­Міністерство освіти і науки України  
Інженерний навчально-науковий інститут

Запорізького національного університету

Кафедра програмного забезпечення

автоматизованих систем

Курсова робота

з дисципліни «Об’єктно-орієнтоване програмування»

Тема: “ Об’єктна модель мікропроцесора пральної машини”

Виконав: Студент групи 6.1210-ПЗС

Коваленко Кирило Максимович

Керівник: доцент кафедри ПЗАС Попівщий В.І.

Запоріжжя

2021

**Завдання курсової роботи:   
“Об’єктна модель мікропроцесора пральної машини”**

Потрібно розробити об'єктну модель програмного забезпечення вбудованого мікропроцесора пральної машини. Машина призначена для автоматичного прання білизни. У машині є бак клапани для забору й зливу води, мотор, обладнання підігріву води, таймер, дверцята для доступу в бак, кілька ємностей для різних мийних засобів, панель керування із кнопками й індикатором. У пам'яті машини зберігаються 5 програм прання, задані виготовлювачем, які можна вибрати за допомогою відповідних кнопок на панелі. Користувачі не можуть вносити в них зміни. Кожна програма визначає температуру води, тривалість прання, використовувані мийні засоби (номер ємності й час подачі), швидкість обертання бака під час прання й отжима.

Для використання машини необхідно відкрити дверцята у формі MainPage, помістити білизну в бак, помістити мийні засоби в ємності та вибрати кількість води для прання у текстовому полі textBox1 та textBox2 у формі EnterData, вибрати програму прання й натиснути на кнопку «Пуск». Перед тим як приступити до прання машина відкриває клапан для забору води, набирає необхідну кількість води, після чого закриває клапан. Далі, машина діє по обраній користувачем програмі: підігріває, якщо необхідно воду до потрібної температури; вмикає таймер і запускає обертання бака для прання; по таймеру подає в бак мийні засоби, передбачені програмою; по закінченню прання зливає воду й запускає віджим.

Під час роботи на індикаторі висвітлюється час, що пройшов від початку прання (хвилини й секунди), поточний режим роботи (прання або віджим), номер поточної програми прання. Користувач має можливість у будь-який момент натиснути на кнопку «Зупинити», щоб примусово зупинити прання й злити воду.

**Реферат**

Курсова робота призначена для закріплення базових знань мови C#.  
Метою є вивчення основ мови C# та .Net Framework. Вивчення алгоритмів, логіки, основних понять та ієрархій класів, та методів.

Метою курсової роботи є побудова об’єктної моделі та програмного забезпечення для виконання практичної реалізації задачі.

В результаті роботи було виконано більшість пунктів поставленої задачі та реалізована модель програмного забезпечення каталогу ресурсів Інтернет. В ході виконання курсової роботи було задіяно використання різних алгоритмів для реалізації програмного забезпечення каталогу ресурсів Інтернет.

Загальний обсяг роботи – 31 сторінка.

Ключові слова – вікно, вікно вибору, вікно перегляду, кнопка, пошук, ресурс, каталог, користувач, клас.

ЗМІСТ

[Вступ 5](#_Toc89900041)

[1 АНАЛІЗ СПЕЦИФІКАЦІЇ ВИМОГ ТЕХНІЧНОГО ЗАВДАННЯ 6](#_Toc89900042)

[1.1 Глосарій проекту 6](#_Toc89900043)

[1.2 Призначення розробки 7](#_Toc89900044)

[1.3 Функціональні вимоги 8](#_Toc89900045)

[2 ОБ’ЄКТНО-ОРІЄНТОВАНИЙ АНАЛІЗ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ 9](#_Toc89900046)

[2.1 Діаграма Use Case 9](#_Toc89900047)

[3.1 CRC карти 10](#_Toc89900048)

[3.2 Діаграми класів 13](#_Toc89900049)

[4 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ 14](#_Toc89900050)

[4.1 Реалізація методів 14](#_Toc89900051)

[4.2 Використані класи 22](#_Toc89900052)

[4.3 Інформація про класи 22](#_Toc89900053)

[5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ ВИКОНАННЯ ПРОГРАМИ 24](#_Toc89900054)

[5.1 Інтерфейс застосунку 24](#_Toc89900055)

[Висновок 30](#_Toc89900056)

[Список літератури 31](#_Toc89900057)

**Вступ**

Потрібно розробити об'єктну модель програмного забезпечення вбудованого мікропроцесора пральної машини. Для цього використано програму Visual Studio 2022 для створення об'єктної моделі програми. Мікропроцесор пральної машини був створений на мові С#, використовуючи інтерфейс програмування додатків - Windows Forms.

