Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

ТРПО

**ОТЧЕТ О ТЕСТИРОВАНИИ ПРОГРАММНОГО МОДУЛЯ**

Выполнил: А.И. Валиев

Группа: ПР-32

Преподаватель: А. С. Попова

2025

## 1. Введение

### 1.1. Цель тестирования

Проверка корректности работы классов TCommand и TBearingParam, а также системы логирования, включая:

* Правильность обработки входных данных
* Корректность возвращаемых значений
* Обработку ошибочных ситуаций

### 1.2. Объекты тестирования

* Класс TCommand (обработка команд)
* Класс TBearingParam (параметры подшипника)
* Система логирования (класс Logger)

## 2. Результаты тестирования

### 2.1. Таблица результатов тестирования

| **Тестовый случай** | **Результат** | **Описание ошибки (если есть)** |
| --- | --- | --- |
| TCommand: Неверный код команды (-1) | PASS | - |
| TCommand: Код команды 1 | PASS | - |
| TCommand: Код команды 2 | PASS | - |
| TCommand: Код команды 4 | PASS | - |
| TCommand: Код команды 6 | PASS | - |
| TCommand: Код команды 20 | PASS | - |
| TBearingParam: Конструктор по умолчанию | PASS | - |
| TBearingParam: Пользовательский конструктор | PASS | - |
| TBearingParam: Установка отрицательного диаметра | PASS | Корректно обрабатывает ошибку |
| Logger: Запись обычного сообщения | PASS | - |
| Logger: Попытка записи null | PASS | Корректно генерирует исключение |

### 2.2. Выявленные проблемы

Проблем не выявлено. Все тестовые случаи выполнены успешно.

## 3. Детали тестирования

### 3.1. Тестирование класса TCommand

**Методы тестирования:**

1. Проверка реакции на неверный код команды (-1)
2. Проверка всех допустимых кодов команд (1, 2, 4, 6, 20)

**Пример теста:**

csharp

Copy

Download

[Test]

public void TestInvalidCommandCode()

{

var command = new TCommand { CommandCode = -1 };

string result = command.GetFullCommandName();

Assert.AreEqual("ОШИБКА: Неверный код команды", result);

}

### 3.2. Тестирование класса TBearingParam

**Методы тестирования:**

1. Проверка конструкторов
2. Проверка установки/получения значений
3. Проверка обработки отрицательных значений

**Пример теста:**

csharp

Copy

Download

[Test]

public void TestNegativeDiameter()

{

var bearing = new TBearingParam();

Assert.Throws<ArgumentOutOfRangeException>(() =>

{

bearing.Diameter = -1.0;

});

}

### 3.3. Тестирование системы логирования

**Методы тестирования:**

1. Проверка записи обычного сообщения
2. Проверка реакции на null
3. Проверка одновременной записи в консоль и файл

## 4. Заключение

### 4.1. Выводы

1. Все тестируемые классы работают в соответствии с требованиями
2. Обработка ошибок реализована корректно
3. Система логирования обеспечивает надежную запись событий

### 4.2. Рекомендации

1. Добавить больше boundary-тестов для проверки граничных значений
2. Реализовать тесты для многопоточной работы
3. Добавить метрики покрытия тестами