

Краевое государственное автономное профессиональное образовательное
учреждение «Пермский авиационный техникум им. А.Д. Швецова»

ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ №1-13

дисциплина: МДК.01.02 Поддержка и тестирование программных модулей

Преподаватель _____ А.В. Басалаев
(подпись, дата)

Студентка, гр. ИСП-20-2 _____ О.Е.Аликина
(подпись, дата)

Содержание

Отчёт по практической работе №1.....	3
Отчёт по практической работе №2.....	6
Отчёт по практической работе №3.....	9
Отчёт по практической работе №4.....	12
Отчёт по практической работе №5.....	14
Отчет по практической работе №6.....	16
Отчет по практической работе №7.....	19
Отчёт по практической работе №8.....	24
Отчёт по практической работе №9.....	26
Отчёт по практической работе №10.....	28
Отчёт по практической работе №11.....	31
Отчёт по практической работе №12.....	39
Отчёт по практической работе №13.....	42

					АТПР.090207.20.201			
Изм	Лист	№ документа	Подпись	Дата				
Разраб.	Аликина				Отчет по практическим работам №1-13	Литера	Лист	Листов
Проверил	Басалаев					У	2	47
Н. контр.					ИСП-20-2			
Утв.								

Отчёт по практической работе №1

Тестирование «белым ящиком»

Цель работы: изучить методы тестирования логики программы, формализованные описания результатов тестирования и стандарты по составлению схем программ.

Задание 1:

Разработали программу

```
Ссылка: 0
static void Main(string[] args)
{
    Console.WriteLine("Введите стороны треугольника");
    string a1 = Console.ReadLine();
    string a2 = Console.ReadLine();
    string a3 = Console.ReadLine();
    string tip = "";
    double s = 0;
    if (int.TryParse(a1, out int a) && int.TryParse(a2, out int b) && int.TryParse(a3, out int c)) {
        if (!(a <= 0 || b <= 0 || c <= 0))
        {
            if (a+b>c && b+c>a && a + c > b)
            {
                if (a == b && b == c)
                {
                    tip = "Равносторонний";
                }
                else if (a == b || a == c || b == c)
                {
                    tip = "Равнобедренный";
                }
                else
                {
                    tip = "Разносторонний";
                }
                int p = (a + b + c)/2;
                s = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
            }
            else
            {
                Console.WriteLine("сумма 2 сторон треугольника не могут быть меньше 3ей стороны");
                Console.Read();
                return;
            }
        }
        else
        {
            Console.WriteLine("Стороны треугольника не могут быть равны 0 или отрицательными");
            Console.Read();
            return;
        }
    }
    else
    {
        Console.WriteLine("Длины сторон должны быть числами");
        Console.Read();
        return;
    }
}
Console.WriteLine("Тип треугольника: "+tip);
Console.WriteLine("Площадь: " + s);
Console.Read();
```

		Аликина			АТПР.090207.20.201	Лист
						3
Изм	Лист	№ документа	Подпись	Дата		

Задание 2:

A	B	C	Ожидаемы й результат	Объект проверки
8	8	8	Тип = равносторонний Площадь = 27,712..	Тип = равносторонний Площадь = 27,712..
5	6	7	Разностор онний Площадь = 14,69..	Разностор онний Площадь = 14,69..
5	5	6	Тип = Равнобедренный Площадь = 12	Тип = Равнобедренный Площадь = 12

Задание 3:

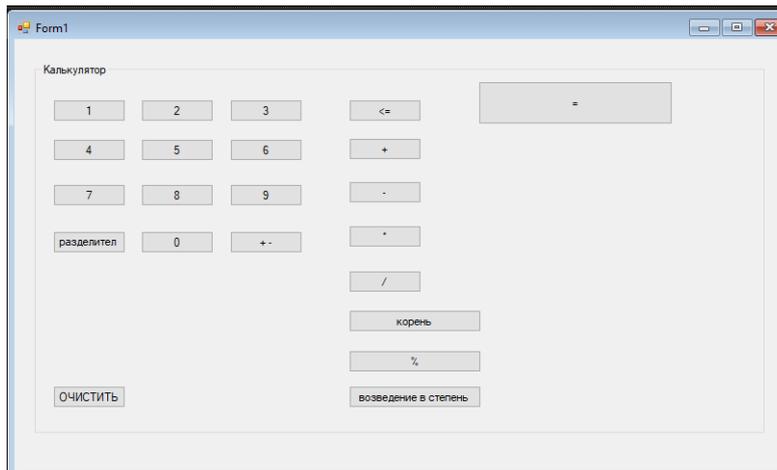
Отчёт по практической работе №2

Тестирование «черным ящиком»

Цель работы: изучить метод тестирования «Черным ящиком».

Задание 1. Написать калькулятор с небольшими багами.

Написали программу калькулятор с багами: не получается возвести в степень и при нажатии равно появляется то 0 то 1.



```
public Form1()
{
    InitializeComponent();
    label1.Text = "0";
}
String chislo1;
int kod_Oper;
//класс 1
public void Esli()
{
    if (label1.Text[0] == '0' && (label1.Text.IndexOf(",") != -1))
        label1.Text = label1.Text.Remove(0, 1); // если 0 в старш
    if (label1.Text[0] == '-')
        if (label1.Text[1] == '0' && (label1.Text.IndexOf(",") != -1))
            label1.Text = label1.Text.Remove(1, 1); // е
}
//класс 1
private void button1_Click(object sender, EventArgs e)
{
    label1.Text += "1"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
//класс 1
private void button2_Click(object sender, EventArgs e)
{
    label1.Text += "2"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
//класс 1
private void button3_Click(object sender, EventArgs e)
{
    label1.Text += "3"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
//класс 1
private void button4_Click(object sender, EventArgs e)
{
    label1.Text += "4"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
//класс 1
private void button5_Click(object sender, EventArgs e)
{
    label1.Text += "5"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
//класс 1
private void button6_Click(object sender, EventArgs e)
{
    label1.Text += "6"; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверка нули в старших разрядах
}
```

```
private void button8_Click(object sender, EventArgs e)
{
    chislo1 = label1.Text; //в переменную записывается предыдущее значение label1
    label1.Text = "0"; // значение label1 очищается
    kod_Oper = 4; // в переменную записывается код операции
}
private void button9_Click(object sender, EventArgs e)
{
    switch (kod_Oper)
    {
        case 1:
            label1.Text = Convert.ToString(double.Parse(chislo1) + double.Parse(label1.Text));
            break;
        case 2:
            label1.Text = Convert.ToString(double.Parse(chislo1) - double.Parse(label1.Text));
            break;
        case 3:
            label1.Text = Convert.ToString(double.Parse(chislo1) * double.Parse(label1.Text));
            break;
        case 4:
            label1.Text = Convert.ToString(double.Parse(chislo1) / double.Parse(label1.Text));
            break;
        case 5:
            label1.Text = Convert.ToString(Math.Sqrt(double.Parse(chislo1)));
            break;
        case 6:
            label1.Text = Convert.ToString(double.Parse(chislo1) X double.Parse(label1.Text));
            break;
        case 7:
            label1.Text = Convert.ToString(Math.Pow(double.Parse(chislo1), double.Parse(label1.Text)));
            break;
        default:
            break;
    }
}
private void button20_Click(object sender, EventArgs e)
{
    chislo1 = label1.Text; //в переменную записывается предыдущее значение label1
    label1.Text = "0"; // значение label1 очищается
    kod_Oper = 4; // в переменную записывается код операции
}
private void button21_Click(object sender, EventArgs e)
{
    chislo1 = label1.Text; //в переменную записывается предыдущее значение label1
    label1.Text = "0"; // значение label1 очищается
    kod_Oper = 5; // в переменную записывается код операции
}
private void button22_Click(object sender, EventArgs e)
{
    chislo1 = label1.Text; //в переменную записывается предыдущее значение label1
    label1.Text = "0"; // значение label1 очищается
    kod_Oper = 6; // в переменную записывается код операции
}
private void button23_Click(object sender, EventArgs e)
{
    chislo1 = label1.Text; //в переменную записывается предыдущее значение label1
    label1.Text = "0"; // значение label1 очищается
    kod_Oper = 7; // в переменную записывается код операции
}
```

```

private void button8_Click(object sender, EventArgs e)
{
    label1.Text += 8; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверяющей нули в старших разрядах
}

private void button9_Click(object sender, EventArgs e)
{
    label1.Text += 9; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверяющей нули в старших разрядах
}

private void button10_Click(object sender, EventArgs e)
{
    label1.Text += 0; //к тексту добавляется символ 1
    Esli(); //вызов подпрограммы, проверяющей нули в старших разрядах
}

private void button11_Click(object sender, EventArgs e)
{
    label1.Text = "0";
}

private void button14_Click(object sender, EventArgs e)
{
    if (!label1.Text.Contains(",")) label1.Text += ",";
    Esli();
}

private void button12_Click(object sender, EventArgs e)
{
    int lenght = label1.Text.Length - 1; //уменьшает текущую длину строки на 1
    string text = label1.Text;
    label1.Text = ""; //очищает содержимое label1
    for (int i = 0; i < lenght; i++)
    {
        label1.Text = label1.Text + text[i]; //записывает в label1 текст без последней цифры
    }
}

private void button13_Click(object sender, EventArgs e)
{
    if (label1.Text[0] == '-') //проверяет наличие знака "-" у числа
        label1.Text = label1.Text.Remove(0, 1); //если он есть, то удаляет его
    else
        label1.Text = "-" + label1.Text; //если нет, то дописывает
}

```

Задание 2. Обменяться программой с другими студентами. Провести тестирование и написать отчет в тетради.

Название теста	Описание сценария	Входные данные	Выходные данные	Удачное/неудачное тестирование	Предложения по исправлению найденных ошибок.	Пожелания пользователей
Функция деления	Деление чисел, которые нацело не делятся. Проверка результата	Первая переменная=9 Вторая переменная=7	Результат=1	Неудачное	Поменять типы int на тип double или float, чтобы могли быть в результате числа после запятой	Сделать возможность повторного действия на знак равно

Функция возведения в степень	Возведение в степень положительного целого числа	Переменная = 6	Результат 0	Неудачное	Исправить ошибку при возведении в степень	При выборе действия, чтобы результат не становился равным 0.
Функция сложения	Сложения чисел разных знаков	Переменные 5 и -6	Результат -1	Удачное	Все хорошо	Нет
Функция вычитания	Вычитание целых положительных чисел	Числа 7 и 10	Результат -3	Удачное	Все хорошо	Нет
Функция умножения	Умножения отрицательных чисел	Числа -9 и -6	Результат 54	Удачное	Все хорошо	Нет

Вывод: потренировались в тестировании программы, разработанной другим человеком, научились искать ошибки в чужой программе и выяснять причину ошибки.

Отчёт по практической работе №3

Отладка в Visual Studio

Ход работы:

Теория:

Отладчик — это узкоспециализированное средство разработки, которое присоединяется к работающему приложению и позволяет проверять код. В документации по отладке для Visual Studio именно это обычно подразумевается под отладкой.