Об’єктно орієнтоване програмування — одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція.

До недавнього часу проектування інформаційних систем виконувалося, в основному, на інтуїтивному рівні із застосуванням неформалізованих методів, заснованих на практичному досвіді, експертних оцінках і дорогих, експериментальних, перевірках якості функціонування інформаційних систем. Накопичений до теперішнього часу досвід проектування інформаційних систем показує, що це логічно складна, трудомістка і тривала за часом виконання робота, що вимагає високої кваліфікації фахівців, які працюють над цим проектом. До того ж, в процесі створення інформаційні потреби користувачів можуть змінюватися та підлягати уточненню, в свою чергу це ще більше ускладнює розробку таких систем.

Через всі ці фактори з’явилися нові, сучасні програмно-технологічні засоби, такі як CASE-засоби. Вони реалізують CASE-технологію створення і супроводу IC. Цей термін, CASE (Computer Aided Software Engineering), на даний час він має дуже поширене застосування й використовується в дуже широкому сенсі.

* 1. **АНАЛІЗ СПЕЦИФІКАЦІЇ ВИМОГ** **ТЕХНІЧНОГО ЗАВДАННЯ**

1. **Глосарій проекту**

**Форма “MainPage”** – стартова форма, з якої починається взаємодія з іншими програмними можливостями.

**Кнопка “Розпочати”** – кнопка, призначена для переходу до форми з введенням деталей прання.

**Кнопка “Підтвердити”** – кнопка, призначена для переходу до форми з головним функціоналом програми.

**Кнопка “Спорт”** – призначена для змінення режима прання на “Спорт”.

**Кнопка “Лайт”** – призначена для змінення режима прання на “Лайт”.

**Кнопка “Делікатність”** – кнопка, призначена для змінення режима прання на “Делікатність”.

**Кнопка “Повний цикл”** – кнопка, призначена для змінення режима прання на “Повний цикл”.

**Кнопка “Швидке прання”** – кнопка, призначена для змінення режима прання на “Швидке прання”.

**Кнопка “Опис режимів”** – кнопка, призначена для переходу на форму з описом усіх можливих режимів прання.

**Кнопка “Змінити”** – кнопка, призначена для переходу на попередню форму для змінення даних користувача.   
 **Кнопка “Пуск”** – кнопка, призначена для початку прання.

1. **Призначення розробки**

Потрібно розробити об'єктну модель програмного забезпечення вбудованого мікропроцесора пральної машини. Задача полягає в тому, що потрібно розробити повністю функціональну модель пральної машини та розробити програмне забезпечення.

В програмі є стартове вікно де користувач обирає потрібну дію для подальшого функціоналу. Присутня кнопка “ Розпочати ” для переходу до форми з заповненням даних прання. При натисканні на неї відбувається перехід до форми EnterData та вводимо кількість води у літрах необхідну для прання та тип порошку, який буде використовуватися.

Натискаючи на кнопку “Продовжити” у формі EnterData, користувач потрапляє до форми з головним функціоналом програми. Форма має такі елементи, як label, textbox, button, checkbox. Користувач може вибрати режими автоматично за допомогою кнопок, або за допомогою власних налаштувань. При натисканні на кнопку “Пуск” у полі textbox відобразиться поточний режим прання та запуститься таймер на заданий користувачем термін. Замість кнопки “Пуск” з’явиться кнопка “Стоп” для примусової зупинки прання.

