

Министерство образования и науки Российской Федерации  
Федеральное государственное образовательное учреждение  
высшего образования «Российский экономический  
университет имени Г.В.Плеханова»  
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

# Презентация для квалификационного экзамена

Подготовил студент группы П50-1-18  
Дрюпин Андрей Александрович

В рамках ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» были освоены 6 профессиональных компетенций:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6. Разрабатывать модули программного обеспечения для мобильных платформ.



# ПК 1.1. алгоритмы разработки

Для создания схем алгоритмов разработки использовались Erwin и Draw.io. Для создания эскизов использовался графический редактор Paint.

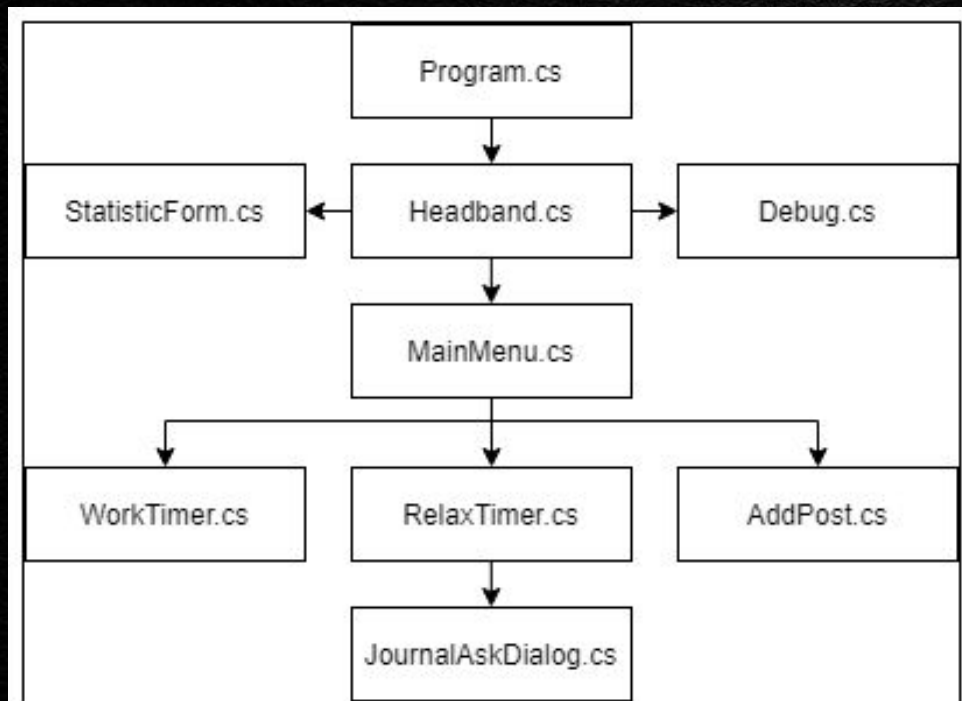


Рисунок 3 – Структурная схема приложения

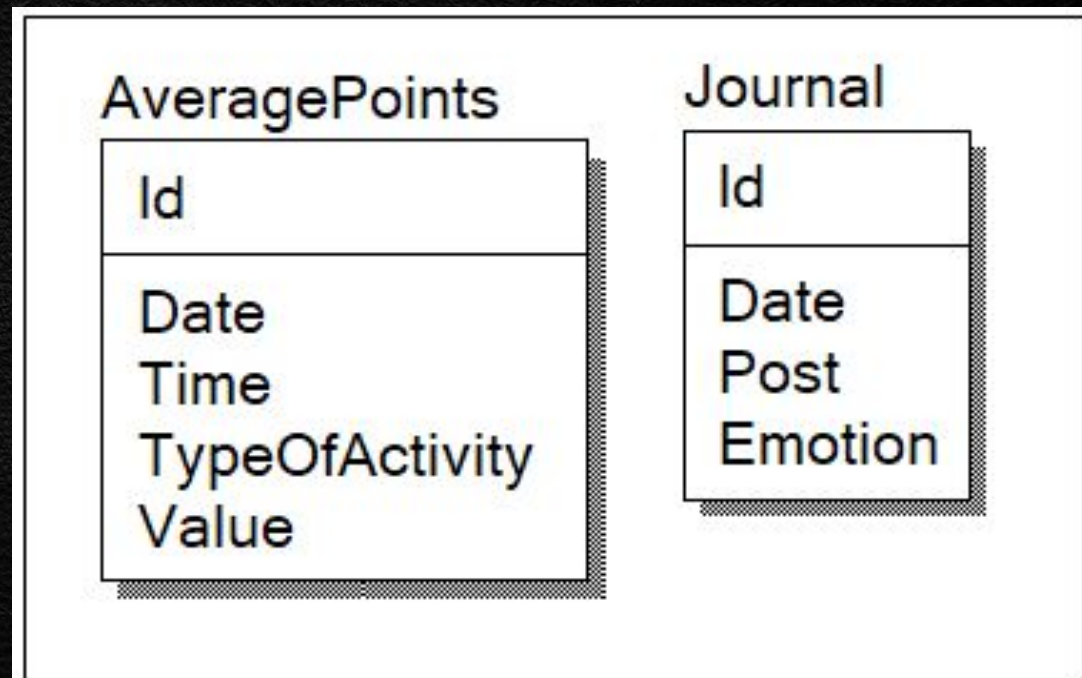


Рисунок 6 – Модель базы данных

# ПК 1.1. алгоритмы разработки

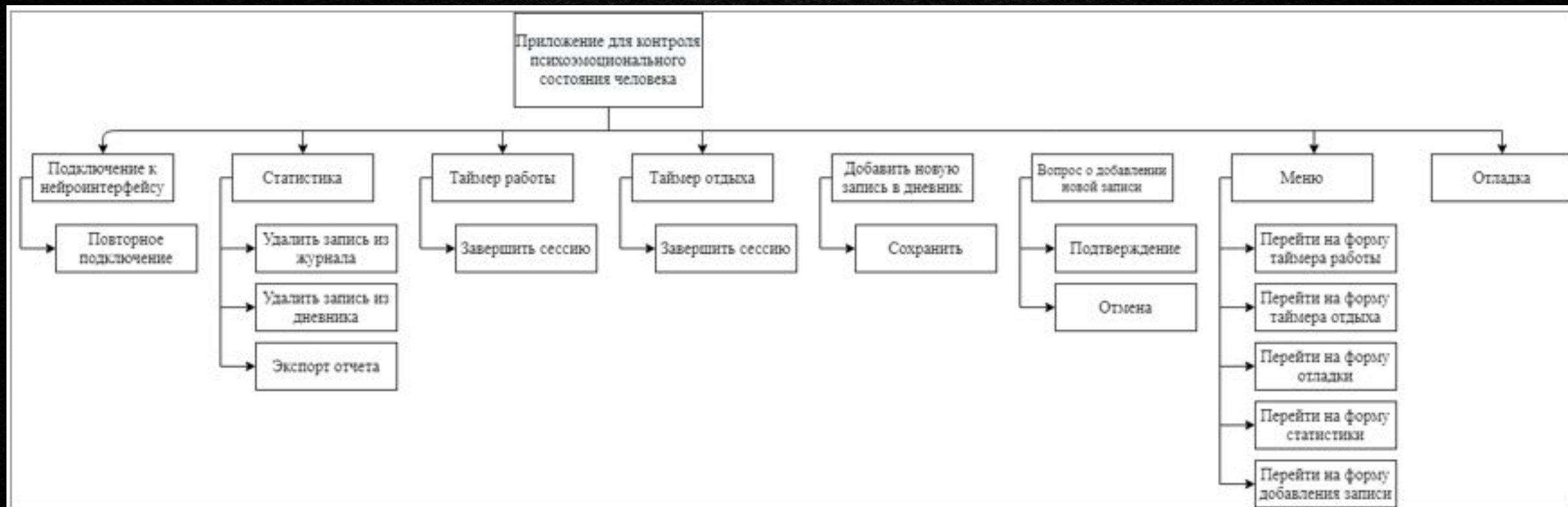


Рисунок 4 - Функциональная схема приложения



# ПК 1.1. алгоритмы разработки

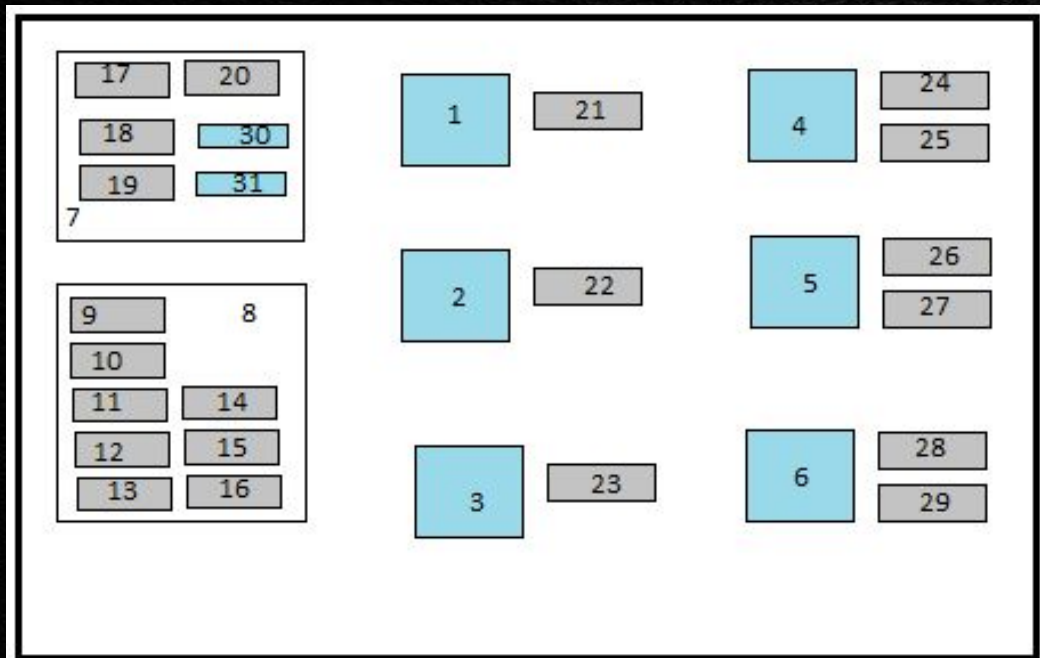


Рисунок 4 - Окно "Debug"

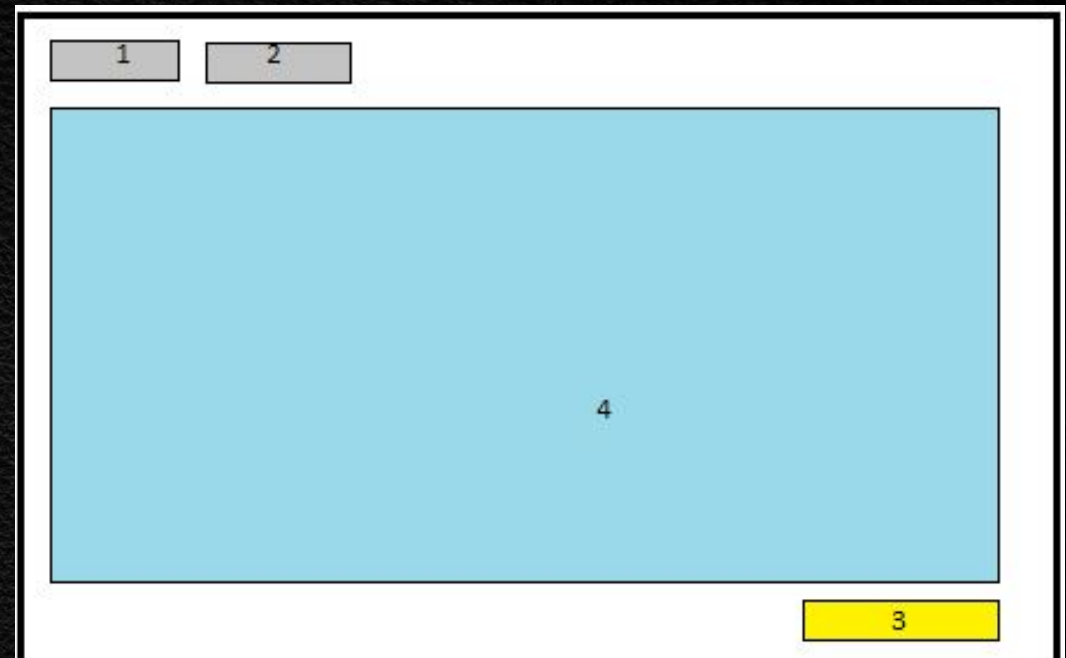


Рисунок 7 - Окно "Занятия"

# ПК 1.2.

The image displays two windows from a software application. The window titled 'Form1' contains a login form with two text input fields labeled 'Логин' and 'Пароль', and three buttons: 'Войти', 'Зарегистрироваться', and 'Восстановить пароль'. The 'Зарегистрироваться' button is highlighted with a blue border. Below it, an error dialog box is open, displaying the message 'Ошибка! Заполнены не все поля' and an 'OK' button. The window titled 'Reg' contains a registration form with seven text input fields labeled 'Логин', 'Пароль', 'Повторите пароль', 'ФИО', 'Серия и номер паспорта', 'Телефон (8)', and 'E-mail', and a 'Зарегистрироваться' button at the bottom, which is also highlighted with a blue border.

