МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук Кафедра «Информационная безопасность»

Утверждено на заседании кафедры

«Информационная безопасность»

« » \_ 202 г., протокол №

Заведующий кафедрой

А.А. Сычугов

#### МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**по выполнению курсовой работы по дисциплине (модулю)**

**«Языки программирования»**

**основной профессиональной образовательной программы высшего образования – программы специалитета**

по специальности:

10.05.03 Информационная безопасность автоматизированных систем с профилем:

Разработка автоматизированных систем в защищенном исполнении Форма обучения: очная

Идентификационный номер образовательной программы: 100503-01-21

Тула 2021 год

#### ЛИСТ СОГЛАСОВАНИЯ

**методических указаний к курсовой работе по дисциплине (модулю)**

**Разработчик:**

Басалова Г.В., к.т.н., доцент

*(ФИО, должность, ученая степень, ученое звание) (подпись)*

## СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_TOC_250012)

1. [ЦЕЛИ И ЗАДАЧИ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ 4](#_TOC_250011)
2. [ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ 4](#_TOC_250010)
   1. [ТЕМАТИКА КУРСОВОЙ РАБОТЫ 4](#_TOC_250009)
   2. [ЗАДАНИЕ И ИСХОДНЫЕ ДАННЫЕ К КУРСОВОЙ РАБОТЕ 4](#_TOC_250008)

[2.З. ОБЪЕМ КУРСОВОЙ РАБОТЫ 5](#_TOC_250007)

* 1. [ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ 5](#_TOC_250006)
  2. [ЗАЩИТА КУРСОВОЙ РАБОТЫ 6](#_TOC_250005)

1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ6
   1. [ПЛАН ПОСТРОЕНИЯ И СОДЕРЖАНИЕ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ 6](#_TOC_250004)
   2. [МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ОТДЕЛЬНЫХ РАЗДЕЛОВ КУРСОВОЙ РАБОТЫ 6](#_TOC_250003)
   3. [ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ 7](#_TOC_250002)
2. [БИБЛИОГРАФИЧЕСКИЙ СПИСОК 9](#_TOC_250001)

[ПРИЛОЖЕНИЕ 10](#_TOC_250000)

## ВВЕДЕНИЕ

Целями освоения дисциплины является теоретическая и практическая подготовка специалистов к деятельности, связанной с проектированием и экс- плуатацией систем защиты информации автоматизированных систем.

Задачами освоения дисциплины являются:

* получение теоретических знаний и практических навыков проекти- рования приложений на различных языках программирования;
* освоение принципов тестирования и отладки приложений;
* приобретение навыков использования современных технологий разработки алгоритмов и программ

## ЦЕЛИ И ЗАДАЧИ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа является важным этапом изучения дисциплины "Языки программирования". Ее выполнение должно способствовать достижению сле- дующих целей: расширение и закрепление знаний по курсу, развитие умения самостоятельно работать с источниками специализированной информации, по- лучение необходимого опыта в проектировании алгоритмов и разработки при- кладных решений с помощью различного программного обеспечения и изучен- ных технологий.

## ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

##### *ТЕМАТИКА КУРСОВОЙ РАБОТЫ*

Тематика курсовой работы соответствует профилю направления и отража- ет актуальные потребности теории и практики использования алгоритмов и языков программирования при разработке различных автоматизированных си- стем. Реализация может происходить с использованием языков программиро- вания C++, C#, Pascal, Python и др. Темы курсовых работ могут отражать сле- дующие направления исследований и разработок:

1. Проектирование консольного приложения
2. Проектирование приложения с оконным интерфейсом
3. Исследование алгоритмов обработки данных
4. Изучение и сравнительный анализ алгоритмов обработки данных

##### *ЗАДАНИЕ И ИСХОДНЫЕ ДАННЫЕ К КУРСОВОЙ РАБОТЕ*

Задание и исходные данные на курсовую работу выдаются руководите- лем (преподавателем) на специальном типовом бланке (см. приложение). От- дельные данные могут быть выбраны студентом самостоятельно и согласованы

с руководителем. В задании указываются:

* тема работы;
* исходные данные к работе;
* рекомендуемая литература;
* сроки выполнения и защиты проекта.