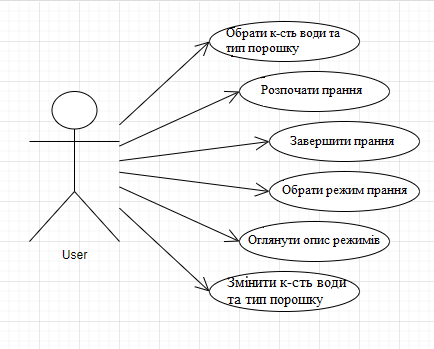
За допомогою кнопки “На головну” користувач може повернутися на головну сторінку програми.

1. **Функціональні вимоги**

Програмне забезпечення підтримує такі вимоги як:

1. Користувач повинен задавати кількість води для прання та тип порошку.
2. Користувач повинен мати можливість перемикатися на різні режими за допомогою відповідних кнопок.
3. Користувач повинен мати можливість зупиняти прання примусово.
4. Користувач повинен мати можливість перемикатися на різні режими за допомогою власних налаштувань.
   1. **ОБ’ЄКТНО-ОРІЄНТОВАНИЙ АНАЛІЗ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ**
5. **Діаграма Use Case**

Діаграма Use Case показує які саме дії може робити користувач в програмі, а саме: обрати кількість води та тип порошку, розпочати та завершити прання, обрати режим прання, оглянути опис режимів, змінити тип порошку.



Мал. 1 Діаграма Use Case

* 1. ОБ’ЄЕКТНО-ОРІЄНТОВАНЕ ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

1. **CRC карти**

CRC картка класу MainPage

В класі MainPage знаходяться кнопки для переходу на інші форми та кнопка закриття форми. (Табл.1).

Методи:

* Конструктор
* private void button1\_Click(object sender, EventArgs e);
* private void button2\_Click(object sender, EventArgs e);

Табл.1 CRC картка класу Resource

|  |  |
| --- | --- |
| Обов’язки | Співробітники |
| Користувач може перейти заповнення користувацької форми button1\_Click() | MainPage, EnterData |
| Користувач може вийти з програми button2\_Click() | MainPage |

CRC картка класу EnterData

В класі EnterData користувач вводить дані в текстове поле для подальшої роботи, наявна кнопка для повернення на головний екран. (Табл.2).

Поля:

* private double sizeWaterCheck;

Методи:

* Конструктор
* private void submitBtn\_Click(object sender, EventArgs e);
* private void button1\_Click(object sender, EventArgs e);
* public bool checkDataInForm();

Табл.2 CRC картка класу Catalog

|  |  |
| --- | --- |
| Обов’язки | Співробітники |
| Користувач може повернутися до головного меню button1\_Click(); | EnterData, MainPage |
| Перевірка даних користувача checkDataInForm (); | EnterData |
| Користувач може перейти на наступну форму submitBtn\_Click (); | EnterData, WashingProgram |

CRC картка класу WashingProgram

В класі WashingProgram користувач може контролювати прання, змінювати режими вречну, або автоматично, перейти до форми з описом режимів. (Табл.3).

Поля:

* private int m;
* private int s;
* public double sizeWater;
* string typeDetergent;
* public double sizeNeedWater;

Методи:

* Конструктор
* private void btnModeSport\_Click(object sender, EventArgs e);
* private void btnModeLight\_Click(object sender, EventArgs e);
* private void btnModeDelicacy\_Click(object sender, EventArgs e);
* private void btnModeFullLoop\_Click(object sender, EventArgs e);
* private void btnModeFast\_Click(object sender, EventArgs e);
* private void btnStart\_Click(object sender, EventArgs e);
* private void button1\_Click(object sender, EventArgs e);
* private void button2\_Click(object sender, EventArgs e);
* private void btnStop\_Click(object sender, EventArgs e);

Табл.3 CRC картка класу WashingProgram

|  |  |
| --- | --- |
| Обов’язки | Співробітники |
| Користувач може перейти до “спорт” режиму прання btnModeSport\_Click(); | WashingProgram |
| Користувач може перейти до “лайт” режиму прання btnModeLight\_Click(); | WashingProgram |
| Користувач може перейти до “Делікатного” режиму прання btnModeDelicacy\_Click(); | WashingProgram |
| Користувач може перейти до режиму прання “ повний цикл” btnModeFullLoop\_Click(); | WashingProgram |
| Користувач може перейти до “швидкого” режиму прання btnModeFast\_Click(); | WashingProgram |
| Користувач може почати прання btnStart\_Click(); | WashingProgram |
| Користувач може примусово зупинити прання btnStop\_Click(); | WashingProgram |
| Користувач може перейти до вікна опису режимів button1\_Click(); | WashingProgram, ModeDescription |
| Користувач може перейти до попереднього вікна button2\_Click(); | WashingProgram, EnterData |

CRC картка класу ModeDescription

Клас ModeDescription слугує для показу описів автоматичних режимів прання. (Табл.4).