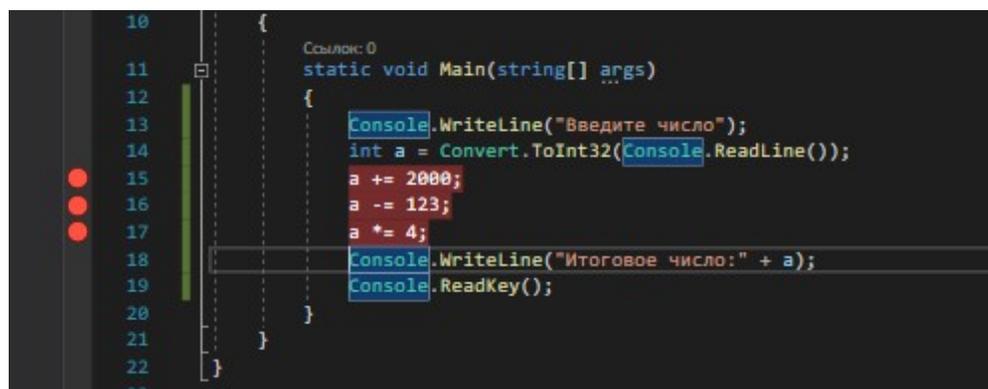
Отладка может иметь разные значения, но в первую очередь он означает устранение ошибок в коде.

Отладка приложения обычно означает запуск приложения с подключенным отладчиком (то есть в режиме отладчика). При этом в отладчике доступно множество способов наблюдения за выполнением кода.

Точки останова — это один из самых простых и важных компонентов надежной отладки. Точка останова указывает, где Visual Studio следует приостановить выполнение кода, чтобы вы могли проверить значения переменных или поведение памяти либо выполнение ветви кода.

Практика:

Первым делом пробывали режим отладки и точки останова



```
10 {
11     Console.WriteLine("Введите число");
12 }
13
14 int a = Convert.ToInt32(Console.ReadLine());
15 a += 2000;
16 a -= 123;
17 a *= 4;
18 Console.WriteLine("Итоговое число:" + a);
19 Console.ReadKey();
20 }
21
22
23 }
```

Смотрим изменения переменных в ходе выполнения программы

		Аликина			АТПР.090207.20.201	Лист
						9
Изм	Лист	№ документа	Подпись	Дата		

```
12
13 Console.WriteLine("Введи число");
14 int a = Convert.ToInt32(Console.ReadLine());
15 a += 2000;
16 a -= 123;
17 a *= 4;
18 Console.WriteLine("Итог: " + a);
19 Console.ReadKey();
```

Вставили код из примера программы, и исправили ошибки, как показано на сайте, постепенно избавляясь от появившихся ошибок.

Первые ошибки исправили с помощью Alt+Delete и подсказки добавить директорию.

```
using System;
using System.Linq;
using System.Text;
using System.IO;
using System.Runtime.Serialization.Json;
using System.Runtime.Serialization;

namespace ConsoleApp1
{
    static void Main(string[] args)
    {
        var localDB = LoadRecords();
        string data = GetJsonData();

        User[] users = ReadToObject(data);

        UpdateRecords(localDB, users);

        for (int i = 0; i < users.Length; i++)
        {
            List<User> result = localDB.FindAll(delegate (User u) {
                return u.lastname == users[i].lastname;
            });
            foreach (var item in result)
            {
                Console.WriteLine($"Matching Record, got name={item.firstname}, lastname={item.lastname}, age={item.totalpoints}");
            }
        }

        Console.ReadKey();
    }

    // Deserialize a JSON stream to a User object.
    public static User[] ReadToObject(string json)
    {
        User deserializedUser = new User();
        User[] users = { };
        MemoryStream ms = new MemoryStream(Encoding.UTF8.GetBytes(json));
        DataContractJsonSerializer ser = new DataContractJsonSerializer(users.GetType());

        try
        {
            users = ser.ReadObject(ms) as User[];
        }
        catch (SerializationException)
        {
            Console.WriteLine("Give user some info or instructions, if necessary");
            // Take appropriate action for your app
        }

        ms.Close();
        return users;
    }
}
```

Следующую ошибку исправили путём смены string на int, так как проблема была в несовместимости типов данных

```
using System;
using System.Linq;
using System.Text;
using System.IO;
using System.Runtime.Serialization.Json;
using System.Runtime.Serialization;

namespace ConsoleApp1
{
    [DataContract]
    internal class User
    {
        [DataMember]
        internal string firstname;

        [DataMember]
        internal string lastname;

        [DataMember]
        // internal double points;
        internal int points;

        [DataMember]
        internal int totalpoints;
    }
}
```

Добавили условие в if, чтобы не было исключения

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		10

```

return db;
}
// ссылка: 1
public static void UpdateRecords(List<User> db, User[] users)
{
    Debug.Assert(db != null);
    Debug.Assert(users != null);
    bool existingUser = false;

    for (int i = 0; i < users.Length; i++)
    {
        foreach (var item in db)
        {
            if (item.lastname == users[i].lastname && item.firstname == users[i].firstname)
            {
                existingUser = true;
                item.points = users[i].points;
                item.totalpoints += users[i].points;
            }
        }
        // Also, add a using statement for System.Diagnostics at the start of the file.
        Debug.Assert(users[i].firstname != null);
        Debug.Assert(users[i].lastname != null);

        if (existingUser == false && users[i].firstname != null && users[i].lastname != null)
        {
            User user = new User();
            user.firstname = users[i].firstname;
            user.lastname = users[i].lastname;

            db.Add(user);
        }
    }
}

```

Также мы добавили try catch чтобы при ошибке, приложение не закрывалось, а выдавало ошибку

```

// Deserialize a JSON stream to a User object.
public static User[] ReadToObject(string json)
{
    User deserializedUser = new User();
    User[] users = { };
    MemoryStream ms = new MemoryStream(Encoding.UTF8.GetBytes(json));
    DataContractJsonSerializer ser = new DataContractJsonSerializer(users.GetType());

    try
    {
        users = ser.ReadObject(ms) as User[];
    }
    catch (SerializationException)
    {
        Console.WriteLine("Give user some info or instructions, if necessary");
        // Take appropriate action for your app
    }

    ms.Close();
    return users;
}

```

Выводы: изучили отладку, точки останова и методы исправления ошибок. Потренировались на практике как исправлять ошибки.

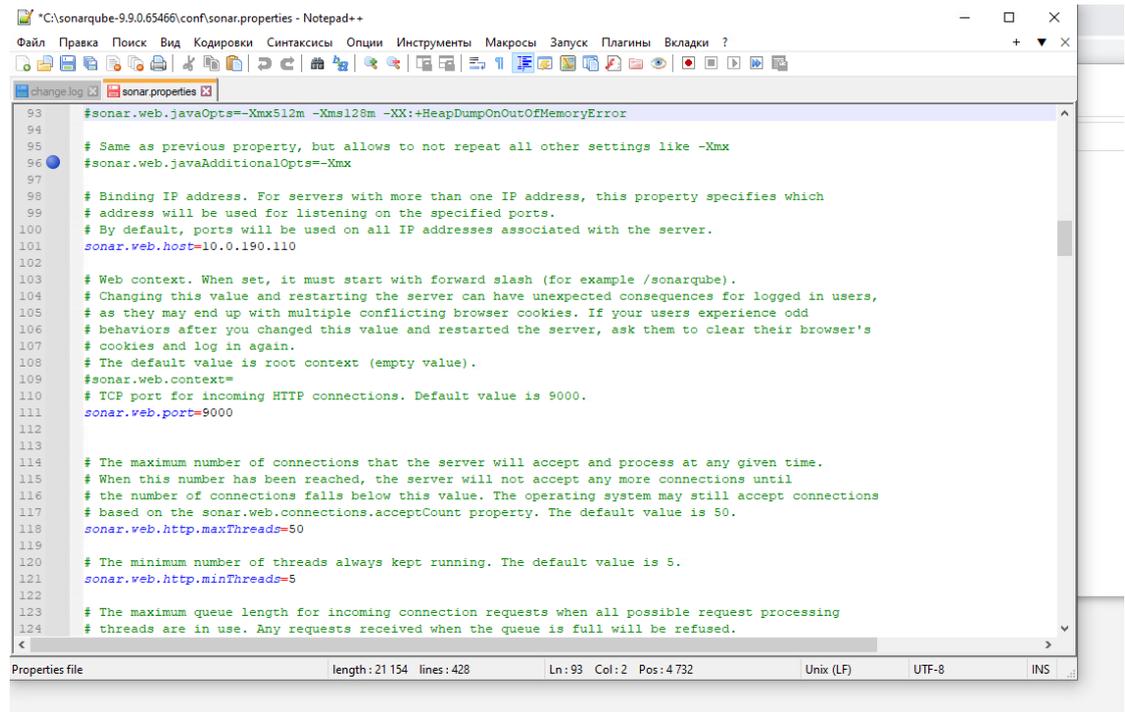
		Аликина			АТПР.090207.20.201	Лист
						11
Изм	Лист	№ документа	Подпись	Дата		

Отчёт по практической работе №4

SonarQube. Изучение платформы с открытым исходным кодом для непрерывного анализа

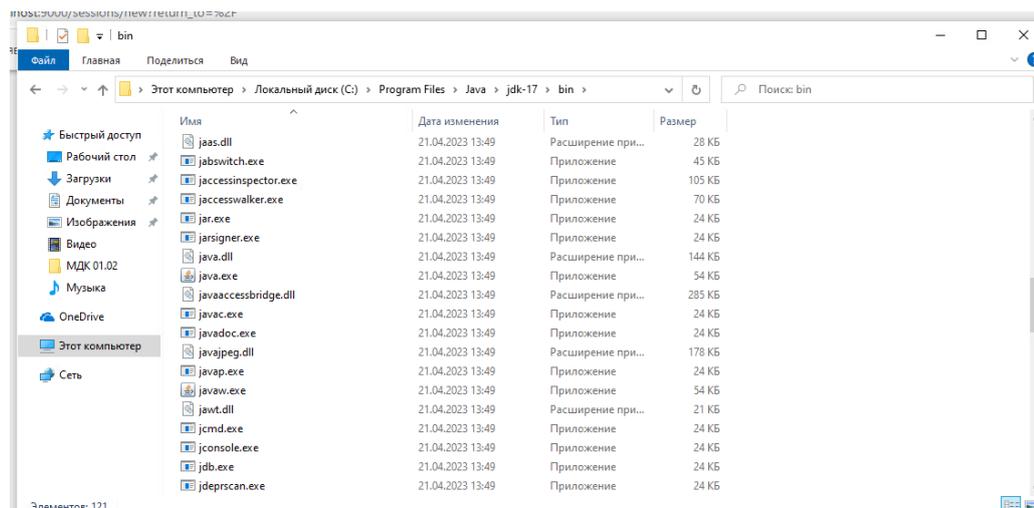
Ход работы:

Скачали SonarQube и настроили sonar.properties



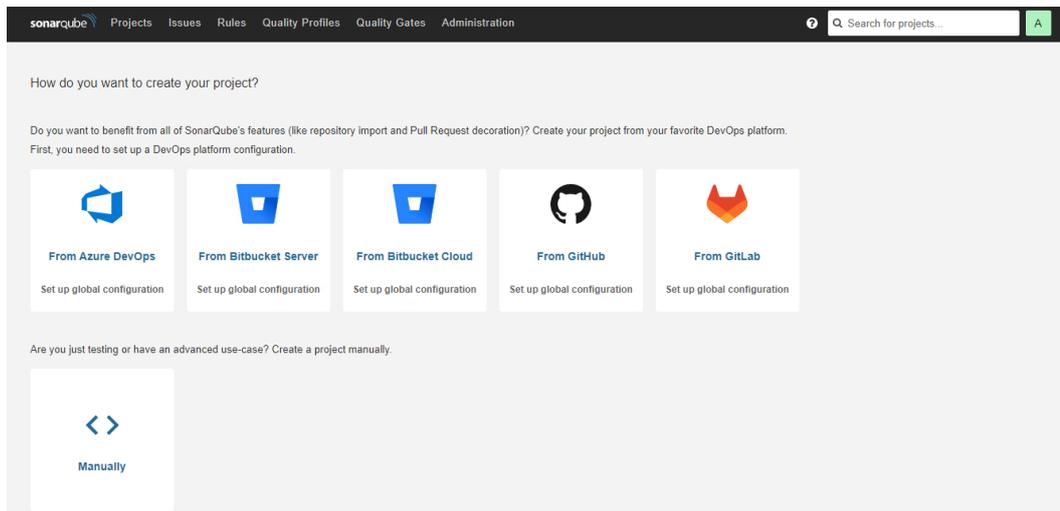
```
93 #sonar.web.javaOpts=-Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError
94
95 # Same as previous property, but allows to not repeat all other settings like -Xmx
96 #sonar.web.javaAdditionalOpts=-Xmx
97
98 # Binding IP address. For servers with more than one IP address, this property specifies which
99 # address will be used for listening on the specified ports.
100 # By default, ports will be used on all IP addresses associated with the server.
101 sonar.web.host=10.0.190.110
102
103 # Web context. When set, it must start with forward slash (for example /sonarqube).
104 # Changing this value and restarting the server can have unexpected consequences for logged in users,
105 # as they may end up with multiple conflicting browser cookies. If your users experience odd
106 # behaviors after you changed this value and restarted the server, ask them to clear their browser's
107 # cookies and log in again.
108 # The default value is root context (empty value).
109 #sonar.web.context=
110 # TCP port for incoming HTTP connections. Default value is 9000.
111 sonar.web.port=9000
112
113
114 # The maximum number of connections that the server will accept and process at any given time.
115 # When this number has been reached, the server will not accept any more connections until
116 # the number of connections falls below this value. The operating system may still accept connections
117 # based on the sonar.web.connections.acceptCount property. The default value is 50.
118 sonar.web.http.maxThreads=50
119
120 # The minimum number of threads always kept running. The default value is 5.
121 sonar.web.http.minThreads=5
122
123 # The maximum queue length for incoming connection requests when all possible request processing
124 # threads are in use. Any requests received when the queue is full will be refused.
```

Для работы Sonar Qube установили Java Developer Kit



После чего мы запустили StartSonar.bat и сервер был SonarQube был запущен, после чего перешли на по адресу localhost:9000 и зашли на сайт

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		12



Во время установки SonarQube было множество проблем с Java настройкой для программы, множество версий не работали с SonarQube.

Вывод: научились устанавливать и настраивать SonarQube для работы с тестированием.

		Аликина			АТПР.090207.20.201	Лист
						13
Изм	Лист	№ документа	Подпись	Дата		

Отчёт по практической работе №5

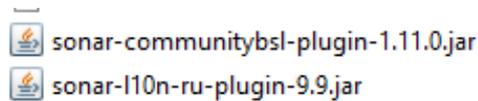
Управление качеством кода 1С с помощью SonarQube

Ход работы:

Скачали и установили sonar scanner 1С

.scannerwork	05.05.2023 14:40	Папка с файлами	
src	05.05.2023 14:37	Папка с файлами	
check.bat	05.05.2023 14:10	Пакетный файл ...	1 КБ
sonar-project.properties	05.05.2023 14:40	Файл "PROPERTIES"	1 КБ

Скачали 2 плагина для работы с русским языком и сканировании файлов конфигурации 1С:

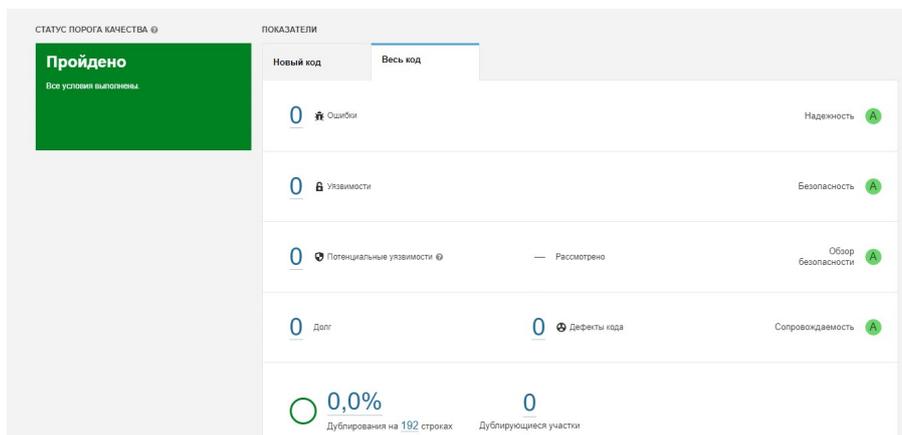


Просканировали нашу конфигурацию разработанную на МДК.04.01

```
INFO: Working dir: C:\Users\student\Downloads\workspace\1c\scannerwork
INFO: Load project settings for component key: 'Project'
INFO: Load project settings for component key: 'Project' (done) | time=32ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=79ms
INFO: Load active rules
INFO: Load active rules (done) | time=1743ms
INFO: Load analysis cache
INFO: Load analysis cache (404) | time=16ms
INFO: Load project repositories
INFO: Load project repositories (done) | time=31ms
INFO: Indexing files...
INFO: Project configuration:
INFO:   Included sources: **/*.bsl, **/*.os, **/*.dt
INFO: 10 files indexed
INFO: 66 files ignored because of inclusion/exclusion patterns
INFO: Quality profile for bsl: BSL Language Server rules
INFO: ----- Run sensors on module Project
INFO: Load metrics repository
INFO: Load metrics repository (done) | time=62ms
```

По итогу тестирования было обнаружено 0 ошибок и проблем в 192 строчек кода:

		Аликина			АТПР.090207.20.201	Лист
						14
Изм	Лист	№ документа	Подпись	Дата		



Вывод: научились сканировать код конфигурации 1С с помощью Sonar Scanner, устанавливая плагины и работать с проектами в SonarQube.

		Аликина			АТПР.090207.20.201	Лист
						15
Изм	Лист	№ документа	Подпись	Дата		

Отчет по практической работе №6

Автоматизированная проверка конфигураций 1С

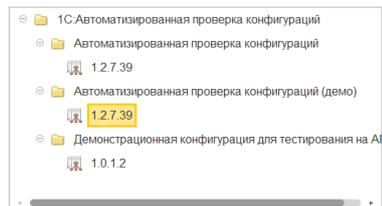
Цель: выполнить установку конфигурации инструмента Автоматизированная проверка конфигураций.

Скачали конфигурацию «Автоматизированная проверка конфигураций».

Добавление информационной базы/группы ×

Создание информационной базы из шаблона

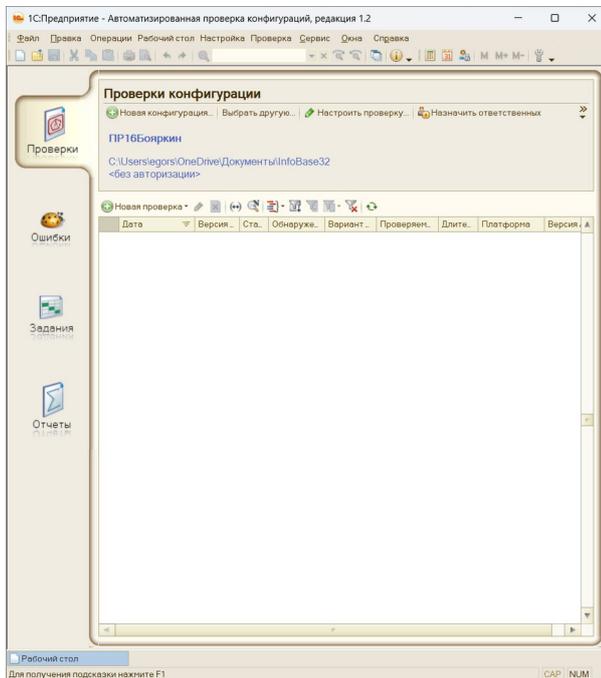
Выберите поставляемую конфигурацию для начала работы или демонстрационный пример для ознакомления.



Создание информационной базы без конфигурации для разработки новой конфигурации или загрузки выгруженной ранее информационной базы

< Назад Далее > Отмена

Открыли 1С АПК.



Нажали на кнопку добавления новой конфигурации и заполнили все необходимые реквизиты для проверки.

		Аликина			АТПР.090207.20.201	Лист
						16
Изм	Лист	№ документа	Подпись	Дата		

Проверка конфигурации: Создание *

Наименование: ПР16Бояркин Версия: от: ..

Полное наименование: ПР16Бояркин

Ответственны: Администратор

Включать в отчетность

Выполнять рассылку результатов проверки

Выполнять рассылку уведомлений о перенаправленных ошибках

Подключение

Платформа

1С Версия: 8.3.16

EDT Запуск: C:\Program Files (x86)\1cv8\8.3.16.1148\bin\1cv8.exe

Параметры подключения

В качестве источника для проверки можно указать демобазу, хранилище конфигурации, каталог выгрузки конфигурации, каталог проекта EDT, cf-файл или dt-файл.

Путь к источнику проверк.: C:\Users\legors\OneDrive\Документы\InfoBase32

Пользовател.: Парол.:

Выбран полный путь к каталогу файловой информационной базы

Загружать конфигурацию из хранилищ.

Хранилище:

Проверить подключение **OK** Записать Закрыть

Проверили успешное ли подключение в АПК.

1С:Предприятие

Подключение выполнено успешно.

OK

Выполняем автоматическую проверку конфигурации, разработанную в МДК.04.01

Запуск проверки

ЗАПУСК ПРОВЕРКИ

Наименование конфигурации:
ПР16Бояркин

Автоматически Демобазы: C:\Users\legors\OneDrive\Документы\InfoBase32
 При проверке будут использованы настройки по умолчанию. Пользователь: <не задан>. Пароль: <не задан>
 Проверяемые требования (146)
 Сценарий: Конфигурация (без расширений)

Вручную
 Вручную выбрать необходимые параметры проверки.