##### *2.З. ОБЪЕМ КУРСОВОЙ РАБОТЫ*

Курсовая работа состоит из пояснительной записки, задания и рецензии. Рекомендуемый объем пояснительной записки – от 25 страниц (без учета при- ложений). Увеличение объема (не более 40 страниц) может потребоваться для полного раскрытия темы курсовой работы.

##### *ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ*

Работа над курсовой работой выполняется по графику, определяемому руководителем. В целях ее планомерного выполнения рекомендуется следую- щий график работы.

|  |  |  |
| --- | --- | --- |
| *Неделя* | *Содержание работ* | *Результаты работы* |
| 1 -2 | Получение и ознакомление с за- данием. | Заполненный бланк задания. |
| 3–4 | Изучение литературы и других исходных материалов. | Обзор литературы. Выбор основных источников |
| 5–6 | Выбор конкретных алгоритмов и способов реализации | Алгоритмы |
| 7-10 | Реализация алгоритмов | Программный код КР |
| 11 - 13 | Тестирование программной части работы | Сформированный набор те- стов |
| 14–15 | Оформление пояснительной за- писки и сдача на проверку. | Пояснительная записка. |
| 16 | Защита курсовой работы. |  |

По всем вопросам, возникающим при выполнении курсовой работы, необходимо обращаться к руководителю, назначаемому кафедрой. Все резуль- таты работы должны предъявляться руководителю для проверки в соответствии с графиком выполнения работы.

Руководитель также проверяет пояснительную записку. Все ошибки, недоработанные места указываются и разъясняются студенту. Если работа удо- влетворяет требованиям, то руководитель подписывает пояснительную записку, тем самым допуская его к защите.

##### *ЗАЩИТА КУРСОВОЙ РАБОТЫ*

Курсовая работа представляется на кафедру для проверки за неделю до ее защиты. При положительной оценке руководителем студент допускается к за- щите работы перед комиссией.

Защита – форма проверки выполненной работы. Курсовая работа защи- щается публично в присутствии студентов перед комиссией, назначаемой заве- дующим кафедрой. Руководитель работы является членом комиссии. При за- щите работы сначала студент выступает с сообщением продолжительностью 8– 10 минут по существу работы. Затем по докладу и содержанию пояснительной записки студенту задаются вопросы членами комиссии, на которые он должен ответить.

Курсовая работа оценивается по 100-балльной системе с учетом:

* обоснованности объема (соответствия заданию) и качества выполнения кур- совой работы;
* степени самостоятельности при выполнении работы;
* качества оформления пояснительной записки и соответствия их требовани- ям данных методических указаний;
* качества защиты и правильности ответов на вопросы.

Студент, не представивший в срок курсовую работу или не защитивший ее по неуважительной причине, считается имеющим академическую задолжен- ность.

В случае получения неудовлетворительной оценки студенту выдается но- вое задание.

## МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

##### *ПЛАН ПОСТРОЕНИЯ И СОДЕРЖАНИЕ РАЗДЕЛОВ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ*

Пояснительная записка должна иметь структуру: титульный лист, зада- ние, содержание, введение, основную часть, заключение, список использован- ных источников, приложения.

К пояснительной записке прилагается незаполненный бланк рецензии (см. приложение).

##### *МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ОТДЕЛЬНЫХ РАЗДЕЛОВ КУРСОВОЙ РАБОТЫ*

Форма титульного листа приведена в приложении. Задание, оформленное в соответствии с требованиями, помещается в пояснительной записке на втором листе. Страница с заданием имеет номер 2. Задание должно быть подписано руководителем работы и студентом.

Содержание включает наименования всех разделов, подразделов и пунк-

тов с указанием номеров страниц.