Методи:

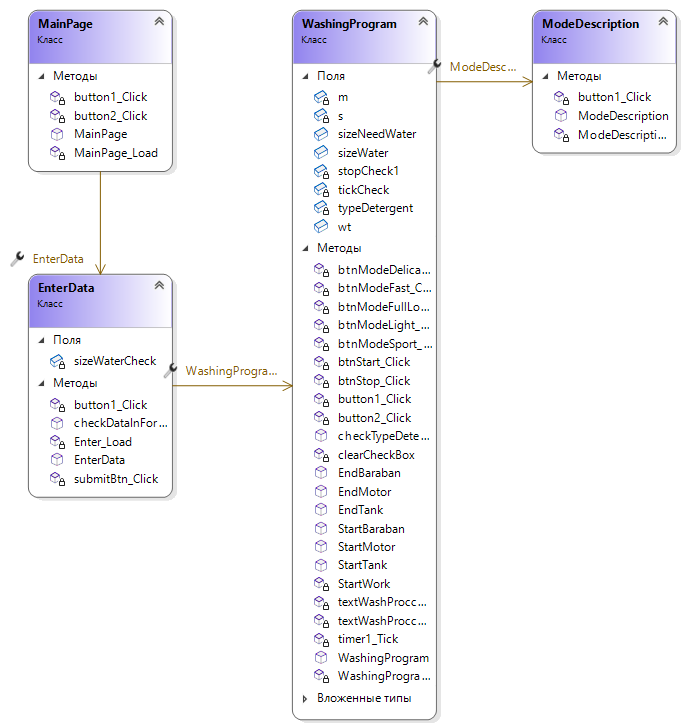
* Конструктор
* private void button1\_Click(object sender, EventArgs e);

Табл.4 CRC картка класу AddResource

|  |  |
| --- | --- |
| Обов’язки | Співробітники |
| Користувач може закрити форму button1\_Click(); | ModeDescription |

1. **Діаграми класів**

На діаграмі класів можемо побачити всі класи які використовуються в програмі, а саме: MainPage, EnterData, WashingProgram, ModeDescription. Завдяки цій діаграмі ми бачимо які поля даних та методи присутні в кожному класі (Мал. 2).



Мал. 2 Діаграми класів

* 1. **РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ**

1. **Реалізація методів**

Методи класу MainPage:

В класі MainPage знаходяться кнопки для переходу на інші форми та кнопка закриття форми.

public partial class MainPage : Form

{

public MainPage()

{

InitializeComponent();

}

private void MainPage\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

EnterData enterData = new EnterData();

enterData.Show();

this.Hide();

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

Методи класу EnterData:

Клас StartPage має конструктор і методи AddResource\_Click, FindResourceButton\_Click, CheckListOfResourceButton\_Click. Кожен з методів активує нове вікно з відповідним призначенням.

public partial class EnterData : Form

{

public EnterData()

{

InitializeComponent();

}

public bool checkDataInForm()

{

if (typeDetergent.Text.toLower()!= "рідкий" && typeDetergent.Text.toLower()!= "сухий"

&& typeDetergent.Text.toLower()!= "комбінований")

{

return false;

}

else if (!double.TryParse(sizeWaterText.Text, out sizeWaterCheck))

{

return false;

}

return true;

}

private void submitBtn\_Click(object sender, EventArgs e)

{

if (!checkDataInForm())

{

MessageBox.Show("З даними щось не так");

}

else

{

WashingProgram washingProgram = new WashingProgram(sizeWaterCheck, typeDetergent.Text);

washingProgram.Show();

this.Hide();

