

Код using System;

using System.Drawing;

using System.Windows.Forms;

namespace GitConflictSimulator

{

public partial class GitConflictSimulatorForm : Form

{

private int currentStep = 0;

public GitConflictSimulatorForm()

{

InitializeComponent();

this.Load += new System.EventHandler(this.GitConflictSimulatorForm\_Load);

this.btnStart.Click += new System.EventHandler(this.btnStart\_Click);

this.btnPull.Click += new System.EventHandler(this.btnPull\_Click);

this.btnResolve.Click += new System.EventHandler(this.btnResolve\_Click);

this.btnCommit.Click += new System.EventHandler(this.btnCommit\_Click);

this.btnPush.Click += new System.EventHandler(this.btnPush\_Click);

this.btnMerge.Click += new System.EventHandler(this.btnMerge\_Click);

this.btnReset.Click += new System.EventHandler(this.btnReset\_Click);

// -----------------------------------------------------------------

}

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

{

UpdateUI();

}

private void UpdateUI()

{

btnStart.Enabled = false;

btnPull.Enabled = false;

btnResolve.Enabled = false;

btnCommit.Enabled = false;

btnPush.Enabled = false;

btnMerge.Enabled = false;

switch (currentStep)

{

case 0:

if (rtbLog.TextLength == 0)

{

rtbLog.AppendText("Симуляция готова к запуску.\n");

rtbLog.AppendText("Нажмите кнопку '1. Получить ошибку', чтобы начать.\n");

}

btnStart.Enabled = true;

break;

case 1:

rtbLog.AppendText("\n[ШАГ 1] ОШИБКА: Конфликт слияния!\n");

rtbLog.AppendText(" > Принято решение: разрешить конфликт вручную.\n");

btnPull.Enabled = true;

break;

case 2:

rtbLog.AppendText("\n[ШАГ 2] ВЫЗОВ КОНФЛИКТА ЛОКАЛЬНО\n");

rtbLog.AppendText(" > Выполняем команду 'git pull origin main'...\n");

btnResolve.Enabled = true;

break;

case 3:

rtbLog.AppendText("\n[ШАГ 3] РУЧНОЕ РАЗРЕШЕНИЕ\n");

rtbLog.AppendText(" > Редактируем файл, оставляя нужный код.\n");

btnCommit.Enabled = true;

break;

case 4:

rtbLog.AppendText("\n[ШАГ 4] СОХРАНЕНИЕ И КОММИТ\n");

rtbLog.AppendText(" > Выполняем 'git add' и 'git commit'...\n");

btnPush.Enabled = true;

break;

case 5:

rtbLog.AppendText("\n[ШАГ 5] ОТПРАВКА ИЗМЕНЕНИЙ НА СЕРВЕР\n");

rtbLog.AppendText(" > Выполняем 'git push'...\n");

btnMerge.Enabled = true;

break;

case 6:

rtbLog.AppendText("\n[ШАГ 6] ЗАВЕРШЕНИЕ СЛИЯНИЯ\n");

rtbLog.AppendText(" > Ветка успешно слита! Конфликт разрешен.\n");

break;

}

rtbLog.SelectionStart = rtbLog.Text.Length;

rtbLog.ScrollToCaret();

}

private void btnStart\_Click(object sender, EventArgs e) { currentStep = 1; UpdateUI(); }

private void btnPull\_Click(object sender, EventArgs e) { currentStep = 2; UpdateUI(); }

private void btnResolve\_Click(object sender, EventArgs e) { currentStep = 3; UpdateUI(); }

private void btnCommit\_Click(object sender, EventArgs e) { currentStep = 4; UpdateUI(); }

private void btnPush\_Click(object sender, EventArgs e) { currentStep = 5; UpdateUI(); }

private void btnMerge\_Click(object sender, EventArgs e) { currentStep = 6; UpdateUI(); }

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

{

currentStep = 0;

rtbLog.Clear();

UpdateUI();

}

}

}