Примерное содержание курсовой работы на тему «Изучение и сравни- тельный анализ алгоритмов обработки данных» имеет вид:

Введение

* + 1. Постановка задачи
    2. Описание исследуемого (-ых) алгоритма (-ов) (Теоретическая часть) 3 Возможности применения алгоритма

4 Решение задач с использованием алгоритма Заключение

Список литературы Приложения

Введение должно отражать историю рассматриваемого вопроса или ме- тода, его теоретическую и практическую значимость (объем введения – ок. 1 страницы).

Основная часть содержит постановку задачи, изложение теоретических положений с необходимыми выводами, варианты реализации алгоритма для решения задач, описание разработанных задач для демонстрации применения алгоритма, обоснование и описание разработанных тестов для проверки рабо- тоспособности решений.

Пример оформления основной части КР приведен в приложении. Если задание на КР включает составление задач по изучаемым алгоритмам, то опи- сание задач должно включать следующие части: условие, описание входных и выходных данных, алгоритм решения, решение на одном или двух языках про- граммирования, примеры входных и выходных данных, описание групп тестов для проверки работоспособности решения.

Заключение должно содержать краткие выводы по работе, оценку полу- ченных результатов. В нем отмечается, какие новые знания и навыки получены студентом при выполнении курсовой работы, указываются перспективы разви- тия, мероприятия по совершенствованию разработанных алгоритмов, компо- нентов и т.д.

В приложения включаются таблицы с исходными данными, тексты про- грамм, громоздкие результаты работы программы или результаты, не представ- ленные в разделе, схемы алгоритмов.

##### *3.3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ*

Записка оформляется на листах формата A4.

Для основного текста рекомендуется использовать тип шрифта Times New Roman с размер шрифта 14 пт. Переносы слов в заголовках не допускают- ся. Каждый раздел следует начинать с новой страницы. Расстояние между заго- ловком и текстом должно быть 10-15 мм.

Разделы должны иметь порядковую нумерацию в пределах всей записки

и обозначаться арабскими цифрами без точки в конце. Точка не ставится и в конце всего заголовка. Введение и заключение не нумеруются. Подразделы ну- меруют арабскими цифрами в пределах каждого раздела. Номер подраздела со- стоит из номера раздела и подраздела, разделенных точкой. Например, «2.3» – подраздел 3 раздела 2. Пункты нумеруют в пределах подраздела, например

«1.1.2» — пункт 2 подраздела 1 раздела 1.

Иллюстрации, кроме таблиц, обозначаются снизу словом "Рис." и нуме- руются последовательно арабскими цифрами в пределах раздела (при неболь- шом числе рисунков допускается сквозная нумерация по всей работе). Номер иллюстрации должен состоять из номера раздела и порядкового номера иллю- страции, например, «Рис. 1.2» – рисунок 2 раздела 1. Если в записке приведена одна иллюстрация, то ее не нумеруют и слово "Рис." не пишут. Иллюстрация должна иметь наименование. При необходимости к иллюстрации приводят по- яснительные надписи (подрисуночный текст). Наименование иллюстрации по- мещают над ней. Пример подписи под рисунком: «Рис. 1.2 Структурная схема программы». И иллюстрация, и подпись под ней выравниваются по центру.

Цифровой и структурированный материал рекомендуется оформлять в виде таблиц. Таблицы нумеруют в пределах раздела. Каждая таблица должна иметь заголовок, который помещается над самой таблицей. Заголовок и слово

«Таблица» пишут с прописной буквы. В правом верхнем углу таблицы над со- ответствующим заголовком помещают надпись, например, «Таблица 1.2» – таблица 2 раздела 1. Если в записке одна таблица, то ее не нумеруют и слово "Таблица" не пишут.

Формулы в записке нумеруют арабскими цифрами в пределах раздела (при небольшом количестве – до 10 – в пределах всей работы). В первом случае номер формулы состоит из номера раздела и порядкового номера формулы в разделе. Номер помещают в правой стороне страницы на уровне формулы в круглых скобках, например, (3.1) – формула 1 раздела 3.

