|  |
| --- |
| ФИЛИАЛ АОО «НАзарбаев интеллектульная школа» НаЗАРБАЕВ ИНТЕЛЛЕКТУАЛЬНАЯ ШКОЛА физико-математического направления города уральск |
| **Проект по информатике**  **«Генерация и проверка надежных паролей»** |
|  |
|  |
|  |
|  |

Выполнил: Хусаинов Марат

|  |
| --- |
|  |

Класс: 9 «А»

Учитель: Тлеккабылова Д.Ж.

Уральск, 2021

Содержание

[Постановка задачи 2](#_Toc101791846)

[Исследование и анализ 2](#_Toc101791848)

[Входные и выходные данные 4](#_Toc101791849)

[Разработка решения 5](#_Toc101791850)

[Разработка 12](#_Toc101791851)

[Листинг программы (с комментариями) 13](#_Toc101791852)

[Тестирование 17](#_Toc101791853)

[Руководство пользователя 20](#_Toc101791854)

[Оценивание 21](#_Toc101791855)

# Постановка задачи

# Постановка задачи: (Формулировка условия задачи, текущее положение, постановка проблемы) В наше время, информационные технологии и интернет являются неотъемлемой частью жизни каждого человека. А поэтому и кибербезопасность обретает огромное значение для нас. Для того чтобы обеспечить защитой свои аккаунты и данные в интернете или на компьютере нам нужны надежные пароли, но, к сожалению, не все пользователи их используют

**Цель:** Создать приложение для проверки надежности пароля и генерации случайных надежных паролей

**Актуальность:**, Мое приложение будет полезно абсолютно всем людям которые пользуются современными технологиями и интернетом независимо от возраста.

**Задачи:** (опишите, какие действия вы планируете выполнить, чтобы добиться цели, минимум 3 задачи)

1. Вывести цель и актуальность проекта
2. Собрать информацию для этой задачи и собрать статистику
3. Определить входные и выходные данные, нужные для решения задачи формулы и правила
4. Составить блок-схемы и создать интерфейс
5. Разработать приложение на языке С# для решения задачи
6. Протестировать написанную программу
7. Создать руководство по использованию приложения
8. Вывод и оценка проекта

# Исследование и анализ

Опишите один вид сбора информации для реализации этой задачи. Обоснуйте, почему Вы использовали этот вид сбора данных. Вставьтевизуальное представление результатов исследования и анализа (диаграммы, таблицы, графики). Сделайте вывод.

Для сбора информации о проблеме с кибербезопасностью я решил провести опрос среди учащихся НИШ. Этот метод сбора информации хорош тем, что мы сможем запросто получить довольно точную статистику об этой проблеме.

В этом опросе я задал всего три вопроса, и тут вы можете видеть результаты:

Chart, pie chart

Description automatically generated

Chart, pie chart

Description automatically generated

Chart, pie chart

Description automatically generated

По этим диаграммам, мы выяснили, что большинство людей не особо задумываются над надежностью паролей, и больше трети подвергаются интернет-мошенничеству. А это еще раз подчеркивает актуальность моего приложения.

А теперь, давайте начнем изучать решение этой проблемы. Сначала, нам нужно узнать об условиях при которых пароль является надежным. Я вывел для этой задачи всего 5 условий:

1. В пароле должна быть хотя бы одна заглавная буква
2. В пароле должна быть хотя бы одна строчная буква
3. В пароле должна быть хотя бы одна цифра
4. В пароле должен быть хотя бы один символ
5. Пароль должен состоять минимум из 10 символов

При проверке пароля мы будем всего лишь проверять его на все эти условия, а при генерации будем создавать новый, который будет соответствовать всем этим условиям

Также, в этой задаче нам потребуются 2 самых главных подпрограмм, и одна вспомогательная. 2 главные подпрограммы являются проверкой надежности пароля и генерацией надежных паролей соответственно. А вспомогательная подпрограмма помогает нам выявить есть ли определенный символ в строке. Это нам очень поможет в проверке всех вышеперечисленных условий.