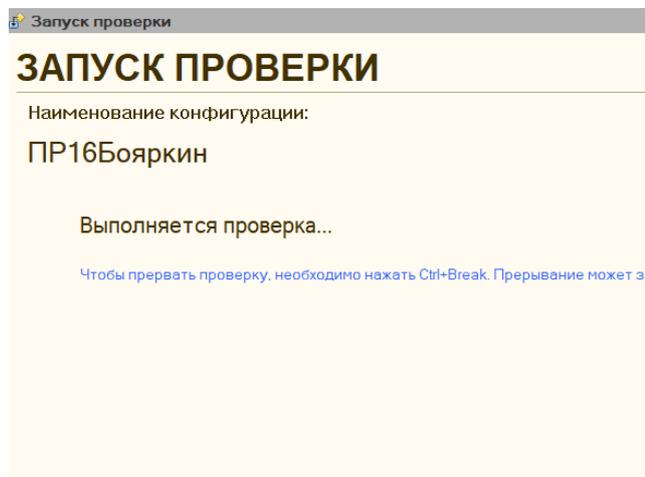
Регистрировать все ошибки как особенности

Проверить объекты: Только измененные Все

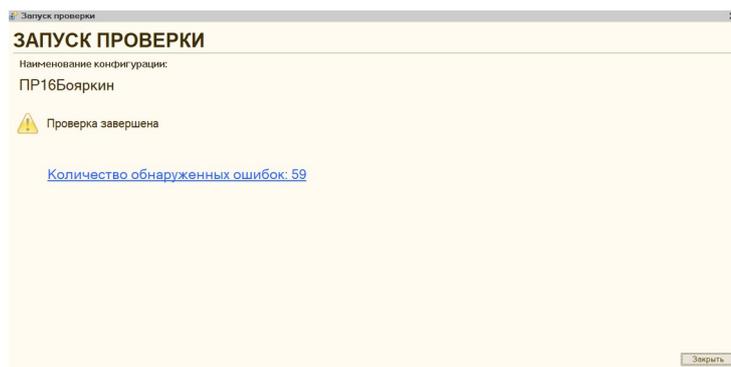
Выполнить проверк. **Отмена**

		Аликина			АТПР.090207.20.201	Лист
						17
Изм	Лист	№ документа	Подпись	Дата		

После чего происходит проверка конфигурации на выявлении ошибок.



После проверки АПК выдал количество ошибок в количестве 59.



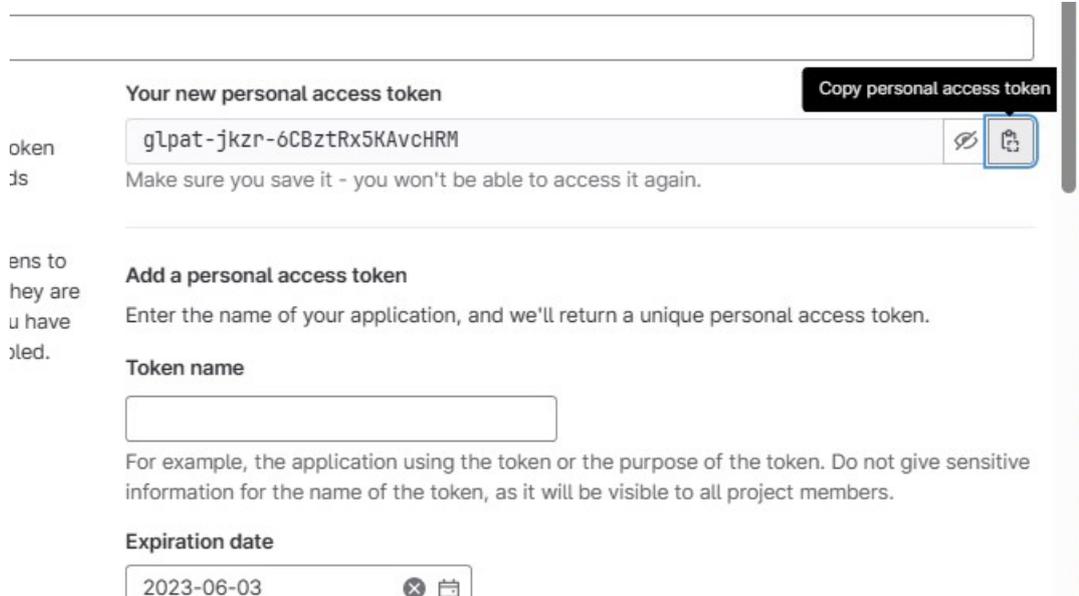
Вывод: Проверили на работоспособность приложение АСС для 1С, научились проверять нашу конфигурацию на наличие ошибок или проблем.

		Аликина			АТПР.090207.20.201	Лист
						18
Изм	Лист	№ документа	Подпись	Дата		

Отчет по практической работе №7

Подключение сервиса SonarQube к сервису GIT

Сгенерируем токен:



Your new personal access token Copy personal access token

oken 🔗 📄

ds Make sure you save it - you won't be able to access it again.

ens to **Add a personal access token**
hey are Enter the name of your application, and we'll return a unique personal access token.
u have
led.

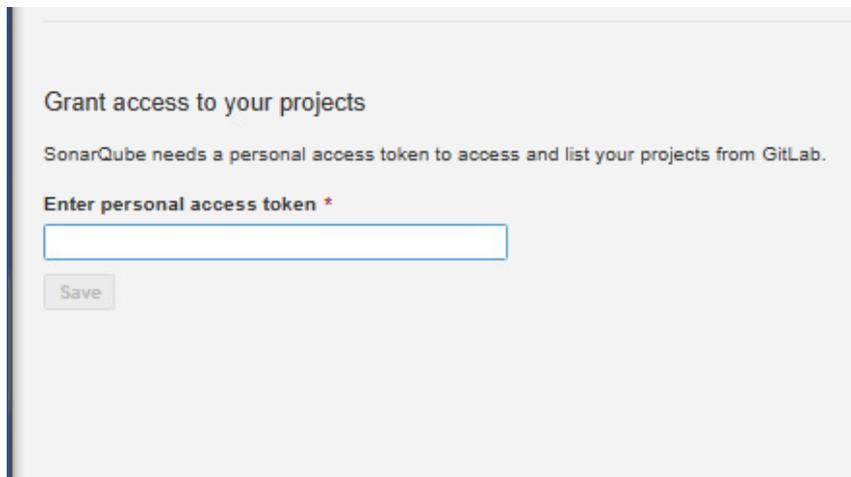
Token name

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

Expiration date

✕ 📅

Предоставим доступ для проекта:



Grant access to your projects

SonarQube needs a personal access token to access and list your projects from GitLab.

Enter personal access token *

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		19

Grant access to your projects

SonarQube needs a personal access token to access a

Enter personal access token *

Save

Создадим в репозитории файл `sonar-project.properties`. С указанием ключа и параметра, мониторящего связь с Sonarqube:

Analyze your project with GitLab CI

- 1 Set your project key

1. What option best describes your build?

Maven Gradle .NET Other (for JS, TS, Go, Python, PHP, ...)
- 2 Add environment variables
- 3 Create or update the configuration file
- 4 You're all set!

		Аликина			АТПР.090207.20.201	Лист
						20
Изм	Лист	№ документа	Подпись	Дата		

Analyze your project with GitLab CI

1 Set your project key

1. What option best describes your build?

2. Create a `sonar-project.properties` file in your repository and paste the following code:

```
sonar.projectKey=sonar5963918_Sonar_AYfm18yAkAhHr2VKIK2Z
sonar.qualitygate.wait=true
```

2 Add environment variables

3 Create or update the configuration file

4 You're all set!

Добавим среды переменных:

2 Add environment variables

1. Define the SonarQube Token environment variable.

In GitLab, go to **Settings > CI/CD > Variables** to add the following variable and make sure it is available for your project:

a. In the **Key** field, enter `SONAR_TOKEN`

b. In the **Value** field, enter an existing token, or a newly generated one:

c. Uncheck the **Protect Variable** checkbox.

d. Check the **Mask Variable** checkbox.

2. Define the SonarQube URL environment variable.

Still in **Settings > CI/CD > Variables** add a new variable and make sure it is available for your project:

a. In the **Key** field, enter `SONAR_HOST_URL`

b. In the **Value** field, enter `http://127.0.0.1:9000`

c. Uncheck the **Protect Variable** checkbox.

d. Leave the **Mask Variable** checkbox unchecked.

3 Create or update the configuration file

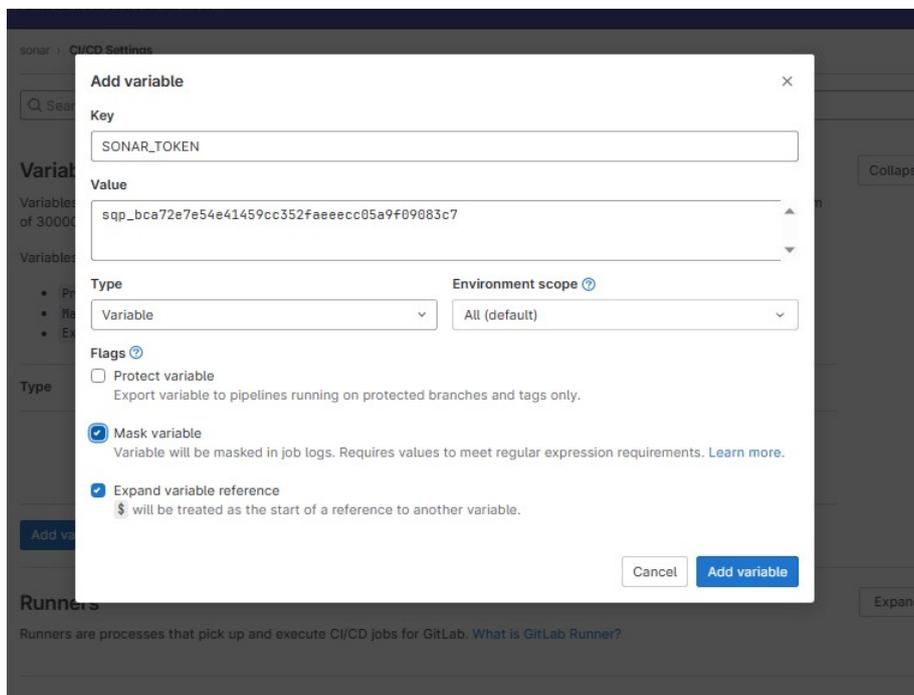
Добавим переменные:

		Аликина			АТПР.090207.20.201	Лист
						21
Изм	Лист	№ документа	Подпись	Дата		

- Protected: Only exposed to protected branches or protected tags.
- Masked: Hidden in job logs. Must match masking requirements.
- Expanded: Variables with \$ will be treated as the start of a reference to another variable.

Type	↑ Key	Value	Options	Environments
Variable	SONAR_HOST_URL	*****	Expanded	All (default)
Variable	SONAR_TOKEN	*****	Masked, Expanded	All (default)

[Add variable](#) [Reveal values](#)



Создадим конфигурацию файла:

3 Create or update the configuration file

Create or update your `.gitlab-ci.yml` file with the following content.

```
sonarqube-check:
  image:
    name: sonarsource/sonar-scanner-cli:latest
    entrypoint: [""]
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar" # Defines the location of the
    GIT_DEPTH: "0" # Tells git to fetch all the branches of the project, requ
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - sonar-scanner
  allow_failure: true
  only:
    - main
```

Note that this is a minimal base configuration to run a SonarQube analysis on your main branch. If you already have a pipeline configured and running, you might want to add the example from this step to your existing yml fi

[Finish this tutorial >](#)

Все готово:

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		22

3 Create or update the configuration file

4 You're all set!

You're all set and ready to improve the quality and security of your code!