Уравнения и формулы отделяются от текста по вертикали одинарным межстрочным интервалом. Если соотношение не умещается в одну строку, то оно должно быть перенесено после математических знаков. Формулы констру- ируются встроенным построителем формул текстового редактора или внешним приложением (например, MS Equation, MathType). Базовый размер шрифта в формулах должен быть таким же, что и размер шрифта основного текста.

Ссылки в тексте на источник указывают порядковым номером по списку источников, выделенным квадратными скобками, например, «в соответствии с [2]». На все источники в списке литературы (возможно, кроме ГОСТ) в работе должны быть ссылки в виде [2], [2, 3], [2-5]. Ссылки на формулы указывают порядковым номером формулы в круглых скобках, например, «в формуле (1.2)». Ссылки на иллюстрации указывают порядковым номером иллюстрации, например, «рис. 1.2». На все таблицы тоже должны быть ссылки в тексте. Сло- во «таблица» пишется полностью, если таблица не имеет номера, и сокращен- но, если имеет номер, например, «табл. 1.2.». В повторных ссылках на таблицы и иллюстрации следует сокращенно указывать слово «смотри», например, «см. табл. 1.2.».

Список использованных источников должен содержать перечень источ-

ников, использованных при выполнении курсовой работы. Источники следует располагать в порядке появления ссылок в тексте записки (в порядке цитирова- ния) или в алфавитном порядке. Сведения об источниках, включенных в спи- сок, необходимо оформлять в соответствии с требованиями ГОСТ 7.1-2003.

Приложения располагают в порядке появления ссылок на них в тексте. Каждое приложение следует начинать с новой страницы с указанием в правом верхнем углу слова «ПРИЛОЖЕНИЕ», написанного прописными буквами; оно должно иметь содержательный заголовок. Если в записке несколько приложе- ний, их нумеруют арабскими цифрами, например, «ПРИЛОЖЕНИЕ 2». Рисун- ки, таблицы, формулы, помещенные в приложении, нумеруют арабскими циф- рами в пределах каждого приложения, например, «Рис. П 1.2» – рисунок 2 при- ложения 1.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 7.32-2017 Отчет о научно-исследовательской работе. Общие требования и правила оформления М: Стандартинформ, 2018. 33 с.
2. ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления М: Стандартинформ, 2008. 23 с.
3. Шилдт Г. С++: базовый курс, 3-е издание. : Пер. с англ. – М.: «Изда- тельский дом «Вильямс», 2005. – 624 с.
4. Вирт Н. Алгоритмы + структуры данных = программы. М.; Mиp, 1985.

– 281 с.

1. Шлее, М. Профессиональное программирование на С++ / М.Шлее .— СПб. : БХВ-Петербург, 2005 .— 544с. : ил. + 1 CD .— (В подлиннике).
2. Страуструп, Б. Язык программирования Си++: Спец. изд./ Б.Страуструп; Пер.сангл. С.Анисимова, М.Кононова; Подред. Ф.Андреева, А.Ушаков . — М. : Бином, 2004 .— 1098с.
3. Шилдт, Г. C# :Учеб.курс / Г.Шилдт; Пер.сангл.А.Падалки .— М. и др. : Питер, 2003 .— 512 с.
4. Секунов, Н.Ю. Разработка приложений на С++ и С# / Н.Ю.Секунов.— М. и др. : Питер, 2003 .— 608с. — (Б-ка программиста) .
5. [Рихтер, Д.](http://library.tsu.tula.ru/cgi-bin/zgate.exe?ACTION=follow&amp;SESSION_ID=3432&amp;TERM=%D0%A0%D0%B8%D1%85%D1%82%D0%B5%D1%80%2C%20%D0%94.%5B1%2C1004%2C4%2C101%5D&amp;LANG=rus)Windows для профессионалов : Создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows / Д.Рихтер;Пер.сангл.:Ю.Е.Е.Купцевич,А.Р.Врублевский;Подобщ.ред.Ю.Е.Купце вича .— 4-е изд. — М.и др. : Питер, 2004 .— 722с.