Разработка модуля авторизации и регистрации



# ПК 1.2.

```
namespace Auth
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void RegBtn(object sender, EventArgs e)
        {
            Reg frm = new Reg();
            frm.Show();
        }

        private void LogBtn(object sender, EventArgs e)
        {
            INIManager manager = new INIManager(Directory.GetCurrentDirectory() + "\\Users.ini");
            string Login = textBox1.Text;
            string Password = textBox2.Text;

            if (Login.Length <= 0 || Password.Length <= 0) MessageBox.Show("Ошибка! Заполнены не все поля");
            else if (Regex.IsMatch(Password, @"^(?=[a-z])(?=[A-Z])(?=[0-9])(?=[^a-zA-Z0-9])\S{1,16}$", RegexOptions.IgnoreCase))
            {
                if (manager.GetPrivateString(Login, "Password").Length != -1 && manager.GetPrivateString(Login, "Password") == Password)
                {
                    else MessageBox.Show("Ошибка! Неправильный логин или пароль");
                }
            }
            else MessageBox.Show("Ошибка! Пароль не соответствует условиям");
        }

        private void ChangeBtn_Click(object sender, EventArgs e)
        {
            Change cng = new Change();
            cng.Show();
        }
    }
}
```

Код модуля авторизации

# ПК 1.2.

```
public partial class Reg : Form
{
    public Reg()
    {
        InitializeComponent();
    }

    private void Register(object sender, EventArgs e)
    {
        int rightOK = 0;
        String Login = textBox1.Text;
        String Password = textBox2.Text;
        String RePassword = textBox3.Text;
        String Name = textBox4.Text;
        String Passport = textBox5.Text;
        String Email = textBox7.Text;
        String Phone = textBox6.Text;

        if (Login.Length <= 0 || Passport.Length <= 0 || Name.Length <= 0 || Password.Length <= 0 || RePassword.Length <= 0 || Email.Length <= 0)
        else
        {
            if (Password != RePassword) MessageBox.Show("Ошибка! Пароли не совпадают");
            else rightOK++;
            if (!Regex.IsMatch(Password, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[a-zA-Z0-9])\S{1,16}$", RegexOptions.IgnoreCase))
            else rightOK++;
            if (!Regex.IsMatch(Phone, @"^((\+7-?)|8)?[0-9]{10}$", RegexOptions.IgnoreCase)) MessageBox.Show("Ошибка! Телефон не соответствует условиям");
            else rightOK++;
            if (Passport.Length != 10) MessageBox.Show("Ошибка! Серия и номер паспорта не соответствует условиям");
            else rightOK++;
            if (!Regex.IsMatch(Email, @"^\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*$", RegexOptions.IgnoreCase)) MessageBox.Show("Ошибка! Email не соответствует условиям");
            else rightOK++;
        }
        if (rightOK == 5)
    }
}
```

Код модуля регистрации



# ПК 1.2.

```
public class INIManager
{
    //Конструктор, принимающий путь к INI-файлу
    public INIManager(string aPath)
    {
        path = aPath;
    }

    //Конструктор без аргументов (путь к INI-файлу нужно будет задать отдельно)
    public INIManager() : this("") { }

    //Возвращает значение из INI-файла (по указанным секции и ключу)
    public string GetPrivateString(string aSection, string aKey)
    {
        //Для получения значения
        StringBuilder buffer = new StringBuilder(SIZE);

        //Получить значение в buffer
        GetPrivateString(aSection, aKey, null, buffer, SIZE, path);

        //Вернуть полученное значение
        return buffer.ToString();
    }

    //Пишет значение в INI-файл (по указанным секции и ключу)
    public void WritePrivateString(string aSection, string aKey, string aValue)
    {
        //Записать значение в INI-файл
        WritePrivateString(aSection, aKey, aValue, path);
    }

    //Возвращает или устанавливает путь к INI файлу
    public string Path { get { return path; } set { path = value; } }
}
```