}

}

private double sizeWaterCheck;

private void Enter\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

MainPage mainPage = new MainPage();

mainPage.Show();

this.Hide();

}

}

Методи класу WashingProgram:

В класі WashingProgram користувач може контролювати прання, змінювати режими вречну, або автоматично, перейти до форми з описом режимів.

public partial class WashingProgram : Form

{

private int m, s;

public WaterTank wt;

public WashingProgram(double sizeWater, string typeDetergent)

{

InitializeComponent();

this.sizeWater = sizeWater;

this.typeDetergent = typeDetergent;

timer1.Interval = 500;

m = 0;

s = 0;

label7.Text = "00";

label9.Text = "00";

label8.Visible = true;

}

public double sizeWater = 0;

string typeDetergent;

public double sizeNeedWater = 0;

public string StartMotor()

{

return "\r\nМотор почав працювати.";

}

public string EndMotor()

{

return "\r\nМотор закінчив працювати.";

}

public string StartBaraban()

{

return "\r\nБарабан почав крутитися.";

}

public string EndBaraban()

{

return "\r\nБарабан закінчив крутитися.";

}

public void StartTank()

{

if (checkBoxTypeDetergent3.Checked == true)

{

textWashProccess.Text += "\r\nТип порошку: " + checkBoxTypeDetergent3.Text;

}

else if(checkBoxTypeDetergent2.Checked == true)

{

textWashProccess.Text += "\r\nТип порошку: " + checkBoxTypeDetergent2.Text;

}

else if (checkBoxTypeDetergent1.Checked == true)

{

textWashProccess.Text += "\r\nТип порошку: " + checkBoxTypeDetergent1.Text;

}

}

public string EndTank()

{

return "\r\nПорошок використаний";

}

public class WaterTank

{

public double sizeWater;

public double sizeNeedWater;

public WaterTank(double sizeNeedWater, double sizeWater)

{

this.sizeNeedWater = sizeNeedWater;

this.sizeWater = sizeWater;

}

public string Start(double sizeNeedWater, double sizeWater)

{

if (sizeWater > sizeNeedWater)

return "У баці достатньо води.";

else

{

MessageBox.Show("Недостатня кілкість води, воду було автоматично долито.");

return "У баці достатньо води.";

}

}

public string End()

{

return "Бак пустий.";

}

}

private void clearCheckBox()

{

checkBoxRotationSpeed1.Checked = false;

checkBoxRotationSpeed2.Checked = false;

checkBoxRotationSpeed3.Checked = false;

checkBoxTime1.Checked = false;

checkBoxTime2.Checked = false;

checkBoxTime3.Checked = false;

checkBoxTemp1.Checked = false;

checkBoxTemp2.Checked = false;

checkBoxTemp3.Checked = false;

checkBoxTypeDetergent1.Checked = false;

checkBoxTypeDetergent2.Checked = false;

checkBoxTypeDetergent3.Checked = false;

}

private void StartWork()

{

textWashProccess.Text = null;

textWashProccess.Text = "Початок прання.\r\n";

StartTank();

textWashProccess.Text += StartMotor();

textWashProccess.Text += StartBaraban();

Thread.Sleep(1000);

textWashProccess.Text = textWashProccess.Text + "\r\n" + "Починаємо полоскання.";

Thread.Sleep(1000);

textWashProccess.Text = textWashProccess.Text + "\r\n" + "Починаємо полоскання.";

}

private void btnModeSport\_Click(object sender, EventArgs e)

{

clearCheckBox();

this.sizeNeedWater = 70.0;

checkBoxTemp2.Checked = true;

checkBoxRotationSpeed1.Checked = true;

checkBoxTime1.Checked = true;

checkTypeDetergent();

}

private void btnModeLight\_Click(object sender, EventArgs e)

{

clearCheckBox();

this.sizeNeedWater = 60.0;

checkBoxTemp3.Checked = true;

checkBoxRotationSpeed3.Checked = true;

checkBoxTime1.Checked = true;

checkTypeDetergent();

}

private void btnModeDelicacy\_Click(object sender, EventArgs e)