## ПРИЛОЖЕНИЕ

В приложении представлены образец оформления основной части рабо- ты, формы титульного листа пояснительной записки курсовой работы, бланка задания и бланка рецензии.

#### Пример состава и оформления некоторых разделов основной части

**курсовой работы**

**1 Постановка задачи**

Требуется создать библиотеку классов для работы с геометрическими объекта- ми и составить задачи по вычислительной математике. Для этого необходимо:

* Изучить основы объектно-ориентированного программирования на языке C++.
* Рассмотреть основные понятия, такие как инкапсуляция, наследо- вание, полиморфизм, класс, объект и другие.
* Изучить общий формат объявления класса, написание функций и перегрузку операторов.
* Изучить ограничения разных типов данных в языке С++.
* Изучить темы связанные с вычислительной математикой.

**2 Теоретическая часть (*фрагмент*)**

Объектно-ориентированное программирование построено на понятии класса.

Класс — это базовая единица инкапсуляции, которая обеспечивает меха- низм создания объектов. Класс – это структурированный тип, включающий в себя в качестве элементов типизированные данные и функции, применяемые по отношению к этим данным.

Объекты — это экземпляры класса.

Инкапсуляция — это такой механизм программирования, который свя- зывает воедино код и данные, которые он обрабатывает, чтобы обезопасить их как от внешнего вмешательства, так и от неправильного использования.

Полиморфизм (от греческого слова polymorphism, означающего "много форм") — это свойство, позволяющее использовать один интерфейс для целого класса действий.

#### …

*Раздел в данном примере сокращен. В каждом конкретном варианте объем этого раздела варьируется в зависимости от индивидуальной темы работы*

#### 3 Создание библиотеки классов для работы с геометрическими объектами

Повторив основные понятия ООП, можем приступить к созданию классов для работы с геометрическими объектами.

Класс точка (вектор) будет содержать координаты x и y точки (вектора). Мы можем так же определить сложение и вычитание векторов, косое и скалярное произведение, квадрат расстояния между точками и другие функции и операто- ры.

Класс прямая создадим для хранения уравнения прямой вида:



Будем хранить коэффициенты a, b и с. Так же можем получить эти коэффици- енты по двум точкам, принадлежащим данной прямой.

Описание разработанных классов (*фрагмент*):

class pt

{

double x, y;

//long long int x, y; public:

pt(double x = 0, double y = 0) : x(x), y(y) {}

//pt(long long int x = 0, long long int y = 0) : x(x), y(y) {} pt operator +(pt b)

{

pt res;

res.x = x + b.x; res.y = y + b.y; return res;

}

pt operator -(pt b)

{

pt res;

res.x = x - b.x;

res.y = y - b.y; return res;

}

// *и т.д.*

};

#### 4 Составление и решение задач по вычислительной математике с исполь- зованием разработанных классов

Полученную библиотеку классов будем использовать в написании и решении задач по вычислительной математике.

##### *Пример оформления задачи*

**Задача ХХ.** Загадка планеты Титон

Планета Титон очень похожа на Землю. На ней тоже есть океаны, материки, острова. Планету населяют титонцы, которые больше всего на свете любят пу- тешествовать. Но с недавних пор корабли титонцев стали пропадать в таин- ственных треугольниках, заполненных густым туманом. К счастью, загадка треугольников была разгадана. Чтобы безопасно проплыть сквозь такой тре- угольник, нужно плыть из любого угла точно по биссектрисе.

Треугольники появляются неожиданно, и корабли путешественников всегда оказываются на одном из его углов. В таком случае нужно незамедлительно менять курс. Помогите титонцам научиться безопасно преодолевать все таин- ственные треугольники.

По данным координат трех точек треугольника ABC определите коэффициенты a, b и c для задания уравнения биссектрисы угла BAC (уравнение имеет вид *a*x

+ *b*y + *c* = 0).

