**УТВЕРЖДАЮ**

**\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Старостин**

**"\_\_ "\_\_\_\_\_\_\_\_\_\_ 2022 г.**

**ПРОГРАММА И МЕТОДИКА КОМПЛЕКСНЫХ ИСПЫТАНИЙ**

**на опытно-конструкторскую работу**

**«Разработка базового функционала библиотеки 2D геометрического ядра»**

**№ 02068143.00221 51**

г. Н. Новгород, 2022 г.

СОДЕРЖАНИЕ

[1. ОБЪЕКТ ИСПЫТАНИЙ 3](#_Toc125503287)

[2. ЦЕЛЬ ИСПЫТАНИЙ 3](#_Toc125503288)

[3. ОРГАНИЗАЦИЯ ПРОВЕДЕНИЯ ИСПЫТАНИЙ 3](#_Toc125503289)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ 4](#_Toc125503290)

[5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 5](#_Toc125503291)

[6. МЕТОДИКА ИСПЫТАНИЙ 5](#_Toc125503292)

[7. ОТЧЁТНОСТЬ 10](#_Toc125503293)

[8. ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ 10](#_Toc125503294)

Настоящая программа и методика приемочных испытаний определяет порядок проведения испытаний программных средств для решения задач плоской геометрии, возникающих при топологическом проектировании.

# ОБЪЕКТ ИСПЫТАНИЙ

Испытанию подлежит программная библиотека «gkernel2d».

# ЦЕЛЬ ИСПЫТАНИЙ

Испытания проводятся с целью проверки библиотеки «gkernel2d» на соответствие требованиям, изложенным в техническом задании (ТЗ) [1].

# ОРГАНИЗАЦИЯ ПРОВЕДЕНИЯ ИСПЫТАНИЙ

Проведение испытаний осуществляется силами ННГУ им. Лобачевского, студентами группы 3821М1ПИ на техническом оборудовании и верификационном базисе Исполнителя. Состав комиссии определяется по согласованию с Заказчиком.

# ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Испытания проводятся в соответствии с пунктами методики испытаний, приведенными в табл. 1.

Таблица 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № пункта | Наименование проверки | | Пункт ТЗ | Пункт методики |
| 1 | Проверка состава ПО | | 5.1 | 6.1 |
| 2 | Проверка требований к библиотеке «gkernel2d» | Требования к геометрическим объектам | 5.2 | 6.2.1 |
| Требования к контейнерам | 5.3,  5.3.1-5.3.6 | 6.2.2 |
| Требования к дереву поиска | 5.4 | 6.2.3 |
| Требования к функции пересечения двух отрезков | 5.5.1 | 6.2.4 |
| Требования к функции поиска пересечений набора отрезков | 5.5.2 | 6.2.5 |
| Требования к алгоритмам преобразования структур | 5.5.3 | 6.2.6 |
| Требования к организации операций над слоями | 5.6 | 6.2.7 |

# 5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Программная документация «gkernel2d» включает в себя:

* руководство программиста

# 6. МЕТОДИКА ИСПЫТАНИЙ

**6.1. Проверка состава ПО**

**Проверка наличия библиотеки gkernel**

* + - 1. Загрузить архив «gkernel-windows-v0.3-release.zip» с GitHub.
      2. Распаковать архив.
      3. Перейти в директорию «bin\_release».
      4. Проверить наличие библиотеки «gkernel.dll».

Испытание считается пройденным, если в каталоге «bin\_release» присутствует файл «gkernel.dll».

**6.2. Проверка требований к библиотеке «gkernel2d»**

**6.2.1. Требования к геометрическим объектам**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к геометрическим объектам, изложенным в пункте 5.2 ТЗ.

**Требования к функционалу**

Набор требуемых функций для объектов «Точка» и «Отрезок» реализован, интерфейсы описаны в руководстве программиста [2] в разделах Point (точка), Segment (отрезок).

**Порядок проведения тестирования**

Проверка соответствия работы функций ожидаемому результату реализована в юнит-тестах.