### Входные и выходные данные

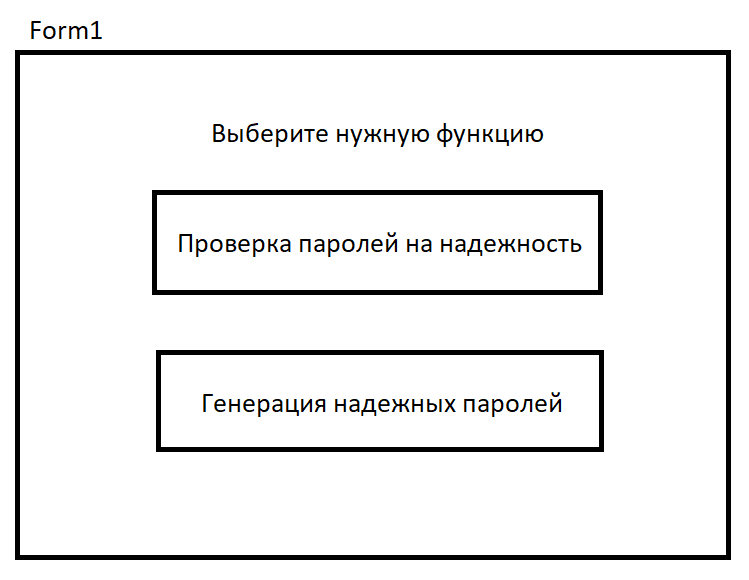
Во входные данные мое приложение должно принимать только пользовательский пароль при выборе первой функции. А если он выбрал вторую функцию, то никаких входных данных не нужно.

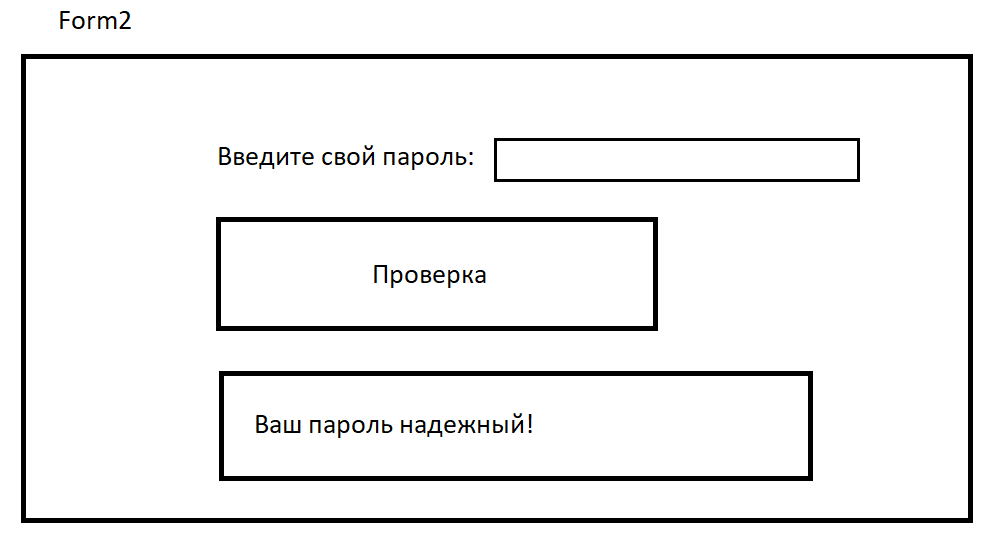
А в выходные данные мое приложение может выдавать:

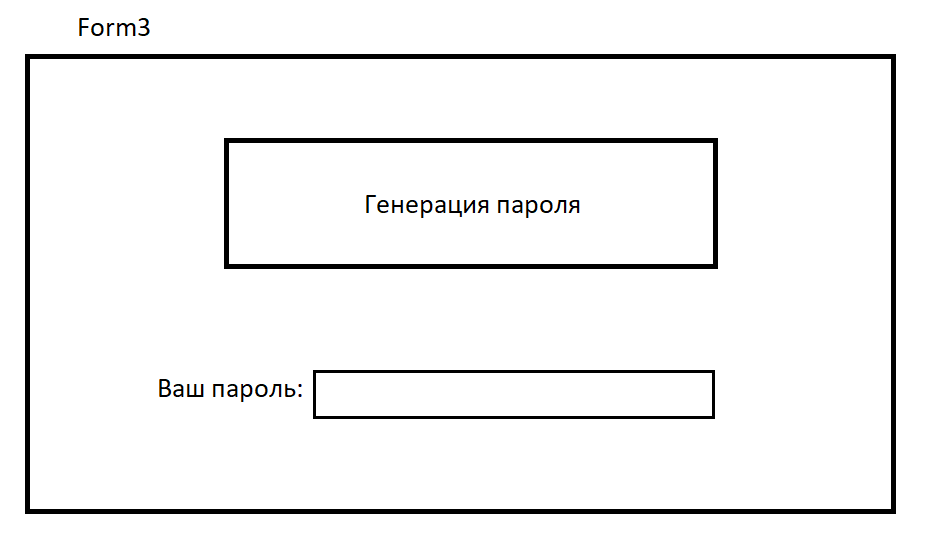
1. Вердикт о надежности пароля (при выборе первой функции)
2. Сгенерированный пароль (при выборе второй функции).