 **Commit and push your code to start the analysis.**
Each new push you make on your main branch will trigger a new analysis in SonarQube.

 **This page will then refresh with your analysis results.**
If the page doesn't refresh after a while, please double-check the analysis configuration, and check your logs.

 Waiting for the first analysis to come in...

4 You're all set!

You're all set and ready to improve the quality and security of your code!

 **Commit and push your code to start the analysis.**
Each new push you make on your main branch will trigger a new analysis in SonarQube.

 **This page will then refresh with your analysis results.**
If the page doesn't refresh after a while, please double-check the analysis configuration, and check your logs.

 Waiting for the first analysis to come in...

Вывод: в данной практической работе SonarQube был подключен к GIT.

		Аликина			АТПР.090207.20.201	Лист
						23
Изм	Лист	№ документа	Подпись	Дата		

Отчёт по практической работе №8

Модульное тестирование в Visual Studio

Ход работы:

Создали проект и реализовали класс для нахождения площади прямоугольника

```
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace MathTaskClassLibrary
8 {
9     public class Geometry
10    {
11        public int RectanleArea(int a, int b)
12        {
13            return a * b;
14        }
15    }
16 }
17
```

В этом же решение создали проект модульного тестирования и реализовали unit тест с исходными данными 3 и 5, предполагаемым результатом 15

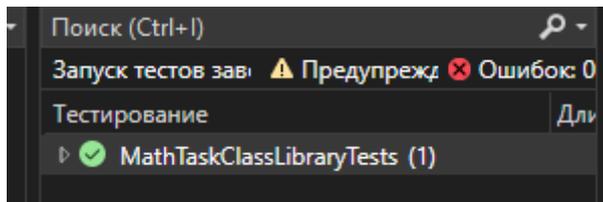
```
askClassLibrary tests
Math taskClassLibra
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;
using MathTaskClassLibrary;
namespace MathTaskClassLibraryTests
{
    [TestClass]
    public class GeometryTests
    {
        [TestMethod]
        public void RectangleArea_3and5_15returned()
        {
            int a = 3;
            int b = 5;
            int expected = 15;

            Geometry g = new Geometry();
            int actual = g.RectanleArea(a, b);

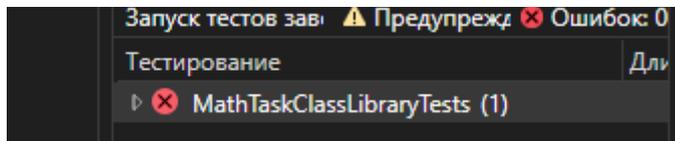
            Assert.AreEqual(expected, actual);
        }
    }
}
```

Провели тестирование и получили положительный результат

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		24



Но при изменении данных исходных, мы получаем ошибку.



Вывод: научились производить модульное тестирование в Visual Studio.

Отчёт по практической работе №9

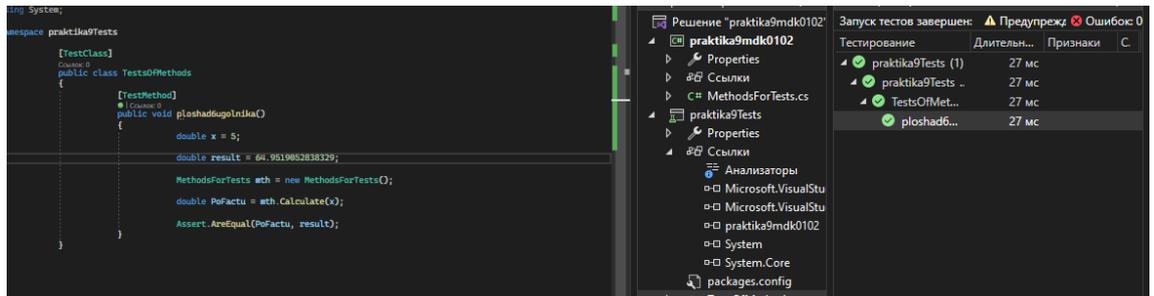
Модульное тестирование

Ход работы:

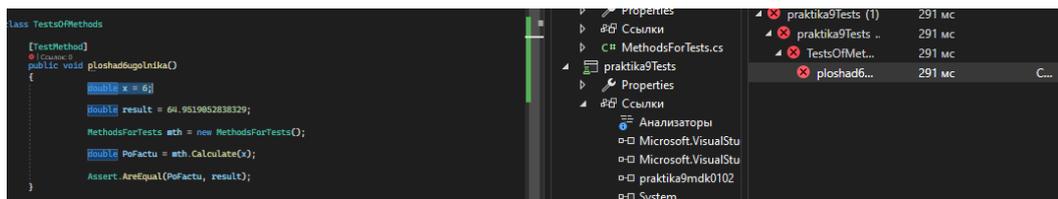
Из прошлых курсов МДК берем пример решения задачи на нахождение площади шестиугольника.

```
Ссылка 0
public class MethodsForTests
{
    Ссылка 1
    public double ploshad(double x)
    {
        //Находим площадь одного треугольника в правильном шестиугольнике
        double s = (Math.Pow(x, 2) * Math.Sqrt(3)) / 4;
        return s;
    }
    Ссылка 0
    public double Calculate(double x)
    {
        double S = 6 * ploshad(x);
        return S;
    }
}
```

В этом же решение создали проект модульного тестирования и протестировали его с положительными входными и выходными данными, получили положительный результат



При изменений данных на неправильные, возникает ошибка тестирования



		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		26

Из-за разницы в ожидаемых и фактических результатах выполнения метода:

тс: <93,5307436087194>. Фактически: <64,9519052838329>.

Вывод: научились самостоятельно тестировать модули программного продукта в Visual Studio, а также проверять работоспособность теста с помощью отладки теста.

		Аликина			АТПР.090207.20.201	Лист
						27
Изм	Лист	№ документа	Подпись	Дата		

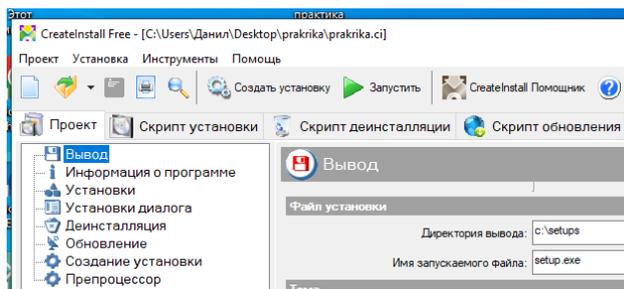
Отчёт по практической работе №10

Тестирование на этапе сопровождения программного продукта

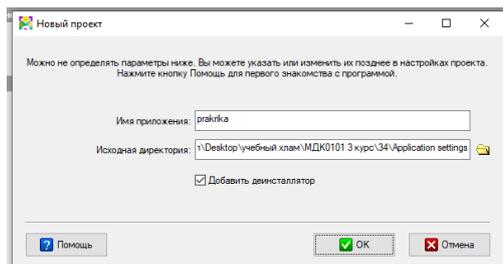
Цель работы: Научиться создавать инсталляционные файлы; выполнять оценочное тестирование программного продукта.

Ход работы:

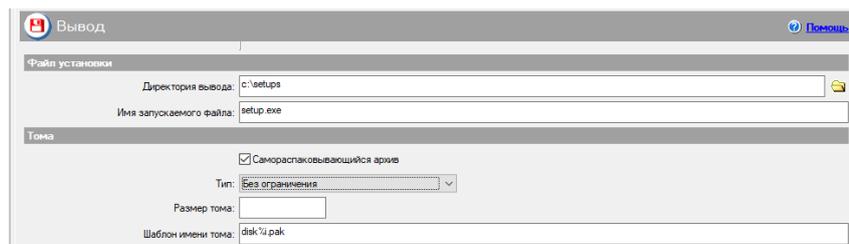
Скачали программу для создания инсталляторов CreateInstall:



Создали новый проект со своей практикой:



Выбрали название установочного файла:

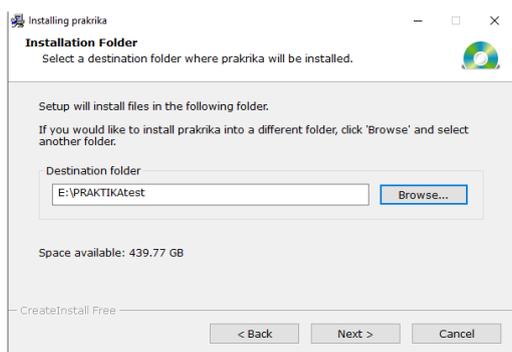


Программа создала инсталлятор, запустили его и начали устанавливать программу:

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		28



Выбрали путь установки:



После выбора пути установки начинается установка программы.

Далее мы попробовали удалить программу, удалилась без проблем.



Тестирование удобства установки прошло успешно, программа установилась быстро и хорошо, выбора при установке мало.

Установщик является рассчитанным на операционные системы Windows, что ограничивает пользование пользователями других ОС, однако, для установки не требуются дополнительные драйвера.

Восстановление в данной программе, не предусмотрено, что обуславливается простым функционалом.

		Аликина			АТПР.090207.20.201	Лист
						29
Изм	Лист	№ документа	Подпись	Дата		

Программа не очень удобна для эксплуатации, она не имеет масштабирования и развертывания. Кнопки подписаны и понятны в использовании, как и поля ввода. Средство отображение не удобное.

Вывод: научились создавать инсталляторы приложение с помощью программ. Также протестировали собственные программы на удобство использования.

Контрольные вопросы:

Что является целью тестирования программ?

Цель тестирования – выявление багов и ошибок, допущенных в ходе разработки программы. Тестирование удобства для будущего пользователя

Какие подходы к тестированию вы знаете? В чем они заключаются?

Тестирование бывает блочным, когда модули тестируются отдельно друг от друга. Следующий тип Интеграционное, когда тестируются группы связанных модулей вместе. И последний основной метод тестирования – системный, тестирование системы в целом.

С другой стороны, тестирование можно разделить по знанию кода, от белого ящика, когда весь код известен, до черного ящика, где код полностью не известен.

Обоснуйте необходимость создания инсталляторов программ.

Во-первых, инсталлятор программы, позволит пользователю самому выбрать что и куда устанавливать, также предупредит о лицензионном соглашении. После чего, автоматически установит все необходимые файлы и настройки системы для данной программы.

Вывод: В ходе данной практической работы был создан установочный файл разработанного нами программного продукта. С ним мы провели тестирование удобства использования установки и программы в целом.

		Аликина			АТПР.090207.20.201	Лист
						30
Изм	Лист	№ документа	Подпись	Дата		

Отчёт по практической работе №11

Техническое задание

Ход работы:

Техническое задание

1 Введение

1.1 Наименование программы

Наименование программы – «Военный округ».

1.2 Краткая характеристика области применения

Система «Военный округ» предназначена для организации структурирования данных о военном округе, состоящих из нескольких частей, рот и другого.