Код класса чтения/записи INI-файлов



# ПК 1.2.

```
//Возвращает или устанавливает путь к INI файлу
public string Path { get { return path; } set { path = value; } }

//Поля класса
private const int SIZE = 1024; //Максимальный размер (для чтения значения из файла)
private string path = null; //Для хранения пути к INI-файлу

//Импорт функции GetPrivateProfileString (для чтения значений) из библиотеки kernel32.dll
[DllImport("kernel32.dll", EntryPoint = "GetPrivateProfileString")]
private static extern int GetPrivateString(string section, string key, string def, StringBuilder buffer, int size, string path);

//Импорт функции WritePrivateProfileString (для записи значений) из библиотеки kernel32.dll
[DllImport("kernel32.dll", EntryPoint = "WritePrivateProfileString")]
private static extern int WritePrivateString(string section, string key, string str, string path);
}
```

Код класса чтения/записи INI-файлов



## ПК 1.3.

Для отладки использовалась среда Visual Studio.

В процессе разработки были выявлены и исправлены такие ошибки:

- Ошибка вывода данных из БД в DataGridView;
- Ошибка экспорта в файл Excel;
- Ошибки синтаксиса;
- Логические ошибки;
- Ошибки компиляции.

# ПК 1.4.

<b>Наименование:</b>		Ввод и удаление символов	<b>№:</b>			ТК 05			
<b>Описание:</b>		Ввод и удаление символов в текстовом поле в программе "dumb.exe"	<b>Статус:</b>			Not Started			
<b>Тестирующий:</b>		Дрюпин А.А.	<b>Дефекты №:</b>			3			
			<b>Дата:</b>			1-июн-2020			
<b>Начальные условия:</b>									
Наличие программы "dumb.exe"		В наличии	<отметка о проверке при выполнении кейса>						
Наличие справки о программе "dumb.exe"		В наличии							
<b>25 шагов</b>		<b>Число шагов по статусам:</b>			0	0	0	<b>% Complete:</b>	0%
<b>Шаг</b>	<b>Действие</b>	<b>Ожидаемый результат</b>	<b>Pass</b>	<b>Fail</b>	<b>N/A</b>	<b>Фактический результат</b>	<b>№ дефекта</b>		
1	Запускаем программу "dumb.exe"	Программа "dumb.exe" запустилась				Программа "dumb.exe" запустилась			
2	Нажимаем на кнопку "File"	Диалоговое окно с кнопками				Диалоговое окно с кнопками			
3	В всплывающем окне нажимаем кнопку "New"	Создался новый список				Создался новый список			
4	Ввод данных в текстовое поле	Ввести можно только символы "a-Z", "0-9"				Ввести можно что угодно	def1		
5	Ставим курсор в конце введенной строки в текстовом поле и нажимаем клавишу "Backspace"	Все символы были удалены				Все символы удалены не были	def2		
6	Ставим курсор в конце введенной строки в текстовом поле и нажимаем клавишу "Delete"	Текст удаляется по символам				Удаление текста по символам не происходит			
7	Выделяем весь текст в текстовом поле и нажимаем клавишу "Backspace"	Удаление всех символов из текстового поля				Удаление всех символов из текстового поля при помощи клавиши "Backspace" не работает	def3		
8	Выделяем весь текст в текстовом поле и нажимаем клавишу "Delete"	Удаление всех символов из текстового поля				Удаление всех символов из текстового поля			



# ПК 1.4.

## 6.1. Добавление новой записи в дневник

Таблица 3 – Проверка функции добавления записи

№	Входные данные		Выходные данные	Примечания
	Текст	Эмоция	Сообщение	
1	-	-	Сообщение «Не заполнен текст заметки» Заметка не добавлена	
2	“Хороший день”	-	Сообщение «Заметка добавлена» Заметка добавлена	По умолчанию эмоция «Не выбрано»
3	«12312412»	-	Сообщение «Заметка добавлена» Заметка добавлена	
4	-	Радость	Сообщение «Не заполнен текст заметки» Заметка не добавлена	
5	“Хороший день”	Радость	Сообщение «Заметка добавлена» Заметка добавлена	

# ПК 1.4.

## 7. РЕЗУЛЬТАТЫ ИСПЫТАНИЙ

Тестовые данные для проведения тестирования программы представлены в таблице 4.

