**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Петербургский государственный университет путей сообщения**

**Императора Александра I»**

**(ФГБОУ ВО ПГУПС)**

Факультет «Автоматизация и интеллектуальные технологии»

Кафедра «Информатика и информационная безопасность»

Лабораторная работа № 3

по дисциплине

«Подтверждение соответствия, сертификация, анализ безопасности программного обеспечения»

на тему: «Проведение испытаний на отсутствие недекларированных возможностей ПО»

|  |  |  |
| --- | --- | --- |
| **Выполнил**  **обучающийся**  Курс 5  Группа КИБ-012 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Г.Е. Груздев |
|  |  |  |
| **Проверил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | С.В. Корниенко |

Санкт-Петербург

2024

|  |  |
| --- | --- |
| **УТВЕРЖДЕНО** | **СОГЛАСОВАНО** |
| Генеральный директор  ООО «Стандарты на века»  (подпись) А.М. Игнатьев  30.12.2024 | Руководитель отдела испытаний  АО «ИНСЕК-СЗ»  (подпись) Э.Я. Кондратьев  30.12.2024 |

ОАО «Фонтаны»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «САДОВНИКИ»

ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

ЛИСТ УТВЕРЖДЕНИЯ

RU.001001.01.10-01 51 02-ЛУ

Листов 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | **СОГЛАСОВАНО:** | |  | ОАО «Кесадилья»  Инженер-конструктор  (подпись) Попов А.А  30.12.2024 | |  |

Санкт-Петербург

2024

УТВЕРЖДЕНО

RU.001001.01.10-01 51 02-ЛУ

ОАО «Фонтаны»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «САДОВНИКИ»

ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

RU.001001.01.10-01 51 02

Листов 18

Санкт-Петербург

2024

**Аннотация**

Настоящий документ представляет собой Программу и методику испытаний Программного обеспечения, реализующего концепцию многозадачности и распределенных вычислений.

Документ разработан согласно ГОСТ 19.301-79, структура и оформление документа соответствуют ГОСТ 19.105-78, правила оформления листа утверждения и титульного листа устанавливаются по ГОСТ 19.104-78, выполнен печатным способом согласно ГОСТ 19.106-78.

# Оглавление

[1. Объект испытаний 1](#_Toc153828187)

[2. Цель испытаний 2](#_Toc153828188)

[3. Требования к программе 3](#_Toc153828189)

[4. Требования к программной документации 4](#_Toc153828190)

[5. Состав и порядок испытаний 5](#_Toc153828191)

[6. Методика испытаний 6](#_Toc153828192)

[Лист регистрации изменений 9](#_Toc153828193)

1. Объект испытаний

В настоящем документе Программы и методики испытаний объектом испытаний является программное обеспечение «Садовники».

Полное наименование испытуемой программы:

Программное обеспечение «Садовники».

Сокращённое название: ПО «Садовники»

Обозначение испытуемой программы: RU.001001.01.10-01

1. Цель испытаний

Испытания проводятся с целью проверки соответствия требованиям Руководящему документу Государственной технической комиссии при Президенте Российской Федерации от 4 июня 1999 года № 114 «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей» по 3 уровню контроля.

1. Требования к программе

ПО должна исполнять следующий алгоритм:

Имеется пустой участок земли (двумерный массив) и план сада, который необходимо реализовать. Эту задачу выполняют два садовника, которые не хотят встречаться друг с другом. Первый садовник начинает работу с верхнего левого угла сада и перемешается слева направо, сделав ряд, он спускается вниз. Второй садовник начинает работу с нижнего правого угла сада и перемещается снизу вверх, сделав ряд, он перемещается влево. Если садовник видит, что участок сада уже выполнен другим садовником, он идет дальше. Садовники должны работать параллельно.

- Программа должна создавать общий ресурс в виде двумерного массива фиксированного размера (минимум 5x5), изначально заполненного нулями.

- Программа должна запускать два независимых потока, которые обрабатывают массив по разным алгоритмам (например, один поток заполняет массив числами 1, другой - числами 2).

- Программа должна обеспечивать синхронизированный доступ потоков к общему ресурсу с использованием примитива синхронизации – критической секции.

- Программа должна корректно отображать текущее состояние массива в консоль после каждой операции изменения.