{

clearCheckBox();

this.sizeNeedWater = 100.0;

checkBoxTemp2.Checked = true;

checkBoxRotationSpeed2.Checked = true;

checkBoxTime2.Checked = true;

checkTypeDetergent();

}

public void checkTypeDetergent()

{

if (typeDetergent == checkBoxTypeDetergent1.Text)

{

checkBoxTypeDetergent1.Checked = true;

}

else if (typeDetergent == checkBoxTypeDetergent2.Text)

{

checkBoxTypeDetergent2.Checked = true;

}

else if (typeDetergent == checkBoxTypeDetergent3.Text)

{

checkBoxTypeDetergent3.Checked = true;

}

}

private void btnModeFullLoop\_Click(object sender, EventArgs e)

{

clearCheckBox();

this.sizeNeedWater = 150.0;

checkBoxTemp2.Checked = true;

checkBoxRotationSpeed2.Checked = true;

checkBoxTime3.Checked = true;

checkTypeDetergent();

}

private void btnModeFast\_Click(object sender, EventArgs e)

{

clearCheckBox();

this.sizeNeedWater = 130.0;

checkBoxTemp1.Checked = true;

checkBoxRotationSpeed1.Checked = true;

checkBoxTime1.Checked = true;

checkTypeDetergent();

}

private int stopCheck1 = 0;

private void btnStart\_Click(object sender, EventArgs e)

{

if ((checkBoxTemp1.Checked != false || checkBoxTemp2.Checked != false

|| checkBoxTemp3.Checked != false) && (checkBoxRotationSpeed1.Checked != false

|| checkBoxRotationSpeed2.Checked != false || checkBoxRotationSpeed3.Checked != false)

&& (checkBoxTime1.Checked != false || checkBoxTime2.Checked != false

|| checkBoxTime3.Checked != false) && (checkBoxTypeDetergent1.Checked != false

|| checkBoxTypeDetergent2.Checked != false || checkBoxTypeDetergent3.Checked != false))

{

btnStop.Show();

if (sizeNeedWater > sizeWater)

{

textWashProccess.Text = null;

textWashProccess.Text += "Статус помилок:\n\nНедостача води";

}

else

{

if (checkBoxTime1.Checked)

{

m = 0;

s = 30;

}

else if (checkBoxTime2.Checked)

{

m = 1;

s = 0;

}

else if (checkBoxTime3.Checked)

{

m = 1;

s = 30;

}

timer1.Start();

btnStart.Hide();

WashingProgram.WaterTank wt = new WashingProgram.WaterTank(this.sizeNeedWater, this.sizeWater);

stopCheck1 = 1;

StartWork();

}

}

else

{

MessageBox.Show("Ви вибрали не всі параметри прання.");

}

}

private void textWashProccess\_TextChanged(object sender, EventArgs e)

{

}

private void WashingProgram\_Load(object sender, EventArgs e)

{

if (typeDetergent == checkBoxTypeDetergent1.Text)

{

checkBoxTypeDetergent1.Checked = true;

}

else if (typeDetergent == checkBoxTypeDetergent2.Text)

{

checkBoxTypeDetergent2.Checked = true;

}

else if (typeDetergent == checkBoxTypeDetergent3.Text)

{

checkBoxTypeDetergent3.Checked = true;

}

label10.Text = "Кількість води: " + sizeWater + "л";

btnStop.Hide();

btnStart.Show();

}

private void textWashProccess\_TextChanged\_1(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

ModeDescription modeDescription = new ModeDescription();

modeDescription.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

EnterData enterData = new EnterData();

this.Hide();

enterData.Show();

}

private void btnStop\_Click(object sender, EventArgs e)

{

if (stopCheck1 == 1)

{

btnStart.Show();

m = 0;

s = 0;

textWashProccess.Text = null;

textWashProccess.Text += "Машина вимкнена.\r\n\r\n";

textWashProccess.Text += "Усі механізми відключені.\r\n";

timer1.Stop();

label7.Text = "00";

label9.Text = "00";

stopCheck1 = 0;

}

btnStop.Hide();

btnStart.Show();

}

private int tickCheck = 0;

private void timer1\_Tick(object sender, EventArgs e)