### 7.1. Добавление новой записи в дневник

Таблица 4 – Проверка функции добавления записи

№	Входные данные		Ожидаемый результат	Фактический результат	Статус теста
	Текст	Эмоция			
1	-	-	Сообщение «Не заполнен текст заметки» Заметка не добавлена	Сообщение «Не заполнен текст заметки» Заметка не добавлена	Пройден
2	«Хороший день»	-	Сообщение «Заметка добавлена» Заметка добавлена	Сообщение «Заметка добавлена» Заметка добавлена	Пройден
3	«12312412»	-	Сообщение «Заметка добавлена» Заметка добавлена	Сообщение «Заметка добавлена» Заметка добавлена	Пройден
4	-	Радость	Сообщение «Не	Сообщение «Не	Пройден

№	Входные данные		Ожидаемый результат	Фактический результат	Статус теста
	Текст	Эмоция			
			заполнен текст заметки» Заметка не добавлена	заполнен текст заметки» Заметка не добавлена	
5	«Хороший день»	Радость	Сообщение «Заметка добавлена» Заметка добавлена	Сообщение «Заметка добавлена» Заметка добавлена	Пройден

⊕ Таблица 4 – Проверка пользовательского интерфейса

№	Название формы	Форма отображается	Все кнопки доступны	Реакция на нажатие кнопки	Интуитивное управление
1	Подключение к <del>на</del> интерфейсу	Да	Да	Да	Нет
2	Меню	Да	Да	Да	Да
3	Таймер работы	Да	Да	Да	Да
4	Таймер отдыха	Да	Да	Да	Да
5	Добавление новой записи в дневник	Да	Да	Да	Да
6	Вопрос о добавлении новой записи	Да	Да	Да	Нет
7	Статистика	Да	Да	Да	Да
8	Отладка	Да	Да	Да	Да



# ПК 1.4.

```
[TestMethod]
public void getPasswordStrench_UpperCase()
{
    //arrange
    string password = "P";
    int expected = 1;

    //act

    int actual = PasswordChecker.GetPasswordStrench(password);

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

```
[TestMethod]
public void getPasswordStrench_Symbol()
{
    //arrange
    string password = "!";
    int expected = 1;

    //act

    int actual = PasswordChecker.GetPasswordStrench(password);

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

```
namespace UserDataTest
{
    [TestClass]
    public class UserTest
    {
        public TestContext TestContext { get; set; }
        private User manager = new User();

        [DataSource("Microsoft.VisualStudio.TestTools.DataSource.XML",
            "data.xml",
            "User",
            DataAccessMethod.Sequential)]
        [TestMethod]
        public void UserManager_Add_FromXML()
        {
            string userId = Convert.ToString(TestContext.DataRow["userid"]);
            string telephone = Convert.ToString(TestContext.DataRow["telephone"]);
            string email = Convert.ToString(TestContext.DataRow["email"]);
            string sex = Convert.ToString(TestContext.DataRow["sex"]);
            int passport = Convert.ToInt32(TestContext.DataRow["passport"]);

            bool result = manager.Add(userId, telephone, email, sex, passport);

            Assert.IsTrue(result, "Пользователь не может быть создан");
        }
    }
}
```

## ПК 1.5.

Оптимизация кода:

- Визуальное разделение методов, функций и классов для удобства чтения кода;
- Названия переменных и элементов должны отражать смысл хранимого содержимого;
- Не должно быть избыточности кода, не используемых переменных.



# ПК 1.5.

```
namespace Auth
{
    public partial class Reg : Form
    {
        public Reg()
        {
            InitializeComponent();
        }
        private void Register(object sender, EventArgs e)
        {
            int rightOK = 0;
            String Login = textBox1.Text;
            String Password = textBox2.Text;
            String RePassword = textBox3.Text;
            String Name = textBox4.Text;
            String Passport = textBox5.Text;
            String Email = textBox7.Text;
            String Phone = textBox6.Text;
            if (Login.Length <= 0 || Passport.Length <= 0 || Name.Length <= 0 || Password.Length <= 0)
            else if (Password != RePassword) MessageBox.Show("Ошибка! Пароли не совпадают");
            else rightOK++;
            if (!Regex.IsMatch(Password, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^\a-zA-Z])$"))
            else rightOK++;
            if (!Regex.IsMatch(Phone, @"^((\+7-?)|8)?[0-9]{10}$", RegexOptions.IgnoreCase))
            else rightOK++;
            if (Passport.Length != 10) MessageBox.Show("Ошибка! Серия и номер паспорта не совпадают");
            else rightOK++;
            if (!Regex.IsMatch(Email, @"^\w+([-+.']\w+)*@\w+([-.\w+)*$"))
            else rightOK++;
        }
    }
}
```