- Потоки должны выполнять обработку с заданной задержкой (не менее 500 мс) между изменениями элементов массива, чтобы пользователю была видна работа программы.

- По завершении работы потоков программа должна корректно освобождать ресурсы и завершать выполнение.

1. Требования к программной документации

Состав программной документации должен включать в себя:

1. Руководство по эксплуатации ПО «Садовники»: RU.001001.01.10-01 91 01.
2. Спецификация ПО «Садовники»: RU.001001.01.10-01.
3. Описание программы ПО «Садовники»: RU.001001.01.10-01 13 01.
4. Текст программы ПО «Садовники»: RU.001001.01.10-01 12 01.
5. Пояснительная записка ПО «Садовники»: RU.001001.01.10-01 81 01.

Программная документация должна быть оформлена в соответствии с ГОСТ 19.105-78 «Единая система программной документации. Общие требования к программным документам».

1. Состав и порядок испытаний

Место проведения испытаний: АО «ИНСЕК-СЗ»

(Российская Федерация, г. Санкт-Петербург, наб. Реки Смоленки, д.14)

Период проведения испытаний: с 30.11.2024 по 31.12.2024.

## Технические и программные средства, используемые во время испытаний

Испытания проводятся на оборудовании и программном обеспечении заказчика в среде Visual Studio Code с дополнением PVS-studio.

В качестве рабочей станции для тестирования ПО должен быть выделен персональный компьютер с аппаратными характеристиками не ниже следующих:

* объем оперативной памяти от 1024 Мб;
* свободное дисковое пространство от 40 Гб;
* процессор Intel Pentium 4 531 с тактовой частотой 3.0 ГГц.

На станции для тестирования ПО должно быть установлено следующее ПО:

* ОС Windows 11;
* Среда Visual Studio Code с предустановленным пакетом C++
* Программа PVS
* Текстовый редактор Word 2019
* Программа для вычисления хеш-суммы и контрольной суммы файлов HashTab;

## Порядок проведения испытаний

Проверочные испытания включают в себя:

1. Проверку состава и содержания документации, представляемой Заявителем, на соответствие требованиям ГОСТ: ГОСТ 19.202-78, ГОСТ 19.402-78, ГОСТ 19.401-78, ГОСТ 19.404-79;
2. Контроль исходного состояния ПО с указанием значений контрольных сумм загрузочных модулей и исходных текстов программ, входящих в состав ПО;
3. Статистический анализ исходных текстов программ, включающий:
   1. Контроль полноты и отсутствия избыточности исходных текстов ПО на уровне файлов;
   2. Контроль соответствия исходных текстов ПО его объективному (загрузочному) коду;
   3. Контроль полноты и отсутствия избыточности исходных текстов ПО на уровне функциональных объектов (процедур);
   4. Контроль связей функциональных объектов (модулей, процедур, функций) по управлению;
   5. Контроль связей функциональных объектов (модулей, процедур, функций) по информации;
   6. Контроль информационных объектов различных типов;
   7. Формирование перечня маршрутов выполнения функциональных объектов (процедур, функций).
4. Динамический анализ исходных текстов программ, включающий:
   1. Контроль выполнения функциональных объектов (процедур, функций);
   2. Сопоставление фактических маршрутов выполнения функциональных объектов (процедур, функций) и маршрутов, построенных в процессе проведения статического анализа (п. 3).
5. Отчётность по результатам проведения испытаний в форме протокола, содержащего результаты:
   1. Контроля исходного состояния ПО;
   2. Контроля полноты и отсутствия избыточности исходных текстов, контролируемого ПО на уровне файлов;
   3. Контроля соответствия исходных текстов ПО его объектному (загрузочному) коду.
   4. Контроля полноты и отсутствия избыточности исходных текстов, контролируемого ПО на уровне функциональных объектов (процедур);
   5. Контроля связей функциональных объектов (модулей, процедур, функций) по управлению;
   6. контроля связей функциональных объектов (модулей, процедур, функций) по информации;
   7. контроля информационных объектов различных типов (например, локальных переменных, глобальных переменных, внешних переменных и т.п.);
   8. формирования перечня маршрутов выполнения функциональных объектов (процедур, функций);
   9. контроля выполнения функциональных объектов (процедур, функций);
   10. сопоставления фактических маршрутов выполнения функциональных объектов (процедур, функций) и маршрутов, построенных в процессе проведения статического анализа.