# Разработка решения

Прототипы:







Метод button1\_Click в Form2

Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram, schematic

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

­­Метод finder

Diagram

Description automatically generated

Метод button1\_Click в Form1

Diagram

Description automatically generated

Метод button2\_Click в Form1

Diagram

Description automatically generated

Метод button1\_Click в Form3

Diagram

Description automatically generated

# Разработка

После разработки интерфейса (в ООП С#) вставьте скриншоты в отчет, подпишите каждый скриншот.

Интерфейс Form1

Graphical user interface, application

Description automatically generated

Интерфейс Form2

Graphical user interface, application, Teams

Description automatically generated

Интерфейс Form3

Graphical user interface, application

Description automatically generated

# Листинг программы (с комментариями)

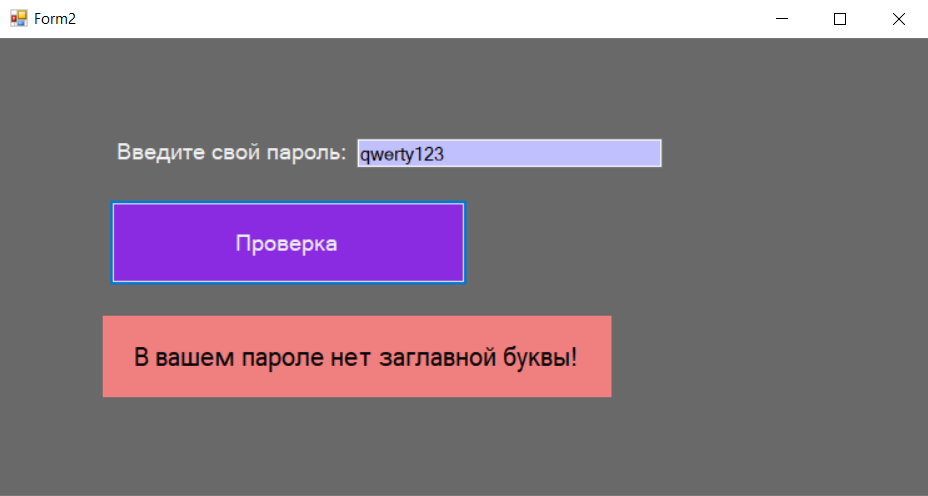
Вставляйте фрагменты программного кода и пишите к ним комментарии для объяснения хода программы.