Формат ввода. Три строки входных данных содержат по два вещественных числа, разделенных пробелом: xi и yi – координаты каждой точки треугольника ABC (все числа заданы в виде десятичного числа с плавающей точкой не более чем из 16 цифр, не превышают по модулю 10000).

Первая строка содержит координаты корабля попавших в беду путешествен- ников (координату точки A(x1; y1) таинственного треугольника). Вторая и тре- тья строки содержат координаты двух оставшихся точек треугольника – B(x2; y2) и С(x3; y3).

Формат вывода. Выведите три целых числа a, b и c, подходящих под уравнение биссектрисы (*a*x + *b*y + *c* = 0).

*Алгоритм решения: ….*

Рассмотрим *группы тестов* для проверки решения данной задачи. Коэффици- ент *с* биссектрисы равен 0. Биссектриса проходит через начало координат.

|  |  |  |
| --- | --- | --- |
| 0 0  0 1 | 0 0  4 6 | 0 0  0 3.5 |

|  |  |  |
| --- | --- | --- |
| -1 0 | 4 -1 | 2.5 0 |
| -0.7071067932881648  -0.7071067932881648  0.0000000000000000 | -0.3605966625409265  0.9327218211049204  0.0000000000000000 | -0.7071067811865475  0.7071067811865475  0.0000000000000000 |

Коэффициент *b* равен 0. Биссектриса параллельна оси Ox. При проверке отве- тов с помощью чекера этот случай стоит рассматривать отдельно, чтобы избе- жать ошибок, связанных с делением на 0.

|  |  |  |
| --- | --- | --- |
| 0 0  -3 3  10 10 | 0 0  3 -3  -6 -6 | 2.2 0.1  1.8 0.7  2.6 0.7 |
| -1.0000000000000000  -0.0000000000000001  0.0000000000000000 | 1.0000000000000000  0.0000000000000000  -0.0000000000000000 | -1.0000000000000000  -0.0000000000000001  2.2000000000000002 |

Коэффициент *a* равен 0. Биссектриса параллельна оси Oy. Случай так же следу- ет рассматривать отдельно.

|  |  |  |
| --- | --- | --- |
| 0 0  4 4  2 -2 | 0 0  -3 3  -8 -8 | 2.2 0.1  2.4 -0.2  2.6 0.7 |
| 0.0000000000000000  1.0000000000000000  0.0000000000000000 | -0.0000000000000001  -1.0000000000000000  0.0000000000000000 | 0.0000000000000003  1.0000000000000000  -0.1000000000000007 |

Все коэффициенты не равны 0. Биссектриса не параллельна осям координат.

|  |  |  |
| --- | --- | --- |
| -3 1  -10 -3  -6 -7 | 0.65 0.9  4.02 7.04  8.66 2.9 | 2.4 -3.6  5.8 -4.8  2.8 -4.2 |
| 0.7614751617463874  -0.6481940897935687  2.9326195750327311 | -0.6105515760049598  0.7919764977812533  -0.3159203235999042 | 0.6139406135149205  0.7893522173763262  1.3682105101189654 |

Так как коэффициенты могут отличаться вплоть до умножения на какую-либо константу, для проверки правильности ответов следует использовать чекер:

#include "testlib.h"

#include <cmath>

using namespace std; const double eps = 1e-8;

int main(int argc, char\* argv[]) { registerTestlibCmd(argc, argv);

Решение с использованием класса pt.

double a = ouf.readDouble(); double b = ouf.readDouble(); double c = ouf.readDouble(); double A = ans.readDouble(); double B = ans.readDouble(); double C = ans.readDouble(); double x1 = 0, x2 = 10; double y1, y2;

if (abs(B) > eps) //Коэффициент b не равен 0

{

y1 = (-A \* x1 - C) / B;

y2 = (-A \* x2 - C) / B;

if ((fabs(a \* x1 + b \* y1 + c)<eps)&&(fabs(a \* x2 + b \* y2 + c) < eps)) { quitf(\_ok, "The result is correct.");