Отчётность, содержащая сведения о прохождении испытаний, оформляет документально в форме протокола испытаний с последующей выдачей технического заключения о соответствии (не соответствии) ПО требованиям Руководящего Документа по 3 уровню контроля отсутствия НДВ.

1. Методика испытаний

При проведении каждой методики испытаний производится пометка об успешном (или проваленном) прохождении испытания напротив названия методики. Результаты прохождения должны заноситься в соответствующую графу Протокола испытаний (приложение 1).

## Контроль состава и содержания документации

Проверка комплектности и состава документации производится непосредственно представителем Заказчика.

В ходе проверки производится соотнесение поставленной комплектации с заявленным перечнем программной документации, необходимой для проведения сертификационных испытаний, состав и содержание которой соответствует пунктам:

* Спецификация (ГОСТ 19.202-78), содержащая сведения о составе ПО и документации на него;
* Описание программы (ГОСТ 19.402-78), содержащее основные сведения о составе (с указанием контрольных сумм файлов, входящих в состав ПО), логической̆ структуре и среде функционирования ПО, а также описание методов, приёмов и правил эксплуатации средств технологического оснащения при создании ПО;
* Исходные тексты программ (ГОСТ 19.401-78), входящих в состав ПО.
* Пояснительная записка (ГОСТ 19.404-79), содержащая основные сведения о назначении компонентов, входящих в состав ПО, параметрах обрабатываемых наборов данных (подсхемах баз данных), формируемых кодах возврата, описание используемых переменных, алгоритмов функционирования и т.п.

Критерием положительного результата испытания считается наличие перечисленной выше документации, и её соответствие требованиям нормативных документов.

## Контроль исходного состояния ПО

Контроль состояния исходного ПО выполняется с помощью программы HashTab для вычисления хэш-значений.

Проводится сравнение рассчитанных контрольных сумм для каждого файла, входящего в состав ПО, с контрольными суммами, приведёнными в описание программы и с эталонным образцом разработчика, находящемся по адресу https://company.ru/products/gardeners

Критерием положительного результата испытания считается полное совпадение вычисленных хеш-сумм.

## Статический анализ

### Контроль полноты и отсутствия избыточности исходных текстов на уровне файлов

Полнота представленных исходных текстов на уровне файлов ПО подтверждается фактом успешной компиляции и сборки исследуемого ПО с учётом результатов контроля соответствия исходных текстов загрузочному коду (лог компиляции).

Критерием соответствия является соответствие заявленного в составе ПО перечня файлов с логом успешной компиляции.

Для контроля избыточности исходных текстов на уровне файлов, определяется список файлов, составляющих ПО и на основе анализа полученных исходных данных определяются избыточные файлы, анализируется их назначение и обоснованность включения в состав программного обеспечения.

Критерием положительного результата испытания является соответствие заявленного в составе ПО перечня файлов с логом успешной компиляции и отсутствие файлов, которые не используются при выполнении функциональных требований ПО.

### Контроль соответствия исходных текстов ПО его объектному (загрузочному) коду

Контроль соответствия исходных текстов ПО его объектному (загрузочному) коду осуществляется методом создания загрузочных модулей из представленных исходных текстов ПО с представленным образцом разработчика.

Критерием положительного результата испытания является однозначное соответствие контрольных сумм файлов компиляции (загрузочного кода) с представленным образцом разработчика.

Если хеш-сумма не совпала, необходимо проанализировать hex-код исполняемого файла с помощью ПО «HxD Hex Editor». Если анализ показывает, что изменения не критичны, то необходимо указать «программы можно считать условно тождественными».

### Контроль связей функциональных объектов по управлению

Сформировать таблицу 1 «Связи функциональных объектов по управлению», характеризующей отношения вызова и достижимости между всеми парами функциональных объектов. В приведенном примере показывается, что f() 1 вызывает f() 2, а f() 2 вызывает объект f() 3.

Формирование таблицы:

Начиная со второго столбца необходимо указать все функциональные объекты программы. В первом столбце, начиная со второй строки также указать все функциональные объекты программы.

Если функциональный объект осуществляет вызов другого функционального объекта, то сведения о данной паре заносятся в таблицу «Связи функциональных объектов по управлению» и обозначаются символом «+». Столбец соответствует функциональному объекту, который вызывает другой функциональный объект, а строка – вызываемому объекту.