2 Основания для разработки

Основанием для разработки является Договор 1 от 01.01.2023. Договор утвержден преподавателем Авиационного техникума Басалаевым Александром Васильевичем, именуемым в дальнейшем Заказчиком, и Бояркин Данил Дмитриевич (самозанятый), именуемым в дальнейшем исполнителем, 01.01.2023.

Согласно Договору, Исполнитель обязан разработать и установить систему «Военный округ» на оборудовании Заказчика не позднее 31.12.2023, предоставить исходные коды и документацию к разработанной системе не позднее 30.12.2023.

Наименование темы разработки – «Защита и администрирование баз данных Военный округ»

Условное обозначение темы разработки (шифр темы) – «ВО».

3 Назначение разработки

Программа будет использоваться в базе военного округа двумя группами пользователей: администратор и секретарь.

		Аликина			АТПР.090207.20.201	Лист
						31
Изм	Лист	№ документа	Подпись	Дата		

3.1 Функциональное назначение

Для посетителя секретаря программа предоставляет возможность просмотра текущей информации о военном округе. Для администратора программа позволяет добавлять новые данные и изменять не актуальные.

3.2 Эксплуатационное назначение

Программа должна эксплуатироваться на компьютерах администраторов в офисе и секретарей в каждой части.

4 Требования к программе или программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

После запуска программы пользователю отображается форма ввода логина и пароля.

В системе существует всего 2 пользователя — администратор и секретарь. Программа проверяет тип пользователя и открывает соответствующий интерфейс.

Для администратора программа предоставляет все функции, предоставляемые секретарю а также дополнительные.

Окно расписания проката для оператора, помимо таблицы, должно содержать кнопки «Добавить» и «Удалить».

4.1.2 Требования к организации входных и выходных данных

Данные о прокатах фильмов и пользователях хранятся в базе данных. СУБД обеспечивает разграничение прав доступа к данным — дает секретарю права на чтение, а оператору — на чтение и запись. Ввод данных в базу (логины и пароли) выполняет администратор, выполняющий поставку программного обеспечения заказчику. Выполняет он это с помощью запросов к СУБД.

После установки программы, ввод данных в систему осуществляет только администратор, валидация данных выполняется на стороне клиента:

		Аликина			АТПР.090207.20.201	Лист
						32
Изм	Лист	№ документа	Подпись	Дата		

дата и время должны быть записаны в формате: «ДД.ММ.ГГГГ ЧЧ:ММ»;

название — последовательность не более чем из 200 любых символов;
возрастные ограничения — “+”.

4.1.3 Требования к временным характеристикам

После изменения данных, находящихся в базе данных, новая информация отображается не позднее, чем через 5 секунд.

4.2 Требования к надежности

Вероятность безотказной работы системы должна составлять не менее 99.99% при условии исправности сети.

4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

организацией бесперебойного питания технических средств;

использованием лицензионного программного обеспечения;

регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не

		Аликина			АТПР.090207.20.201	Лист
						33
Изм	Лист	№ документа	Подпись	Дата		

крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий секретаря при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

4.3 Условия эксплуатации

Программа (клиент) запускается на компьютере секретаря. База данных находится на третьем компьютере. Должна существовать устойчивая связь по сети между клиентами и базой данных.

Окно программы должно быть открыто на весь экран, не должно быть возможности закрыть, свернуть приложение или запустить любое стороннее программное обеспечение. Запуск программы должен осуществляться сразу после старта операционной системы.

4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

4.3.3 Требования к численности и квалификации персонала

При установке и настройке системы необходим системный администратор. В процессе эксплуатации с программой работают секретарь и администратор.

		Аликина			АТПР.090207.20.201	Лист
						34
Изм	Лист	№ документа	Подпись	Дата		

Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:

- установка клиентских приложений;
- настройка СУБД;
- настройка сети между клиентами и СУБД.

Пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

Администраторы должны быть аттестованы на II квалификационную группу по электробезопасности (для работы с конторским оборудованием).

К квалификации секретаря специальные требования не предъявляются.

4.4 Требования к составу и параметрам технических средств

Состав технических средств:

Компьютер оператора, включающий в себя:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь, клавиатура.

Компьютер посетителя, включающий в себя:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь.

Два компьютера для СУБД (основной и резервный), включающий в себя:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь.

4.5 Требования к информационной и программной совместимости

		Аликина			АТПР.090207.20.201	Лист
						35
Изм	Лист	№ документа	Подпись	Дата		

Приложения обмениваются с СУБД сообщениями по локальной сети, при этом используется протокол НТТР. Должно быть исключено появление посторонних устройств в сети.

4.6 Требование к маркировке и упаковке

Программное изделие передается по сети Internet в виде архива — загружается с официального сайта производителя. Специальных требований к маркировке не предъявляется. Для проверки подлинности программного обеспечения рекомендуется проверять контрольные суммы загруженных файлов со значениями, указанными на официальном сайте.

4.7 Требования к транспортированию и хранению

Специальных требований не предъявляется.

4.8 Специальные требования

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса, разработанного согласно рекомендациям компании-производителя операционной системы.

5 Требования к программной документации

Предварительный состав программной документации:
техническое задание (включает описание применения);
руководство программиста.

6 Техничко-экономические показатели

Программа «Военный округ» пригодна для небольших кинотеатров, не рассматривающих возможность продажи билетов через Internet. Скорее всего программа будет использоваться в промышленных военных округах. Функциональность программы совпадает с аналогами. В связи с тем, что из года в год военных округов не становится значительно больше, а количество маленьких военных округов даже снижается — не стоит ожидать роста годовой потребности. Однако, в случае бесплатного распространения программы, потребность в ней может быть весьма высокой

		Аликина			АТПР.090207.20.201	Лист
						36
Изм	Лист	№ документа	Подпись	Дата		

— в каждом крае есть военный округ. Экономический эффект при этом может быть обеспечен за счет платной установки системы.

7 Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

- техническое задание;
- технический (и рабочий) проекты;
- внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;
- разработка программной документации;
- испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Содержание работ по этапам:

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;
- определение и уточнение требований к техническим средствам;
- определение требований к программе;
- определение стадий, этапов и сроков разработки программы и документации на нее;
- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

		Аликина			АТПР.090207.20.201	Лист
						37
Изм	Лист	№ документа	Подпись	Дата		

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

разработка, согласование и утверждение порядка и методики испытаний;

проведение приемо-сдаточных испытаний;

корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

8 Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний. На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

		Аликина			АТПР.090207.20.201	Лист
						38
Изм	Лист	№ документа	Подпись	Дата		

Отчёт по практической работе №12

Составление руководства пользователя ИС

Цель работы: Ознакомиться с процедурой разработки руководства пользователя на создание программного продукта (ПП) с применением РД 50-34-698-90 «Автоматизированные системы требования к содержанию документов»

Ход работы:

Руководство пользователя

1. Введение

1.1 Область применения

Информационная система «Военный округ» применяется в государственных военных округах России. При этом объектами автоматизации при последующем внедрении будут государственные военные учреждения.

1.2 Краткое описание возможностей

Основные возможности:

- хранение данных о составе, части, роте и местоположений военных округов в базе данных;
- автоматизация добавление, обновления и удаления данных;
- удобный просмотр информации из любого военного округа.

1.3 Уровень подготовки пользователя

Все системные администраторы должны обладать навыками работы с СУБД Microsoft SQL, а также знать азы языка C#.

Все секретари и пользователи, непосредственно работающие с информацией, должны уметь пользоваться операционными системами Microsoft Windows.

1.4 Перечень эксплуатационной документации

		Аликина			АТПР.090207.20.201	Лист
						39
Изм	Лист	№ документа	Подпись	Дата		

Эксплуатационная документация должна содержать следующее:

- Техническое задание;
- Общее описание системы;
- Руководство пользователя;
- Руководство администратора.

2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Виды деятельности, функции

Основным предметом автоматизации являются функции военных округов России, по части сбору, хранению и передаче информации.

2.2 Программные и аппаратные требования к системе

Работа пользователей в информационной системе возможна при выполнении требований к характеристикам персонального компьютера, а также при наличии указанных операционной системы и дополнительного ПО.

3 ПОДГОТОВКА К РАБОТЕ

3.1 Состав дистрибутива

В состав дистрибутива входит:

- СУБД Microsoft SQL;
- Приложение установки БД;
- Программная часть Windows приложения.

3.2 Запуск системы

Перед запуском системы, необходимо выполнить установку СУБД и настроить корректную её работу, после чего, необходимо настроить клиентский ПК для работы с программой.

3.3 Проверка работоспособности системы

Система будет являться работоспособной, если после полной установке и настройке системы, у пользователя при запуске не появляются ошибки.

4 ОПИСАНИЕ ОПЕРАЦИЙ

Операции осуществляемые в ходе работы с информационной системой:

		Аликина			АТПР.090207.20.201	Лист
						40
Изм	Лист	№ документа	Подпись	Дата		

Добавление данных, действий выполняется, когда пользователь нажимает «Добавить» в программном окне продукта.

После завершения операции, данные, введенные в поля, отправляются в базу данных, после чего, отображаются у всех пользователей

В качестве ресурса, расходуется место, на жестком диске базы данных.

Изменение данных, действий выполняется, когда пользователь нажимает «Изменить» в программном окне продукта.

После завершения операции, данные, введенные в поля, отправляются в базу данных и обновляются на новые, заменяя старые, после чего, отображаются у всех пользователей

Удаление данных, действие выполняется, когда пользователь нажимает «Удалить» в программном окне продукта.

После завершения операции, данные, которые пользователь намеревался удалить, стираются из базы данных и пропадают во всех приложениях.

Просмотр данных, действие выполняется, начиная с запуска программы пользователем, никак не влияет на других пользователей и базу данных.

5 АВАРИЙНЫЕ СИТУАЦИИ. ВОССТАНОВЛЕНИЕ БАЗЫ ДАННЫХ

При сбое в работе ПК, восстановление системы должно производиться после перезагрузки ОС.

При неверных действиях пользователя, программа выдаст ошибку с соответствующим сообщением и продолжает работу в нормальном состоянии

При ошибке с программным обеспечением, восстановление работоспособности возлагается на операционную систему и после на системного администратора.

6 РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

Для успешного освоения информационной системы, необходимо иметь навыки работы с компьютером и настоящее «Руководство пользователя».

		Аликина			АТПР.090207.20.201	Лист
						41
Изм	Лист	№ документа	Подпись	Дата		

Вывод: составили руководство пользователя, согласно ГОСТам.