Не оптимизированный код

```
namespace Auth
{
    public partial class Reg : Form
    {
        public Reg()
        {
            InitializeComponent();
        }
        private void Register(object sender, EventArgs e)
        {
            int rightOK = 0;
            String Login = textBox1.Text;
            String Password = textBox2.Text;
            String RePassword = textBox3.Text;
            String Name = textBox4.Text;
            String Passport = textBox5.Text;
            String Email = textBox7.Text;
            String Phone = textBox6.Text;

            if (Login.Length <= 0 || Passport.Length <= 0 || Name.Length <= 0 || Password.Length <= 0)
            else
            {
                if (Password != RePassword) MessageBox.Show("Ошибка! Пароли не совпадают");
                else rightOK++;
                if (!Regex.IsMatch(Password, @"^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[^\a-zA-Z])$"))
                else rightOK++;
                if (!Regex.IsMatch(Phone, @"^((\+7-?)|8)?[0-9]{10}$", RegexOptions.IgnoreCase))
                else rightOK++;
                if (Passport.Length != 10) MessageBox.Show("Ошибка! Серия и номер паспорта не совпадают");
                else rightOK++;
                if (!Regex.IsMatch(Email, @"^\w+([-+.']\w+)*@\w+([-.\w+)*$"))
                else rightOK++;
            }
        }
    }
}
```

Оптимизированный код



# ПК 1.5.

```
23 private void Register(object sender, EventArgs e)
24 {
25     int rightOK = 0;
26     String Login = textBox1.Text;
27     String Password = textBox2.Text;
28     String RePassword = textBox3.Text;
29     String Name = textBox4.Text;
30     String Passport = textBox5.Text;
31     String Email = textBox7.Text;
32     String Phone = textBox6.Text;
33     String SNILS = "";
34     String LastName = "";
35 }
```

0 Ошибки 2 Предупреждения 2 Сообщения

Код	Описание
CS0219	Переменной "SNILS" присвоено значение, но оно ни разу не использовано.
CS0219	Переменной "LastName" присвоено значение, но оно ни разу не использовано.

Не оптимизированный код

```
private void Register(object sender, EventArgs e)
{
    int rightOK = 0;
    String Login = textBox1.Text;
    String Password = textBox2.Text;
    String RePassword = textBox3.Text;
    String Name = textBox4.Text;
    String Passport = textBox5.Text;
    String Email = textBox7.Text;
    String Phone = textBox6.Text;

    if (Login.Length <= 0 || Passport.Length <= 0 || Name
    else
мы не найдены.
```

0 Ошибки 0 Предупреждения 0 Сообщения

Оптимизированный код



# ПК 1.5.

```
String Login = textBox1.Text;  
String Password = textBox2.Text;  
String RePassword = textBox3.Text;  
String Name = textBox4.Text;  
String Passport = textBox5.Text;  
String Email = textBox7.Text;  
String Phone = textBox6.Text;
```

Не оптимизированный код

```
String Login = Login.Text;  
String Password = Password.Text;  
String RePassword = RePassword.Text;  
String Name = Name.Text;  
String Passport = Passport.Text;  
String Email = Email.Text;  
String Phone = Phone.Text;
```

