

Оценка соответствия стандартам кодирования

1. Введение

В рамках данной практической работы была проведена оценка качества исходного кода информационной системы «Радость бухгалтера» на соответствие общепринятым стандартам кодирования для языка C#.

Целью работы является проверка соблюдения соглашений по именованию, форматированию, структуре кода и применению лучших практик разработки на платформе .NET.

2. Используемые стандарты кодирования

Оценка кода выполнялась на основе следующих официальных документов и рекомендаций:

1. **Microsoft C# Coding Conventions** — официальные соглашения по написанию кода на C#;
2. **.NET Framework Design Guidelines** — рекомендации по проектированию API и библиотек;
3. **C# Naming Guidelines** — правила именования идентификаторов.

3. Критерии оценки

Код проверялся по следующим критериям:

3.1. Именование идентификаторов

- **PascalCase** для классов, методов, свойств, событий;
- **camelCase** для локальных переменных и параметров методов;
- **_camelCase** (с подчеркиванием) для приватных полей класса;
- Осмысленные имена, отражающие назначение переменных и методов.

3.2. Форматирование и структура

- Отступы: 4 пробела (стандарт Visual Studio);
- Фигурные скобки на новой строке (стиль Allman);
- Одна строка — один оператор;
- Максимальная длина строки: 120-140 символов.

3.3. Организация кода

- Директивы using в начале файла;
- Порядок членов класса: поля → конструкторы → свойства → методы;
- Группировка по модификаторам доступа (public, private, protected).

3.4. Лучшие практики

- Использование using для работы с IDisposable (SqlConnection, SqlCommand);
- Применение LINQ для операций с коллекциями;

- Обработка исключений с помощью try-catch;
- Применение интерполяции строк (\$) вместо конкатенации.

4. Результаты оценки

Критерий	Соответствие	Примечание
Именованние классов	✓ Полное	PascalCase соблюден
Именованние методов	✓ Полное	PascalCase соблюден
Именованние приватных полей	✓ Полное	_camelCase соблюден
Именованние локальных переменных	✓ Полное	camelCase соблюден
Стиль фигурных скобок	⚠ Частичное	Стиль Allman
Отступы и форматирование	✓ Полное	4 пробела
Директивы using	✓ Полное	В начале файла
Использование using для IDisposable	✓ Полное	SqlConnection, SqlCommand
Обработка исключений	✓ Полное	try-catch блоки
Использование LINQ	✓ Полное	Where, FirstOrDefault
Интерполяция строк	✓ Полное	\$"" используется
XML-комментарии	⚠ Частичное	Можно добавить

Таблица 1 — Результаты оценки соответствия стандартам кодирования

5. Примеры соответствия стандартам

5.1. Именованние

Пример правильного именования в классе Employee:

```
public class Employee { // PascalCase для класса
    private readonly DatabaseService _db; // _camelCase для поля
    public int EmployeeID { get; set; } // PascalCase для свойства
    public string FullName => ...; // PascalCase для свойства
    public void LoadData() // PascalCase для метода
}
var employees = _db.GetAllEmployees(); // camelCase для переменной
```

5.2. Использование using для IDisposable

Корректная работа с SqlConnection:

```
using (var conn = GetConnection())
```

```
{
    conn.Open();
    using (var cmd = new SqlCommand(query, conn))
    {
        cmd.ExecuteNonQuery();
    }
}
```

5.3. Использование LINQ

Фильтрация данных с помощью LINQ:

```
dgPayroll.ItemsSource = _allPayrolls.Where(p =>
p.EmployeeName.ToLower().Contains(search) ).ToList();
```

5.4. Интерполяция строк

Использование интерполяции вместо конкатенации:

```
// Правильно: MessageBox.Show($"Ошибка загрузки: {ex.Message}", "Ошибка");
// Вместо: MessageBox.Show("Ошибка загрузки: " + ex.Message, "Ошибка");
```

6. Рекомендации по улучшению

Несмотря на высокий уровень соответствия стандартам, можно отметить следующие возможные улучшения:

4. **Добавить XML-комментарии** для публичных классов и методов для автоматической генерации документации;
5. **Рассмотреть использование async/await** для операций работы с базой данных для повышения отзывчивости интерфейса;
6. **Добавить логирование** критических операций для упрощения отладки.

7. Заключение

По результатам проведенной оценки установлено, что исходный код информационной системы «Радость бухгалтера» в полной мере соответствует официальным стандартам кодирования C# и рекомендациям Microsoft.

Код демонстрирует высокий уровень качества: правильное именование идентификаторов, консистентное форматирование, применение современных конструкций языка (LINQ, интерполяция строк), корректная работа с ресурсами через using-блоки, адекватная обработка исключений.