Критерием положительного результата испытания является совпадение полученных результатов с описанием, приведённым в программной документации на исследуемое программное обеспечение.

Таблица 1. Связи функциональных объектов по управлению

|  |  |  |  |
| --- | --- | --- | --- |
|  | f() 1 | f() 2 | f() 3 |
| f() 1 |  |  | + |
| f() 2 | + |  |  |
| f() 3 |  | + |  |

Вывод о соответствии данному пункту указать в соответствующей графе таблицы «Интегральная оценка испытаний»: «соответствует» или «не соответствует».

### Контроль связей функциональных объектов по информации

Сформировать таблицу 2 «Связи функциональных объектов по информации», характеризующей процесс совместного использования одних информационных объектов в различных функциях исследуемого программного обеспечения.

Таблица 2 «Связи функциональных объектов по информации» отображает факты совместного использования информационных объектов различными функциональными объектами.

Таблица 2. «Связи функциональных объектов по информации»

|  |  |  |  |
| --- | --- | --- | --- |
|  | f() 1 | f() 2 | f() 3 |
| f() 1 |  |  |  |
| f() 2 | Имя глобальной переменной |  |  |
| f() 3 |  | Имя глобальной переменной |  |

Критерием положительного результата испытания является отсутствие в таблице «Связи функциональных объектов по информации» связей, не указанных в программной документации на поставляемое ПО.

### Контроль информационных объектов

Сформировать таблицу 3 «Информационные объекты». На данном этапе выполнения методики статического анализа выполняется проверка программных текстов на наличие глобальных переменных, контрольных секций и/или повторного использования переменных, сведения о которых заносятся в таблицу 3.

Таблица 3. Информационные объекты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функциональный объект | Имя | Тип | Граничные значения | Область видимости | Размерность |
| sting message\_1 | message\_1 | sting | - | локальная | до 2 ГБ |
| int num\_1 | num\_1 | int | от -2147483648 до 2147483647 | глобальная | 4 байта |

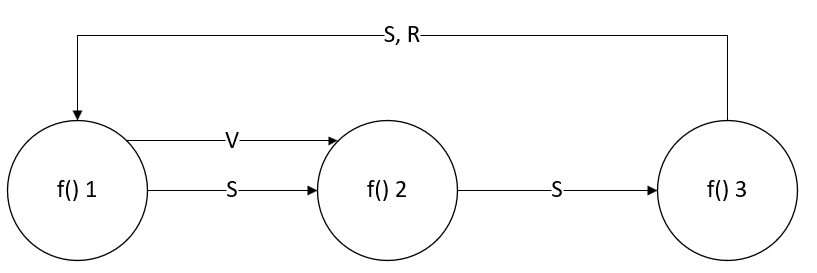
### Критерием положительного результата испытания является отсутствие информационных объектов, не заявленных в предоставленной программной документации

### Формирование перечня маршрутов выполнения функциональных объектов

Сформировать граф по управлению «Маршруты выполнения функциональных объектов на уровне функций», необходимый для определения вызывающих и вызываемых функциональных объектов, на основе таблиц 1-3.

Граф по управлению «Маршруты выполнения функциональных объектов на уровне функций» формируется на основе таблиц п. 6.3.3, п. 6.3.4, п. 6.3.5.

Рисунок 1. «Маршруты выполнения функциональных объектов на уровне функций»



где f() – название функции, S, R, V – передаваемые переменные.

Критерием положительного результата испытания является наличие в графе всех вершин, соответствующих связям между функциональными объектами, указанными в программной документации.

## Динамический анализ исходных текстов программ

### Контроль выполнения функциональных объектов

Сигнатура датчика: LOG\_FUNCTION\_CALL(function\_name, #имена\_переменых)

// Макрос для регистрации вызова функции в файл

#define BEG\_LOG\_FUNCTION\_CALL(function\_name, ...) \

do { \

std::ofstream log\_file("function\_calls.log", std::ios\_base::app); \

if (std::string(#\_\_VA\_ARGS\_\_).empty()) { \

log\_file << "Вход в функцию: " << #function\_name << ". Функция не получает никаких параметров"<< std::endl; \

} else { \

log\_file << "Вход в функцию: " << #function\_name << "; Параметры: " << #\_\_VA\_ARGS\_\_ << std::endl; \