}

else {

quitf(\_wa, "Wrong answer: expected = %f, %f, %f, found = %f, %f, %f", A, B, C, a, b, c);

}

}

else {

if (abs(A)>eps) //Коэффициент a не равен 0

{ x1 = x2 = -C / A;

if ((fabs(a \* x1 + c) < eps) && (fabs(b) < eps)) { quitf(\_ok, "The result is correct.");

}

else {

quitf(\_wa, "Wrong answer: expected = %f, %f, %f, found = %f, %f, %f", A, B

, C, a, b, c);

}

}

}

}

int main()

{

pt a, b, c, p;

cin >> a >> b >> c;

pt v1 = b - a, v2 = c - a; v1 = v1 / (sqrt(v1.dist2()));

v2 = v2 / (sqrt(v2.dist2())); p = v1 + v2;

p = p / sqrt(p.dist2()); pt a0;

line A(a0, p);

cout << fixed << setprecision(16) << A << ' ' << cross(a, p); return 0;

}

*Пример описания групп тестов:*

Тесты:

№1 - 5 – a, b, c, l, r в пределах ±100, запросов не более 10, переменная a от 0 до 1 с точностью 10-1 .

№6 - 10 – a, b, c, l, r в пределах ±100, запросов не более 100, переменная a от 0 до 1 с точностью 10-2.

№11 - 15 – a, b, c в пределах ±1000, l, r в пределах ±100, запросов не более 1000, переменная a от 0 до 1 с точностью 10-3.

№16 - 30 – a, b, c в пределах ±1000, l, r в пределах ±10000, запросов не более 100000, переменная a = 0.

#### МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Тульский государственный университет»**

Кафедра информационной безопасности

# ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

### «Языки программирования»

на тему

\_

Автор работы

студент гр. \_

(дата, подпись) (фамилия и инициалы)

Руководитель работы

(дата, подпись) (должность) (фамилия и инициалы)

Работа защищена

Члены комиссии

(дата)

с оценкой \_

\_

(дата, подпись) (должность) (фамилия и инициалы)

\_

(дата, подпись) (должность) (фамилия и инициалы)

\_

(дата, подпись) (должность) (фамилия и инициалы)

Тула 20

УТВЕРЖДАЮ

Зав. кафедрой ИБ

А.А. Сычугов

« \_»

20 г.

# ЗАДАНИЕ

на курсовую работу по дисциплине

### «Языки программирования»

студенту гр. Тема работы

\_

(фамилия, имя, отчество)

Входные данные

\_

Задание получил \_

(подпись) (дата)

График выполнения работы

\_

Замечания консультанта

К защите. Консультант работы

(подпись) (дата)

#### РЕЦЕНЗИЯ

на курсовую работу (курсовой проект) студента Тульского государственного университета,

гр.

(фамилия, имя, отчество)

по дисциплине

на тему:

Тематика работы (задание)

(соответствует/не соответствует)

профилю (направленности)

обязательной программы, а также в полной мере способствует формированию необходимых компетенций (установленных в рабочей программе) у обучающихся.

Содержание работы Объем работы Оформление работы

(соответствует/не соответствует)

(соответствует/не соответствует)

(соответствует/не соответствует)

заданной тематике. для раскрытия темы.

установленным требованиям.

Тема работы раскрыта в

(полной/неполной)

мере.

Использованная при выполнении работы (проекта) литература

(актуальна/не актуальна)

Качество приложений (при наличии)

Замечания:

установленным требованиям.

(соответствует/не соответствует)

(отсутствуют или перечисляются замечания рецензента к работе)

Качество выполнения работы свидетельствует о

(недостаточном/пороговом/повышенном)

уровне

сформированности необходимых компетенций (установленных в рабочей программе).

Работа выполнена на

(высоком/среднем/низком)

уровне и, при соответствующей защите

может быть оценена на ‹‹ ››.

**Рецензент Дата**

**/ /**

**подпись должность, звание, Ф.И.О.**