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_geom\_structures.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 2 | 2 passed | 0 failed | 0 skipped

[doctest] assertions: 4 | 4 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.2. Требования к контейнерам**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к контейнерам, изложенным в пунктах 5.3, 5.3.1-5.3.6 ТЗ.

**Требования к функционалу**

Набор требуемых функций контейнеров, представленных в пунктах 5.3, 5.3.1 – 5.3.6 ТЗ, изложены в руководстве программиста в разделах: VertexChain(цепь), Circuit (контур), SegmentsSet (набор отрезков), SegmentsLayer (слой отрезков), CircuitsSet (набор контуров), CircuitsLayer (слой контуров).

**Порядок проведения тестирования**

Проверка соответствия работы функций ожидаемому результату реализована в юнит-тестах.

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_containers.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 5 | 5 passed | 0 failed | 0 skipped

[doctest] assertions: 19 | 19 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.3. Требования к дереву поиска**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к дереву поиска, изложенным в пункте 5.4 ТЗ.

**Требования к функционалу**

Набор требуемых функций дерева поиска изложены в руководстве программиста в разделе RBTree.

**Требования к временным характеристикам**

Проверка соответствия требованиям к сложности операций дерева поиска изложены в научно-техническом отчёте [3].

**Порядок проведения тестирования**

Проверка соответствия работы функций ожидаемому результату реализована в юнит-тестах.

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_rbtree.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 4 | 4 passed | 0 failed | 0 skipped

[doctest] assertions: 22 | 22 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.4. Требования к функции поиска пересечения двух отрезков**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к функции пересечения двух отрезков, изложенным в пункте 5.5.1 ТЗ.

**Требования к временным характеристикам**

Проверка соответствия требованиям к сложности операции поиска пересечения двух отрезков изложены в научно-техническом отчёте.

**Порядок проведения тестирования**

Проверка корректности работы функционала реализована в юнит-тесте.

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_two\_segments\_intersection.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 3 | 3 passed | 0 failed | 0 skipped

[doctest] assertions: 6 | 6 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.5. Требования к функции поиска пересечений набора отрезков**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к функции поиска пересечений набора отрезков, изложенным в пункте 5.5.2 ТЗ.

**Порядок проведения тестирования**

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_intersection.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 6 | 6 passed | 0 failed | 0 skipped

[doctest] assertions: 6 | 6 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.6. Требования к алгоритмам преобразования структур**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к алгоритмам преобразования структур, изложенным в пункте 5.5.2 ТЗ.

**Порядок проведения тестирования**

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_converter.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 7 | 7 passed | 0 failed | 0 skipped

[doctest] assertions: 86 | 86 passed | 0 failed |

[doctest] Status: SUCCESS!

**6.2.7 Требования к организации операций над слоями**

Данный сценарий определяет порядок проведения проверки соответствия требованиям к организации операций над слоями, изложенным в пункте 5.6 ТЗ.

**Порядок проведения тестирования**

1. Открыть каталог «bin\_release».
2. Открыть консоль в текущем каталоге.
3. Запустить юнит-тесты, введя команду: test\_pipeline.exe.

После запуска ожидается сообщение в консоли о статусе прохождения тестов.

Испытание считается пройденным, если в консоли выведено сообщение:

[doctest] test cases: 1 | 1 passed | 0 failed | 0 skipped

[doctest] assertions: 187 | 187 passed | 0 failed |

[doctest] Status: SUCCESS!

# 7. ОТЧЁТНОСТЬ

По всем видам испытаний оформляется протокол испытаний, который подписывается членами комиссии, проводившими испытания.

Протокол испытаний утверждается председателем комиссии.

# 8. ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Техническое задание на опытно-конструкторскую работу «Разработка базового функционала библиотеки 2D геометрического ядра».
2. Руководство программиста библиотеки gkernel2d.
3. Научно-технический отчёт на опытно-конструкторскую работу «Разработка базового функционала библиотеки 2D геометрического ядра».