Отчёт по практической работе №13

Итоговое задание

Ход работы:

Тестирование SonarScanner:

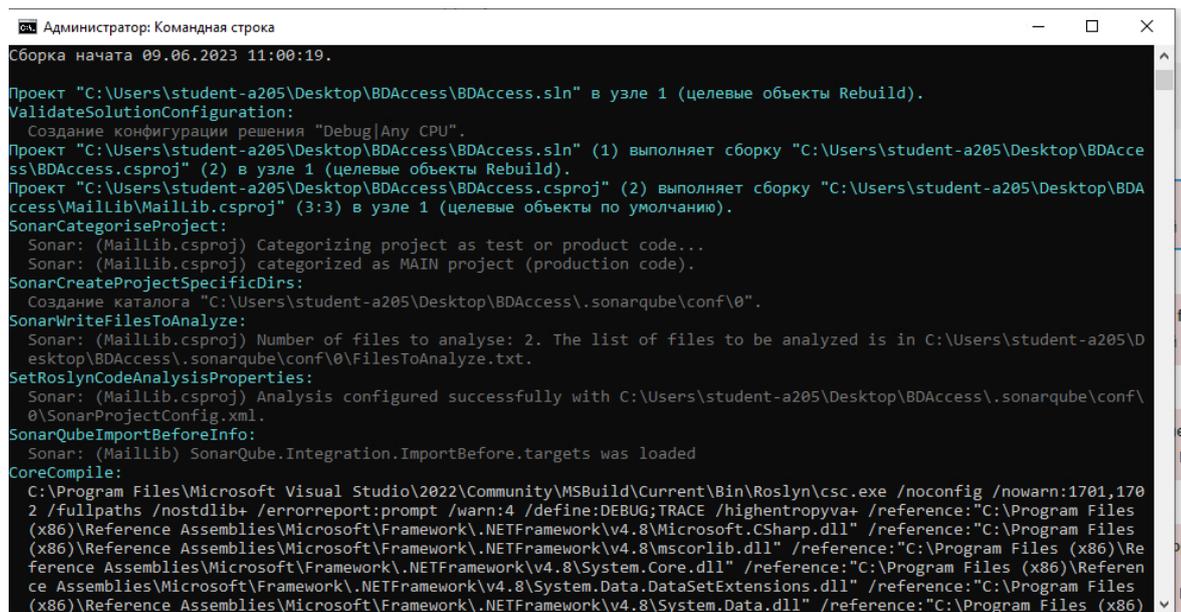
Первым делом мы занялись сканированием с помощью SonarQube

```
Microsoft Windows [Version 10.0.19045.2965]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Windows\system32>cd C:\Users\student-a205\Desktop\BDAccess

C:\Users\student-a205\Desktop\BDAccess>C:\Users\student-a205\Desktop\ssC\SonarS
SonarScanner for MSBuild 5.13
Using the .NET Framework version of the Scanner for MSBuild
Pre-processing started.
Preparing working directories...
10:46:35.238 Updating build integration targets...
10:46:35.367 Fetching analysis configuration settings...
10:46:36.085 Provisioning analyzer assemblies for cs...
10:46:36.1 Installing required Roslyn analyzers...
10:46:36.1 Processing plugin: csharp version 8.51.0.59060
10:46:36.414 Processing plugin: vbnet version 8.51.0.59060
10:46:36.721 Provisioning analyzer assemblies for vbnet...
10:46:36.721 Installing required Roslyn analyzers...
10:46:36.721 Processing plugin: csharp version 8.51.0.59060
10:46:36.721 Processing plugin: vbnet version 8.51.0.59060
10:46:36.737 Incremental PR analysis: Base branch parameter was not provided.
10:46:36.737 Cache data is empty. A full analysis will be performed.
10:46:36.768 Pre-processing succeeded.
```

Выполнили сборку проекта с помощью MSBuild



```
Администратор: Командная строка
Сборка начата 09.06.2023 11:00:19.

Проект "C:\Users\student-a205\Desktop\BDAccess\BDAccess.sln" в узле 1 (целевые объекты Rebuild).
ValidateSolutionConfiguration:
  Создание конфигурации решения "Debug|Any CPU".
Проект "C:\Users\student-a205\Desktop\BDAccess\BDAccess.sln" (1) выполняет сборку "C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj" (2) в узле 1 (целевые объекты Rebuild).
Проект "C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj" (2) выполняет сборку "C:\Users\student-a205\Desktop\BDAccess\Maillib\Maillib.csproj" (3:3) в узле 1 (целевые объекты по умолчанию).
SonarCategoriseProject:
  Sonar: (Maillib.csproj) Categorizing project as test or product code...
  Sonar: (Maillib.csproj) categorized as MAIN project (production code).
SonarCreateProjectSpecificDirs:
  Создание каталога "C:\Users\student-a205\Desktop\BDAccess\.sonarqube\conf\0".
SonarWriteFilesToAnalyze:
  Sonar: (Maillib.csproj) Number of files to analyze: 2. The list of files to be analyzed is in C:\Users\student-a205\Desktop\BDAccess\.sonarqube\conf\0\FilesToAnalyze.txt.
SetRoslynCodeAnalysisProperties:
  Sonar: (Maillib.csproj) Analysis configured successfully with C:\Users\student-a205\Desktop\BDAccess\.sonarqube\conf\0\SonarProjectConfig.xml.
SonarQubeImportBeforeInfo:
  Sonar: (Maillib) SonarQube.Integration.ImportBefore.targets was loaded
CoreCompile:
  C:\Program Files\Microsoft Visual Studio\2022\Community\MSBuild\Current\Bin\Roslyn\csc.exe /noconfig /nowarn:1701,1702 /fullpaths /nostdlib+ /errorreport:prompt /warn:4 /define:DEBUG;TRACE /highentropyva+ /reference:C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.8\Microsoft.CSharp.dll /reference:C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.8\mscorlib.dll /reference:C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.8\System.Core.dll /reference:C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.8\System.Data.DataSetExtensions.dll /reference:C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.8\System.Data.dll /reference:C:\Program Files (x86)...
```

		Аликина				Лист
						42
Изм	Лист	№ документа	Подпись	Дата		

АТПР.090207.20.201

Собрали проект в SonarScanner

```
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\Maillib\bin\Debug\Maillib.dll" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\Maillib.dll".
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\WpfControlLibrary1\bin\Debug\WpfControlLibrary1.dll" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\WpfControlLibrary1.dll".
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\Maillib\bin\Debug\Maillib.pdb" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\Maillib.pdb".
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\WpfControlLibrary1\bin\Debug\WpfControlLibrary1.pdb" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\WpfControlLibrary1.pdb".
Обращение к "C:\Users\student-a205\Desktop\BDAccess\obj\Debug\BDAccess.csproj.CopyComplete".
CopyAppConfigFile:
Целевой объект "_СоруAppConfigFile" пропускается, так как все выходные файлы актуальны по отношению к входным.
CopyFilesToOutputDirectory:
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\obj\Debug\BDAccess.exe" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\BDAccess.exe".
BDAccess -> C:\Users\student-a205\Desktop\BDAccess\bin\Debug\BDAccess.exe
Копирование файла из "C:\Users\student-a205\Desktop\BDAccess\obj\Debug\BDAccess.pdb" в "C:\Users\student-a205\Desktop\BDAccess\bin\Debug\BDAccess.pdb".
SonarWriteProjectData:
Sonar: (BDAccess.csproj) Project processed successfully
Сборка проекта "C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj" завершена (целевые объекты Rebuild).
Сборка проекта "C:\Users\student-a205\Desktop\BDAccess\BDAccess.sln" завершена (целевые объекты Rebuild).
Сборка успешно завершена.
```

В ходе сборки было обнаружено 64 предупреждения

```
Администратор: Командная строка
C:\Users\student-a205\Desktop\BDAccess\HashCode.cs(14,23): warning S4790: Make sure this weak hash algorithm is not used in a sensitive context here. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Pages\Login.xaml.cs(75,21): warning S2696: Make the enclosing instance method 'static' or remove this set on the 'static' field. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Pages\Login.xaml.cs(120,13): warning S2486: Handle the exception or explain in a comment why it can be ignored. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Pages\Login.xaml.cs(120,19): warning S108: Either remove or fill this block of code. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Classes\User.cs(20,20): warning S107: Constructor has 8 parameters, which is greater than the 7 authorized. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Classes\User.cs(33,23): warning S1481: Remove the unused local variable 'create_user'. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Classes\User.cs(35,42): warning S107: Method has 9 parameters, which is greater than the 7 authorized. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Classes\User.cs(37,23): warning S1481: Remove the unused local variable 'create_user'. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]
C:\Users\student-a205\Desktop\BDAccess\Pages\UserRegistration.xaml.cs(152,21): warning S6444: Pass a timeout to limit the execution time. [C:\Users\student-a205\Desktop\BDAccess\BDAccess.csproj]

Предупреждений: 64
Ошибок: 0

Прошло времени 00:00:14.18
C:\Users\student-a205\Desktop\BDAccess>C:\Users\student-a205\Desktop\ssC\SonarScanner.MSBuild.exe end /d:sonar.login="sq_a_78d260946374d3d18c71d16558d68f9679b1ef9f"
SonarScanner for MSBuild 5.13
Using the .NET Framework version of the Scanner for MSBuild
Post-processing started.
Calling the TFS Processor executable...
```

После нахождения ошибок и предупреждений, загрузили эти данные в проект в базу данных SonarQube

		Аликина			АТПР.090207.20.201	Лист
						43
Изм	Лист	№ документа	Подпись	Дата		

```

C:\Users\student-a205\Desktop\BDAccess>C:\Users\student-a205\Desktop\ssC\SonarScanner.MSBuild.exe end /d:sonar.login="sq
a_78d260946374d3d18c71d16558d68f9679b1ef9f"
SonarScanner for MSBuild 5.13
Using the .NET Framework version of the Scanner for MSBuild
Post-processing started.
Calling the TFS Processor executable...
The TFS Processor has finished
Calling the SonarScanner CLI...
INFO: Scanner configuration file: C:\Users\student-a205\Desktop\ssC\sonar-scanner-4.8.0.2856\bin\..\conf\sonar-scanner.p
roperties
INFO: Project root configuration file: C:\Users\student-a205\Desktop\BDAccess\sonarqube\out\sonar-project.properties
INFO: SonarScanner 4.8.0.2856
INFO: Java 17.0.7 Oracle Corporation (64-bit)
INFO: Windows 10 10.0 amd64
INFO: User cache: C:\Users\student-a205\sonar\cache
INFO: Analyzing on SonarQube server 9.9.0.65466
INFO: Default locale: "ru_RU", source code encoding: "UTF-8"
INFO: Load global settings
INFO: Load global settings (done) | time=150ms
INFO: Server id: 00A7BDFB-AYFHFik31S6GU8enbXog
INFO: User cache: C:\Users\student-a205\sonar\cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=162ms
INFO: Load/download plugins (done) | time=412ms
INFO: Process project properties
INFO: Process project properties (done) | time=16ms