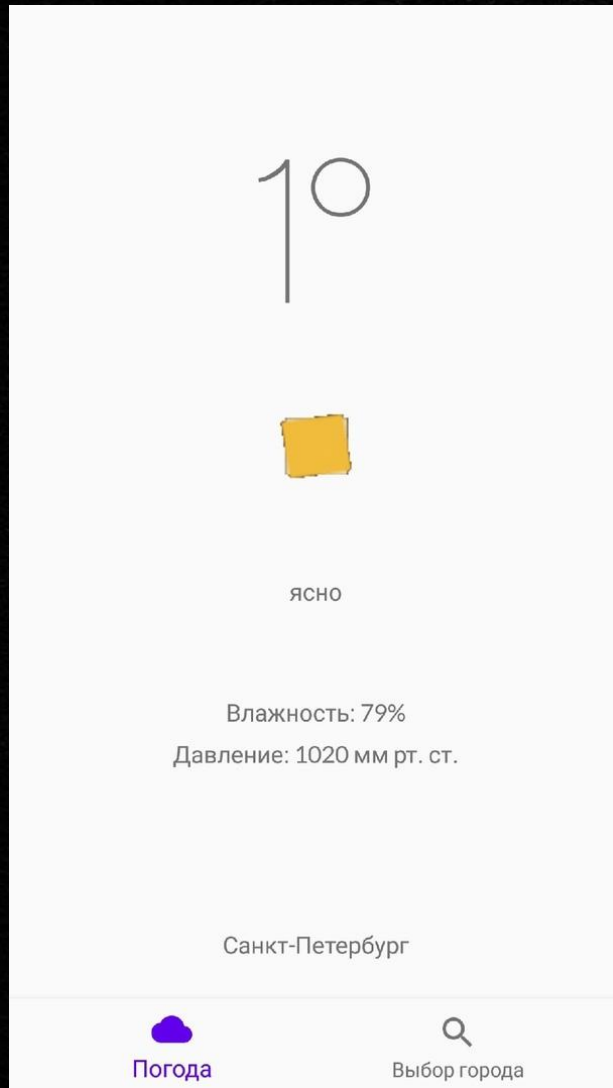
Оптимизированный код

## ПК 1.6.

В процессе освоения данной профессиональной компетенции были изучены различные элементы, использующиеся в дизайне приложений на платформу Android, а так же работа с API, различными библиотеками, работа с камерой и гироскопом. Был изучен SQLite.



# ПК 1.6.



Приложение «Perfect Weather»  
Было разработано в рамках учебной  
практики.  
Были использованы PREFERENCES,  
приложение работает с API,  
отслеживающим погоду в различных  
городах мира, в дизайне соблюдается  
принцип минимализма, для отражения  
текущих погодных условий  
используется краткая характеристика и  
GIF



# ПК 1.6.

```
class Welcome : AppCompatActivity() {  
  
    val APP_PREFERENCES = "WeatherApp"  
    val APP_PREFERENCES_CurrentCity = ""  
    //val APP_PREFERENCES_CurrentCityPosition = ""  
  
    @SuppressWarnings("SourceLockedOrientationActivity")  
    @RequiresApi(Build.VERSION_CODES.O)  
    override fun onCreate(savedInstanceState: Bundle?) {  
        requestWindowFeature(Window.FEATURE_NO_TITLE)  
        super.onCreate(savedInstanceState)  
        this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);  
        window.setFlags(  
            WindowManager.LayoutParams.FLAG_FULLSCREEN,  
            WindowManager.LayoutParams.FLAG_FULLSCREEN)  
        setContentView(R.layout.activity_welcome)  
        getSupportActionBar()?.hide()  
  
        var mSettings = getSharedPreferences(APP_PREFERENCES, Context.MODE_PRIVATE)  
  
        if(mSettings.getString(APP_PREFERENCES_CurrentCity, "") != "") {  
            val intent = Intent(this, MainActivity::class.java)  
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_NEW_TASK)  
            startActivity(intent)  
        }  
    }  
}
```

Фрагмент кода



# ПК 1.6.

```
<resources>
  <string name="app_name">PerfectWeather</string>
  <string name="title_home">Погода</string>
  <string name="title_SelectCity">Выбор города</string>
  <string name="Welcome_EnterButton">Сохранить</string>
  <string name="deleted">Удалено</string>
  <string name="saved">Сохранено</string>
  <string name="connection_error">Ошибка подключения к серверу</string>
  <string name="connection_error_parse">Ошибка получения данных</string>

  <string-array name="CityNames">
    <item>Москва</item>
    <item>Санкт-Петербург</item>
    <item>Екатеринбург</item>
    <item>Краснодар</item>
    <item>Абакан</item>
    <item>Азов</item>
    <item>Александров</item>
    <item>Алексин</item>
    <item>Альметьевск</item>
    <item>Анапа</item>
    <item>Ангарск</item>
    <item>Анжеро-Судженск</item>
    <item>Апатиты</item>
    <item>Арзамас</item>
```

Предопределенный список городов и сообщений

# Заключение

В течение второго и третьего курса мною были освоены все 6 профессиональных компетенций, входящих в ПМ.01

В данной презентации были продемонстрированы все приобретенные за два года навыки