} \

} while(0)

// Макрос для регистрации окончания функции в файл

#define EDN\_LOG\_FUNCTION\_CALL(function\_name, ...) \

do { \

std::ofstream log\_file("function\_calls.log", std::ios\_base::app); \

if (std::string(#\_\_VA\_ARGS\_\_).empty()) { \

log\_file << "Выход из функции: " << #function\_name << ". Функция не возвращает никаких параметров"<< std::endl; \

} else { \

log\_file << "Выход из функции: " << #function\_name << "; Параметры: " << #\_\_VA\_ARGS\_\_ << std::endl; \

} \

} while(0)

Контроль выполнения функциональных объектов (процедур, функций) подразумевает внесение в исходный код исследуемого программного обеспечения так называемых «датчиков», выполнение программного обеспечения в соответствии с официальной эксплуатационной документацией, фиксацию поведения исследуемого программного обеспечения в заявленных разработчиком режимах функционирования.

Необходимо осуществить сборку проекта, который будет содержать в исходном коде исследуемой программы «датчики». Результат работы «датчиков» о проходе контрольных точек необходимо вывести во внешний файл. Совокупность информации (о передаче управления функциональных объектов друг другу, а также передачи информации между функциональными объектами) образует сведения о реальных маршрутах выполнения исследуемого программного обеспечения.

Составить граф выполнения исследуемого программного обеспечения на основе логов внешнего файла, необходимо организовать систему записи информации о прохождении контрольных точек.

Критерием положительного результата испытания является выполнение всех заявленных в программной документации функций, реализующие функциональные требования к испытуемому ПО. Контроль выполнения определяется на основании генерируемого внешнего файла о прохождении контрольных точек.

### Сопоставление фактических маршрутов выполнения функциональных объектов и маршрутов, построенных в процессе проведения статического анализа

Составить граф выполнения исследуемого программного обеспечения на основе логов внешнего файла, полученных в предыдущем пункте.

При сопоставлении фактических маршрутов проводят анализ соответствия сведений, полученных в результате статического анализа, с данными рисунка 1. «Маршруты выполнения функциональных объектов на уровне функций».

Критерием соответствия является совпадение маршрутов в графе, выполнения функциональных объектов, составленного в пункте 6.3.6 с данными из сформированного в пункте 6.4.1 файла.

Критерием положительного результата испытания является совпадение маршрутов в графе, построенном на основе логов внешнего файла, графу выполнения функциональных объектов, построенному в процессе проведения статического анализа.

**7 ИНТЕГРАЛЬНАЯ ОЦЕНКА ИСПЫТАНИЙ**

После проведения испытаний, требуется составить протокол об их проведении с указанием полученных результатов:

|  |  |  |
| --- | --- | --- |
| № | Наименование испытания | Соответствие 3му классу |
| 1. | Контроль состава и содержания документации |  |
| 2. | Контроль исходного состояния ПО |  |
| 3. Статический анализ исходных текстов программ | | |
| 3.1. | Контроль полноты и отсутствия избыточности исходных текстов |  |
| 3.2. | Контроль соответствия исходных текстов ПО его объектному (загрузочному) коду |  |
| 3.3. | Контроль связей функциональных объектов по управлению |  |
| 3.4. | Контроль связей функциональных объектов по информации |  |
| 3.5. | Контроль информационных объектов |  |
| 3.6. | Формирование перечня маршрутов выполнения функциональных объектов |  |
| 4. Динамический анализ исходных текстов программ | | |
| 4.1. | Контроль выполнения функциональных объектов |  |
| 4.2. | Сопоставление фактических маршрутов выполнения функциональных объектов и маршрутов, построенных в процессе проведения статического анализа |  |

Если по итогам проведенных испытаний все десять проверок имеют положительный результат, то исследуемая программа по контролю отсутствия недекларированных возможностей тестового ПО соответствует 3 уровню контроля.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Заместитель начальника  АО «ИНСЕК-СЗ» | | | | |  | | | | К.В. Дуализм | | |
| Лист регистрации изменений | | | | | | | | | | |
| Номер листов (страниц) | | | | | Всего листов (страниц) | № докум | Входящий номер сопроводит. документа и дата | | Подпись | Дата |
| Изм | изменен | заменен | новых | аннулир |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | 9 | 10 |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  | |  |  |