{

if (label8.Visible)

{

if (m < 10)

label7.Text = "0" + m.ToString();

else

label7.Text = m.ToString();

if (s <= 59 && s > 0)

{

s--;

if (s < 10)

label9.Text = "0" + s.ToString();

else

label9.Text = s.ToString();

}

else

{

if (m < 59 && m > 0)

{

m--;

if (m < 10)

label7.Text = "0" + m.ToString();

else

label7.Text = m.ToString();

s = 59;

label9.Text = "00";

}

else

{

m = 0;

label7.Text = "00";

}

}

label8.Visible = false;

}

else

{

label8.Visible = true;

}

if (label7.Text == "00" && label9.Text == "00" && label8.Visible == true)

{

switch (tickCheck)

{

case 1:

textWashProccess.Text += "\r\nЗапустився віджим.";

break;

case 4:

textWashProccess.Text += "\r\nВода злита.";

break;

case 8:

textWashProccess.Text += "Кінець прання.\r\n";

break;

case 12:

textWashProccess.Text += EndTank() + "\r\n";

break;

case 16:

textWashProccess.Text += EndMotor() + "\r\n";

break;

case 20:

textWashProccess.Text += EndBaraban() + "\r\n";

break;

}

tickCheck++;

}

}

}

Методи класу ModeDescription:

Клас ModeDescription слугує для показу описів автоматичних режимів прання.

public partial class ModeDescription : Form

{

public ModeDescription()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Hide();

}

private void ModeDescription\_Load(object sender, EventArgs e)

{

}

}

1. **Використані класи**
2. MainPage
3. EnterData
4. WashingProgram
5. ModeDescription
6. Program
7. **Інформація про класи**

Клас Program

Цей клас являє собою головну точку входу для додатку, так як в ньому знаходиться метод Main().В методі Main()є метод Run(), який викликає форму класу MainPage.

Клас MainPage

Цей клас являє собою головну форму входу програми. За допомогою кнопок користувач може вибрати план подальших дій.

Клас EnterData

В класі EnterData користувач вводить дані для подальшої роботи, такі як кількість води у літрах та тип порошку.

Клас WashingProgram

Цей клас несе у собі головний функціонал програми: користувач може вибрати режими прання автоматично за допомогою кнопок, або за допомогою власних налаштувань, запустити прання та зупинити його; переглянути форму опису режимів, змінити дані попередньої форми.

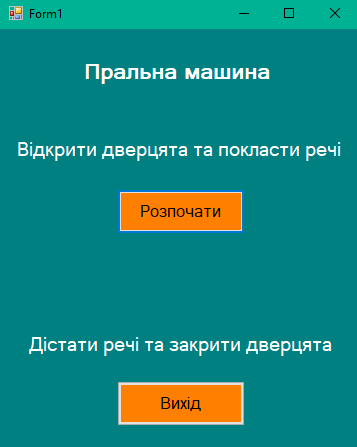
Клас ModeDescription

Цей клас є одним з основних, бо саме тут користувач може створити новий ресурс, указавши повну інформацію про ресурс. AddResource успадковується від класу Form. При виклику цього класу на екрані користувача з’являється вікно на якому користувач може створити новий ресурс, указавши усі дані. Усі методи класу направленні на створення нового ресурсу.

* 1. **ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ ВИКОНАННЯ ПРОГРАМИ**

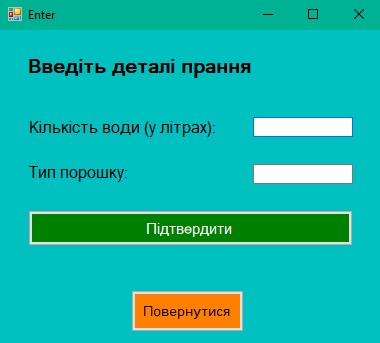
1. **Інтерфейс застосунку**

Інтерфейс головного вікна має основний заголовок, два підзаголовка та дві кнопки для старту програми та для завершення програмної роботи. (Малюнок 3).



