) {

previousAnswerYes = da\_Box->Checked;

if (da\_Box->Checked == true) {

i++;

dalee->Enabled = false;

if (count >= 7) {

flag1 = true;

}

if (j >= 7) {

flag2 = true;

}

if (k >= 7) {

flag3 = true;

}

if (i >= 7) {

flag4 = true;

}

}

else {

count = count;

j = j;

k = k;

i = i;

dalee->Enabled = false;

if (count >= 7) {

flag1 = true;

}

if (j >= 7) {

flag2 = true;

}

if (k >= 7) {

flag3 = true;

}

if (i >= 7) {

flag4 = true;

}

}

}

}

if (flag1 ==true) {

flagcount++;

}

if (flag2 == true) {

flagcount++;

}

if (flag3 == true) {

flagcount++;

}

if (flag4 == true) {

flagcount++;

}

}

private: System::Void nazad\_Click(System::Object^ sender, System::EventArgs^ e) {

if (n != 0 && n < 40) {

nazad->Enabled = true;

dalee->Enabled = true;

if (previousAnswerYes) {

if (n < 10) {

count--;

}

if (n > 9 && n < 20) {

j--;

}

if (n > 19 && n < 30) {

k--;

}

if (n > 29 && n < 39) {

i--;

}

}

net\_Box->Checked = false;

da\_Box->Checked = false;

next(this->n - 1);

this->n = n - 1;

}

}

private: System::Void konec\_Click(System::Object^ sender, System::EventArgs^ e) {

if (flagcount == 0 || flagcount >= 3) {

MyForm18^ f18 = gcnew MyForm18();

f18->Show();

MyForm1::Hide();

}

else {

if (flag1 == true) {

MyForm2^ f3 = gcnew MyForm2();

f3->Show();

MyForm1::Hide();

if (flag3 == true && flag1 == true) {

f3->Hide();

}

if (flag1 == true && flag2 == true) {

f3->Hide();

}

if (flag1 == true && flag4 == true) {

f3->Hide();

}

}

if (flag2 == true) {

MyForm3^ f4 = gcnew MyForm3();

f4->Show();

MyForm1::Hide();

if (flag2 == true && flag3 == true) {

f4->Hide();

}

if (flag2 == true && flag1 == true) {

f4->Hide();

}

if (flag2 == true && flag4 == true) {

f4->Hide();

}

}

if (flag3 == true) {

MyForm4^ f5 = gcnew MyForm4();

f5->Show();

MyForm1::Hide();

if (flag3 == true && flag2 == true) {

f5->Hide();

}

if (flag3 == true && flag1 == true) {

f5->Hide();

}

if (flag3 == true && flag4 == true) {

f5->Hide();

}

}

if (flag4 == true) {

MyForm5^ f6 = gcnew MyForm5();

f6->Show();

MyForm1::Hide();

if (flag4 == flag1) {

f6->Hide();

}

if (flag4 == flag2) {

f6->Hide();

}

if (flag4 == flag3) {

f6->Hide();

}

}

if (flag3 == true && flag2 == true) {

MyForm6^ f7 = gcnew MyForm6();

f7->Show();

MyForm1::Hide();

if (j > k) {

f7->Hide();

}

}

if (flag3 == true && flag2 == true && j > k) {

MyForm7^ f8 = gcnew MyForm7();

f8->Show();

MyForm1::Hide();

}

if (flag3 == true && flag1 == true) {

MyForm8^ f9 = gcnew MyForm8();

f9->Show();

MyForm1::Hide();

if (count > k) {

f9->Hide();

}

}

if (flag3 == true && flag1 == true && count > k) {

MyForm9^ f10 = gcnew MyForm9();

f10->Show();

MyForm1::Hide();

}

if (flag3 == true && flag4 == true) {

MyForm10^ f11 = gcnew MyForm10();

f11->Show();

MyForm1::Hide();

if (i > k) {

f11->Hide();

}

}

if (flag3 == true && flag4 == true && i > k) {

MyForm11^ f12 = gcnew MyForm11();

f12->Show();

MyForm1::Hide();

}

if (flag2 == true && flag1 == true) {

MyForm12^ f13 = gcnew MyForm12();

f13->Show();

MyForm1::Hide();

if (count > j) {

f13->Hide();

}

}

if (flag2 == true && flag1 == true && count > j) {

MyForm13^ f14 = gcnew MyForm13();

f14->Show();

MyForm1::Hide();

}

if (flag2 == true && flag4 == true) {

MyForm14^ f15 = gcnew MyForm14();

f15->Show();

MyForm1::Hide();

if (i > j) {

f15->Hide();

}

}

if (flag2 == true && flag4 == true && i > j) {

MyForm15^ f16 = gcnew MyForm15();

f16->Show();

MyForm1::Hide();

}

if (flag1 == true && flag4 == true) {

MyForm16^ f17 = gcnew MyForm16();

f17->Show();

MyForm1::Hide();

if (i > count) {

f17->Hide();

}

}

if (flag1 == true && flag4 == true && i > count) {

MyForm17^ f18 = gcnew MyForm17();

f18->Show();

MyForm1::Hide();

}

}

}

};

}

3)Пример одной из результативных форм

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

Close();

Application::Exit();

}

};

**}**