```

Сканирование проекта заняло 12 секунд и нашло предупреждения.

```

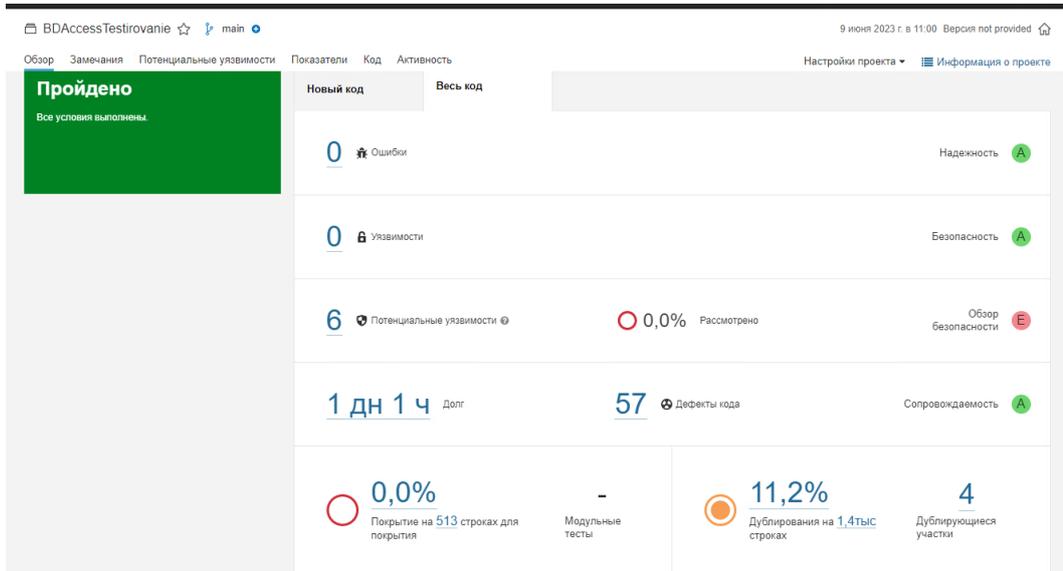
Администратор: Командная строка
INFO: Sensor C# [csharp] (done) | time=781ms
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=16ms
INFO: Sensor C# File Caching Sensor [csharp]
INFO: Sensor C# File Caching Sensor [csharp] (done) | time=16ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=31ms
INFO: SCM Publisher is disabled
INFO: CPD Executor 3 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 22 files
INFO: CPD Executor CPD calculation finished (done) | time=78ms
INFO: Analysis report generated in 203ms, dir size=273.1 kB
INFO: Analysis report compressed in 140ms, zip size=90.8 kB
INFO: Analysis report uploaded in 94ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=BDAccessTestirovanie
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis re
port
INFO: More about the report processing at http://127.0.0.1:9000/api/ce/task?id=AYev1qV34DT1gNyiP7j
INFO: Analysis total time: 10.387 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 12.411s
INFO: Final Memory: 16M/60M
INFO: -----
The SonarScanner CLI has finished
11:00:55.501 Post-processing succeeded.
C:\Users\student-a205\Desktop\BDAccess>

```

Проект появился в localhost:9000

На главной странице проекта увидели информацию о дублировании строк, дефектах кода и потенциальных уязвимостях

		Аликина			АТПР.090207.20.201	Лист
						44
Изм	Лист	№ документа	Подпись	Дата		



Посмотрели полный список проблем кода

Филтеры Очистить все филтеры

Новый код

ДЕФЕКТ КОДА Очистить

Ошибки 0

Уязвимость 0

Дефект кода 57

Сти - клик, чтобы добавить к выбору

Серьезность

- Блпору... 0
- Незнач... 30
- Критичес... 7
- Информ... 0
- Важное 20

Тип кода

Решение

Статус

Категория безопасности

Дата создания

Язык

Правило

Тэг

Classes\User.cs

- Constructor has 8 parameters, which is greater than the 7 authorized. Дефект кода Важное Открыто Не назначено 20min усилий Прокомментировать
- Remove the unused local variable 'create_user'. Дефект кода Незначительное Открыто Не назначено 5min усилий Прокомментировать
- Method has 9 parameters, which is greater than the 7 authorized. Дефект кода Важное Открыто Не назначено 20min усилий Прокомментировать
- Remove the unused local variable 'create_user'. Дефект кода Незначительное Открыто Не назначено 5min усилий Прокомментировать

HashCode.cs

- Add a 'protected' constructor or the 'static' keyword to the class declaration. Дефект кода Важное Открыто Не назначено 10min усилий Прокомментировать

MainWindow.xaml.cs

- Remove this commented out code. Дефект кода Важное Открыто Не назначено 5min усилий Прокомментировать
- Make this field 'private' and encapsulate it in a 'public' property. Дефект кода Незначительное Открыто Не назначено 10min усилий Прокомментировать
- Make this field 'private' and encapsulate it in a 'public' property. Дефект кода Незначительное Открыто Не назначено 10min усилий Прокомментировать

Посмотрели каждую из проблем и по необходимости исправили их.

Приоритет проверки: **ВЫСОКИЙ**

Authentication 2

Please review this hard-coded password.

MailLib/SmtplibMessage.cs

"pwd" detected here, make sure this is not a hard-coded credential.

MainWindow.xaml.cs

Приоритет проверки: **СРЕДНИЙ**

Denial of Service (DoS) 1

Weak Cryptography 1

Приоритет проверки: **НИЗКИЙ**

Others 2

6 из 6 показано

Hard-coded credentials are security-sensitive. csharp squid:S2068

Статус: **К ПРОСМОТРУ**

Необходимо проанализировать эту потенциальную уязвимость, чтобы оценить, есть ли уязвимость в коде.

Изменить статус

Ответственный: Не назначено

Где риск? В чём риск? Вы рискуете? Как это исправить?

MailLib/SmtplibMessage.cs Открыть в IDE Получить постоянную ссылку

```

11 public class SmtplibMessage
12 {
13     public string SenderName { get; set; } = "System Admin";
14     public string Subject { get; set; } = "Disposable code";
15     public string ToMail { get; set; }
16     public string Code { get; } = new Random().Next(1000, 9999).ToString();
17     public void SendCode()
18     {
19         SmtplibClient Smtplib = new SmtplibClient("smtp.yandex.ru", 587);
20         Smtplib.EnableSsl = true;
21         Smtplib.Credentials = new NetworkCredential("superadmin@yandex.ru", "helpmegetplease");
22
23         MailMessage Message = new MailMessage();
24         Message.From = new MailAddress("superadmin@yandex.ru", SenderName);

```

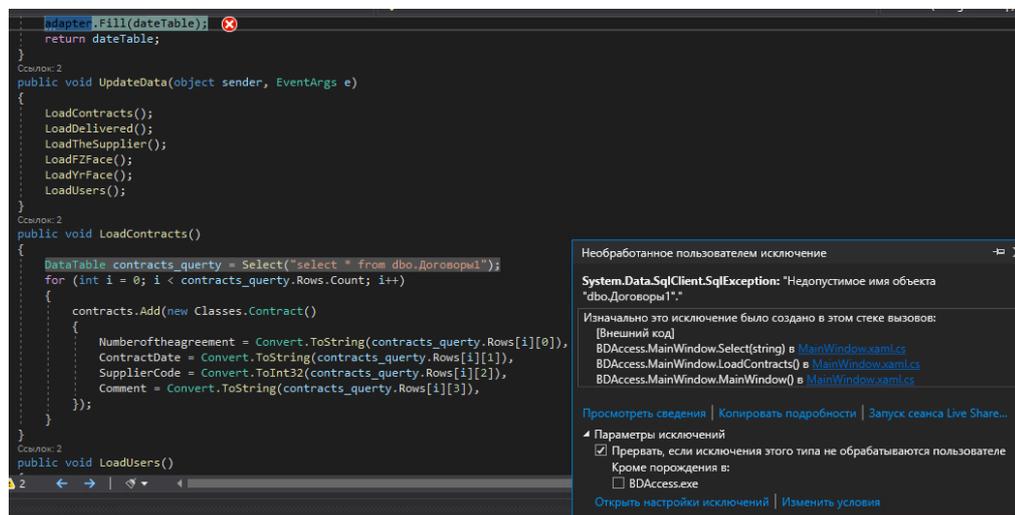
Please review this hard-coded password. Прокомментировать

		Аликина			АТПР.090207.20.201	Лист
Изм	Лист	№ документа	Подпись	Дата		45

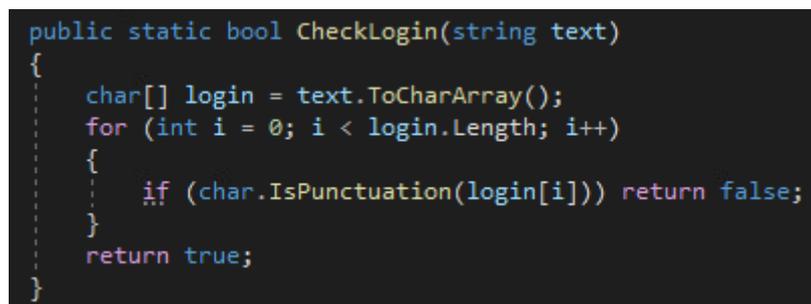
Ручной тестирование:

В ходе ручного тестирования мы искали ошибки в коде самостоятельно, было найдено 21 ошибка, разных типов, из них: валидация, запросы и другое.

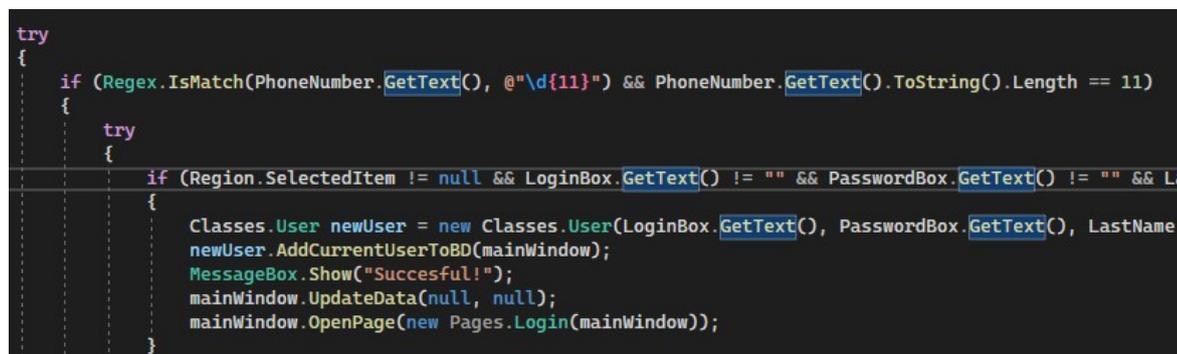
Одна из ошибок с запросами:



Одна из ошибок с валидацией:



Одна из ошибок, связанных с структурой кода:



		Аликина			АТПР.090207.20.201	Лист
						46
Изм	Лист	№ документа	Подпись	Дата		

Тестирование с помощью UNIT-тестов:

В ходе тестирования было создано 10 UNIT тестов для проверки разных модулей и функций.

К примеру, UNIT тест для проверки что в поле номера допустимо вводить только 11 символов:

```
[TestMethod]
| Ссылка: 0
public void PhoneNumberOver11()
{
    bool result = false;
    Assert.AreEqual(BDAccess.Pages.UserRegistration.CheckPhoneNumber("123456789011"), result);
}
```

Итоги тестирования:

В ходе работы, мы протестировали программу тремя способами: Ручное тестирование, Unit-тесты и тестирование с помощью SonarScanner.

Все проведенные тесты и найденные ошибки, указаны в отчете по тестированию.

В результате тестирования, мы получили готовый проект, отчет по тестированию программного продукта, готовое исправленное техническое задание.

Вывод: потренировались в тестировании программного продукта, самостоятельно его протестирую всеми изученными способами и исправили найденные ошибки.

		Аликина			АТПР.090207.20.201	Лист
						47
Изм	Лист	№ документа	Подпись	Дата		