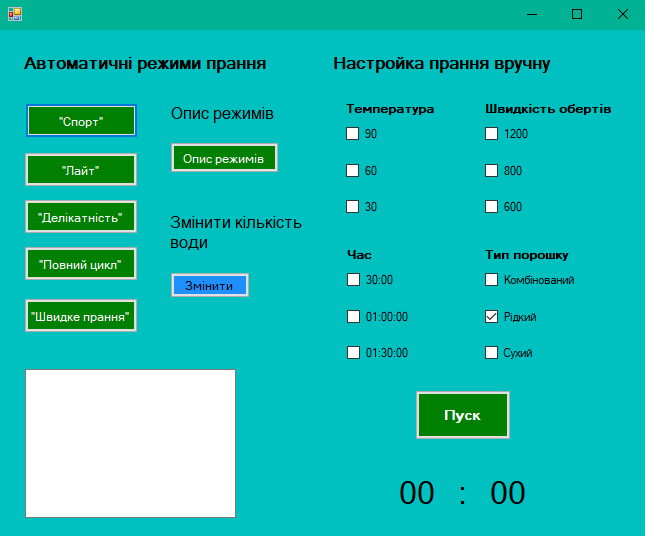
Малюнок 3. Інтерфейс головного вікна

Інтерфейс вводу даних для прання має один заголовок, два текстових поля (для кількості води та типу порошку), дві кнопки для продовження роботи у наступній формі та для повернення на попередню сторінку (Малюнок 4).



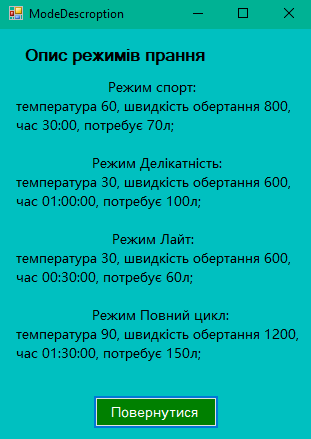
Малюнок 4. Інтерфейс вводу даних для прання

Інтерфейс вікна налаштування прання один заголовок, сім підзаголовків та три заголовки для розробки таймера; дванадцять елементів checkbox для налаштування прання вручну; п’ять кнопок для автоматичного налаштування; кнопка для детального опису автоматичного налаштування; кнопка повернення на попередню форму; кнопка запуску прання; один елемент textbox для виводу результатів прання (Малюнок 5).



Малюнок 5. Інтерфейс вікна налаштування прання та запуску

Інтерфейс вікна опису прання має один заголовок та один підзаголовок для детального опису усіх режимів прання, кнопку повернення на попередній інтерфейс (Малюнок 6).



Малюнок 6. Інтерфейс вікна опису прання

Інтерфейс виконує такі функції:

* + 1. Перехід між вікнами.
    2. Перегляд опцій прання.
    3. Налаштування опцій прання.
    4. Перегляд функціоналу автоматичних кнопок.
    5. Ввід даних.
    6. Вихід з додатку.

**Висновок**

У процесі роботи була розроблена об'єктна модель програмного забезпечення вбудованого мікропроцесора пральної машини та було виконано всі можливі пункти поставленої задачі. Інтерфейс додатку простий, не навантажує систему, має дбайливий вигляд, оптимізовано. Було витрачено багато часу та зусиль для розробки цього додатку. Додаток працює швидко та без усіляких проблем.

**Список літератури**

1. Попівщий В.І ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ Навчально-методичний посібник Частина 1 для студентів ЗДІА спеціальності 121 "Інженерія програмного забезпечення"

Запоріжжя, 2017. 192 с.

1. Бублик В.В. Об’єктно-орієнтоване програмування. Київ : ІТ-книга, 2015. 624 с.
2. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. Вильямс, 2013. 1311 с.
3. Кравець П.О. Об'єктно-орієнтоване програмування. Львів : Видавництво Лівівської політехніки, 2012. 624 с.
4. Гради Буч Объектно-ориентированный анализ и проектирование с примерами приложений. Вильямс, 2008. 720 с.
5. Лафоре Р. Объектно-ориентированное программирование в C++. СПб. : Питер, 2004. 924 с.
6. Jack Purdum Beginning Object-Oriented Programming with C#. Wrox,

2012. 624 p.