

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Локтионова Оксана Геннадьевна  
Должность: проректор по учебной работе  
Дата подписания: 09.12.2024 14:48:27  
Уникальный программный ключ:  
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

**МИНОБРАЗОВАНИЯ РОССИИ**

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

«Юго-Западный государственный университет»

(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

« 12 » 12

2022 г.



## ИНСПЕКЦИЯ И СКВОЗНОЙ ПРОСМОТР КОДА ПРОГРАММНОЙ СИСТЕМЫ

Методические указания по выполнению лабораторной работы №2 по дисциплине «Тестирование ПО»

Курск 2022



## **Цель работы**

Научиться проводить инспекции и сквозные просмотры исходного кода программ.

## **Теоретические сведения**

Инспекции и сквозные просмотры относятся к методам тестирования программы вручную. С помощью просмотра исходного кода программы можно эффективно обнаруживать ошибки в программе на ранних этапах до автоматизированного тестирования.

Инспекция и сквозной просмотр — это чтение и визуальный просмотр текста программы группой лиц с последующим обсуждением результатов проверки. Цель — найти ошибки, но не устранять их (этим занимается отладка).

Эти методы позволяют обнаруживать не только отдельные ошибки, но и группы ошибок. С их помощью можно обнаружить до 70% ошибок кодирования, но нельзя обнаружить ошибки проектирования и анализа требований. Эти методы дополняют другие способы тестирования, такие как автоматизированное тестирование. Сквозные просмотры подходят как для новых так и для модифицированных программ.

Инспекция исходного текста — это изучение текста программы группой специалистов. Обычно она проходит в несколько этапов. Сначала программист рассказывает о своей программе, отвечает на вопросы, при этом сам программист может найти ошибки в своей программе. Затем программа проверяется по списку общих возможных ошибок. Найденные ошибки фиксируются. При необходимости после исправления ошибок процедура повторяется.

### **Список общих возможных ошибок в программах.**

#### **Ошибки обращения к данным.**

1. Обращение к не инициализированным переменным.

```
int a;
```

```
int b = a + 10; // a — не инициализировано
```

2. Индекс массива выходит за границы массива.

```
int a[20];
```

```
a[20] = 1; // индекс массива от 0 до 19
```

3. Указатель или ссылка указывает на не распределенную память или объект.

```
int *b;
```

```
*b = 20; // указатель не был инициализирован или выделена память
```

4. Начальные значения переменных неправильные.

```
int a[20];
```

```
int i = 1; // неправильный начальный индекс
```

```
while (i < 20) {
```

```
    a[i] = i; // не весь массив заполняется
```

5. Неправильный класс памяти переменной (auto, static).

```
int func(int c)
```

```
{
```

```
    static int b = 10; // значение b будет инициализировано один раз
```

```
    b = b + c;
```

```
    return b;
```

}

6. Неправильный тип переменной.

```
int a = 100;
```

```
byte b = a + 200;
```

**Ошибки вычислений.**

1. Отсутствует приведение типов при выполнении арифметических операций с разными типами, например целым и вещественным.

```
int a = 10;
```

```
float b = 2.2;
```

```
a = a + b; // неявное приведение типов
```

2. Неправильный порядок вычислений в выражении (отсутствуют скобки).

```
a = a >> b + 1 ; // сначала выполняется сложение, потом сдвиг
```

3. Переполнение и потеря результата при вычислении.

```
short a = 1000;
```

```
a = a * 1000; // потеря результата из-за переполнения
```

4. Деление на ноль.

```
a = 0;
```

```
b = 10 / a;
```

5. Ошибки округления вещественных чисел.

```
a = sqrt(1.0 / 1000000); // слишком маленькое число
```

6. Значение переменной выходит за ее диапазон.

```
short b = 10 << 10; // потеря значения
```

7. Ошибки при использовании беззнаковых типов в вычислениях со знаками и наоборот.

```
int b = -1;
```

```
unsigned int a = b; // неявное преобразование, b = 0xFFFFFFFF
```

### **Ошибки сравнений.**

1. Неправильный знак отношения при сравнении.

```
int i = 0;
```

```
while (i > 10) { // ошибка, вместо > должно быть <
```

2. Неправильная логическая операция (И, ИЛИ, НЕ) в составных условиях.

```
if (a > 0 || a < 10) // сравнение на диапазон, вместо ИЛИ должно быть И
```

3. Ошибка в результате неявного приведения типа из целочисленного в логический.

```
if (a + b) // результат всегда истина если хотя бы одна переменная больше нуля
```

4. Ошибка сравнения чисел с плавающей точкой.

```
if (a == 2.0) // возможно никогда не выполнится, правильно fabs(a — 2.0) < EPSILON, где EPSILON — погрешность
```

5. Неправильный порядок условий в выражении.

```
if (*a == 1 && a != NULL) // сначала будет разыменован указатель,
```

*затем проверка на пустой указатель, ошибка если указатель пустой*

## **Ошибки в потоке управления**

1. Цикл не завершается

```
int i = 0;
```

```
while (i < 5)
```

```
    a[i] = 2;
```

2. Недостижимый код в функции

```
return a;
```

```
a = 10; // никогда не выполнится
```

3. Неправильный блок операторов

```
int i = 0;
```

```
while (i < 5)
```

```
    a[i] = 2;
```

```
    i++; // отсутствуют скобки блока
```

## **Ошибки интерфейсов**

1. Неправильный порядок аргументов

```
void Draw(int x, int y, int count);
```

*Draw(10, 0, 1); // из-за того, что типы параметров одинаковы, можно перепутать порядок, и компилятор не выдаст ошибку*

## **Ошибки ввода-вывода**

1. Отсутствие обработки ошибок ввода-вывода.

```
FILE *f = fopen(«file», «rb»);
```

```
fread(buf, size, 1, f); // отсутствует проверка, был ли открыт файл
```

### **Сквозной просмотр кода**

Процедура сквозного просмотра кода похожа на инспекцию, отличие ее заключается в том, что вместо чтения текста программы участки просмотра мысленно выполняют программу с данными из составленных тестовых наборов. Состояние программы (значение переменных) фиксируется и отслеживается.

### **Задание для лабораторной работы.**

Проведите инспекцию своей программы по приведенному выше перечню возможных ошибок. Найденные ошибки отсортируйте по категориям приведенным выше.

### **Контрольные вопросы.**

1. Что такое инспекция исходного кода?
2. Что такое сквозной просмотр исходного кода?
3. В чем заключается процедура инспекции исходного кода?
4. Какие бывают ошибки данных?
5. Какие ошибки относятся к ошибкам ввода-вывода?
6. Назовите основные ошибки при вычислениях?
7. Какие бывают ошибки управления?

## **Список литературы.**

1. Мякишев, Д. В. Принципы и методы создания надежного программного обеспечения АСУТП : методическое пособие / Д. В. Мякишев. - Москва ; Вологда : Инфра-Инженерия, 2017. - 115 с. - URL: <http://biblioclub.ru/index.php?page=book&id=466489>. – Текст: электронный.
2. Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 286 с. - URL: <http://biblioclub.ru/index.php?page=book&id=429034>. – Текст: электронный.
3. Технология программирования : учебное пособие / Ю. Ю. Громов [и др.]. - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2013. - 173 с. - URL: <http://biblioclub.ru/index.php?page=book&id=277802>. – Текст: электронный.