|  |  |
| --- | --- |
| **Фрагмент программного кода** | **Комментарий** |
| private void button1\_Click(object sender, EventArgs e)  {  Form2 form2 = new Form2();  form2.Show();  } | Создание второй формы и ее показ при нажатии на первую кнопку первой формы |
| private void button2\_Click(object sender, EventArgs e)  {  Form3 form3 = new Form3();  form3.Show();  } | Создание третьей формы и ее показ при нажатии на вторую кнопку |
| private bool finder(char elem, string password)  {  for (int i = 0; i < password.Length; i++)  {  if (password[i] == elem)  {  return true;  }  }  return false;  } | Метод для нахождения символа в строке. В нем реализован цикл который проходится по всей строке и сравнивает каждый символ с нужным. При нахождении возвращает true, иначе false |
| private void button1\_Click(object sender, EventArgs e)  {  const string small = "abcdefghijklmnopqrstuvwxyz", big = "ABCDEFGHIJKLMNOPQRSTUVWXYZ",  num = "1234567890", symbols = "\_@#$%!";  bool c1 = false, c2 = false, c3 = false, c4 = false; | Функция вызываемая при нажатии первой кнопки второй формы. Объявление нужных переменных (константы со всеми нужными символами и булевые переменные) |
| string password = textBox1.Text; | Получение введенного пароля |
| if (password == "")  {  MessageBox.Show("Пожалуйста, введите пароль");  return;  } | Вывод сообщения если пользователь не ввел пароль |
| for (int i = 0; i < password.Length; i++)  {  if (finder(password[i], big))  {  c1 = true;  }  if (finder(password[i], small))  {  c2 = true;  }  if (finder(password[i], symbols))  {  c3 = true;  }  if (finder(password[i], num))  {  c4 = true;  }  } | Проверка пароля циклом, изменение нужных переменных при нахождении соответствующего символа |
| label2.BackColor = Color.LightCoral;  label2.ForeColor = Color.Black; | Изменение цвета надписи |
| if (!c1){  label2.Text = "В вашем пароле нет заглавной буквы!";  }  else if (!c2){  label2.Text = "В вашем пароле нет строчной буквы!";  }  else if (!c4){  label2.Text = "В вашем пароле нет цифры!";  }  else if (!c3){  label2.Text = "В вашем пароле нет другого символа!";  }  else if (password.Length < 10)  {  label2.Text = "Ваш пароль слишком короткий!(должно быть 10 символов)";  }  else  {  label2.Text = "Ваш пароль надежный!";  label2.BackColor = Color.LimeGreen;  }  label2.Visible = true; | Вывод нужного сообщения о несоблюдении безопасности пароля или сообщение о безопасности пароля в объекте label. Делаем надпись видимой |
| private void button1\_Click(object sender, EventArgs e)  {  string all = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890\_@#$%!";  string small = "abcdefghijklmnopqrstuvwxyz", big = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  string nums = "1234567890", symbols = "\_@#$%!"; | Функция вызываемая при нажатии первой кнопки третьей формы. Объявление переменных с нужными символами |
| string a = "";  char elem;  int elem\_num; | Переменная для хранения сгенерированного пароля, выбранного символа и случайного целого числа соответственно |
| Random random = new Random(DateTime.Now.Second); | Создание генератора случайных чисел |
| elem\_num = random.Next(0, big.Length);  elem = big[elem\_num];  a = a + elem; | Получение одного символа заглавной буквы и добавление в переменную с паролем |
| elem\_num = random.Next(0, small.Length);  elem = small[elem\_num];  a = a + elem; | Получение одного символа строчной буквы и добавление в переменную с паролем |
| elem\_num = random.Next(0, symbols.Length);  elem = symbols[elem\_num];  a = a + elem; | Получение одного специального символа и добавление в переменную с паролем |
| elem\_num = random.Next(0, nums.Length);  elem = nums[elem\_num];  a = a + elem; | Получение одной цифры и добавление в переменную с паролем |
| for (int i = 0; i < 6; i++)  {  elem\_num = random.Next(0, all.Length);  elem = all[elem\_num];  a = a + elem;  } | Заполнение пароля шестью случайными символами (любой категории) |
| textBox1.Text = a; | Вывод переменной с паролем в компонент TextBox |
| textBox1.Visible = true;  label1.Visible = true; | Делаем видимыми компоненты TextBox и Label |

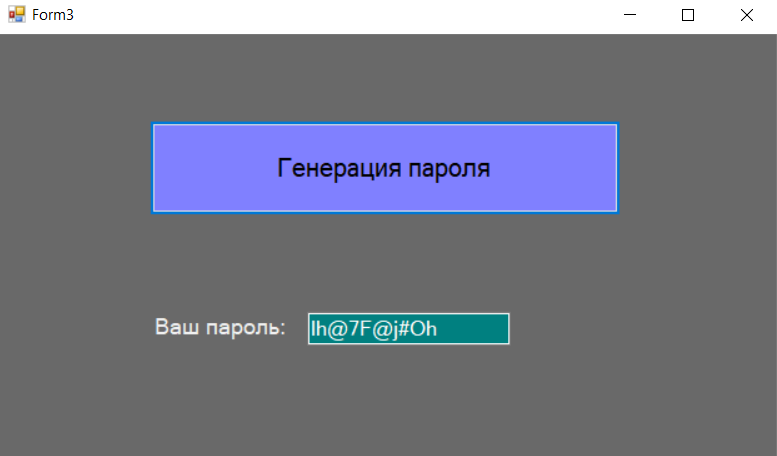
# Тестирование

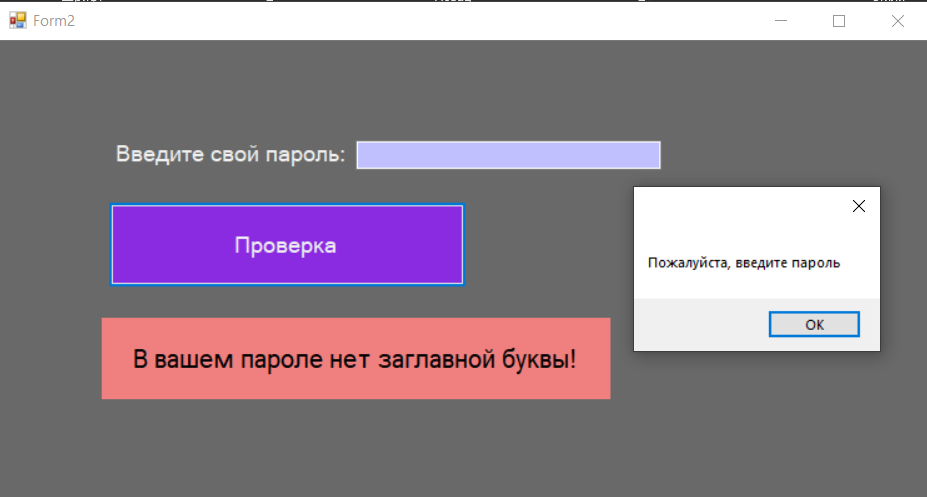
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Цель  тестирования | Использованные  тестовые данные | Ожидаемый  результат | Фактический  результат |
| 1 | Получить правильный вердикт при ненадежном пароле | qwerty123 | Получение вердикта об отсутствии заглавной буквы | Вердикт об отсутствии заглавной буквы, ожидаемый результат |
| 2 | Получить корректно сгенерированный пароль | - | Получения пароля соответствующий всем критериям | Был получен надежный пароль, ожидаемый результат |
| 3 | Не вводить пароль и получить нужное сообщение | - | Выход нужного сообщения | Получено сообщение о необходимости ввода пароля |
| 4 | Получить вердикт о надежности пароля | Nis\_uralsk1! | Выходит вердикт о надежности пароля | Выведена надпись о надежности пароля, ожидаемый результат |

Доказательство к тесту 1

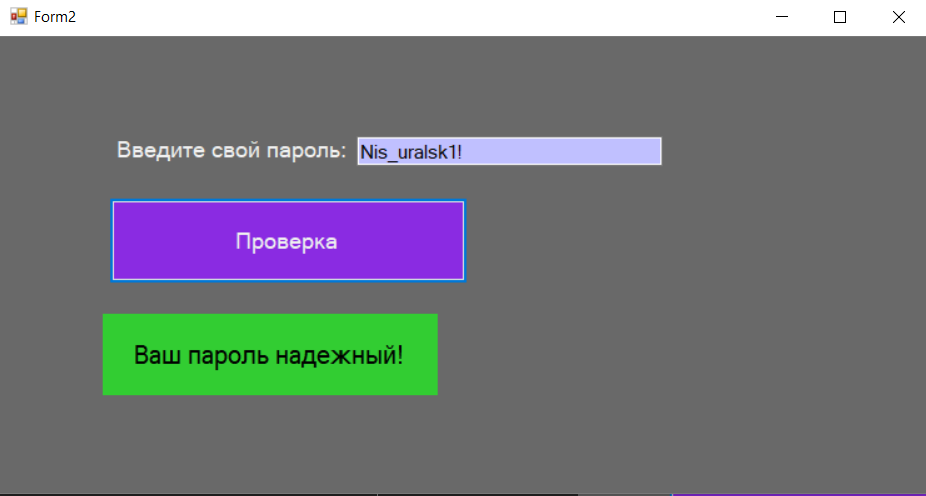


Доказательство к тесту 2

 Доказательство к тесту 3



Доказательство к тесту 4

****

# Руководство пользователя

Содержание

(пример,

Как изучить теоретический материал?

Как решить задачу?

Как пройти тестирование?

Как узнать о программе?

…

Далее описываете полное руководство по использованию Вашего приложения. Даете пошаговую инструкцию для потенциального пользователя, который не всегда является ИКТ компетентным. Обязательно приложите скриншоты.

Как изучить теоретический материал?

1.Откройте программу …

2.Выберите команду … или Перейдите на закладку…

(скриншот)

3.Изучите предложенный материал.

(скриншот)

И т.д.)

# Оценивание

В оценивании ответьте на вопросы:

1. Смогли ли Вы достичь поставленной цели?
2. Что у Вас хорошо получилось (минимум два пункта/аспекта)
3. Какие сложности возникли при разработке проекта? Удалось ли их решить? Каким способом?
4. Всё ли задуманное удалось выполнить?
5. Как оценили проект ваши одноклассники?
6. Каким образом можно улучшить функциональность Вашего проекта? Что бы Вы хотели добавить в будущем?

Использованные источники:

Перечислите источники, которыми Вы воспользовались при выполнении проекта с помощью нумерованного списка