

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ И ОФОРМЛЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

2 СЕМЕСТР

Выполнение лабораторных работ подразумевает командную работу студентов в парах. Перед началом написания программного кода рекомендуется внимательно изучить задание, разбить её на подзадания и продумать решение каждого этапа. Работа считается выполненной, когда преподавателю продемонстрирована работа программы и оформлен отчет.

Отчет о лабораторной работе – технический документ, который содержит систематизированные данные о лабораторной работе, описывает теорию, используемую в лабораторной работе, ход лабораторной работы, расчеты и результаты, полученные в ходе лабораторной работы.

Отчет о лабораторной работе состоит из следующих основных элементов:

- ✓ Титульный лист
- ✓ Цель работы
- ✓ Теоретические сведения
- ✓ Расчетно-графическая часть
- ✓ Выводы по работе

Титульный лист является первой страницей отчета по лабораторной работе и служит источником информации, необходимой для поиска и обработки документа. Титульный лист обязательно должен содержать:

- ✓ Наименование вышестоящей организации
- ✓ Наименование типа учебного заведения
- ✓ Наименование учебного заведения
- ✓ Название дисциплины, по которой проводится лабораторная работа
- ✓ Номер лабораторной работы
- ✓ Название лабораторной работы
- ✓ Данные о группе и студенте (студентах), выполнивший(-их) эту работу
- ✓ Данные о преподавателе, проверяющего отчет
- ✓ Город и год

При проверке преподавателем студенческих отчетов по лабораторным работам на титульном листе преподавателем записываются замечания по отчету. Поэтому в случае необходимости переоформления отчета или внесения в содержание отчета исправлений титульный лист остается первоначальным (не заменяется новым) для того, чтобы при вторичной проверке отчета преподаватель видел все предыдущие замечания.

Цель работы указывается в точной формулировке, как указано в вашем варианте.

Теоретические сведения указываются в зависимости от поставленной задачи. В данной части требуется указать, какие математические инструменты и методы планируется использовать для решения поставленной задачи.

Расчетно-графическая часть является основной в отчете о лабораторной работе. В начале этой части указываются подзадачи, на которые была разбита основная задача и определяются входные и выходные данные. Затем приводится блок-схема (или блок-схемы) алгоритма, который решает поставленную задачу, а также словесное описание алгоритма. Далее необходимо указать расчет сложности алгоритма методом BigO.

Для второго задания иерархия классов и структура проекта должна быть представлена в виде UML-диаграммы

Если представлено несколько алгоритмов, объединенных впоследствии в один, то расчет сложности общего алгоритма. После блок-схем приводится листинг программы на языке C++. Листинг должен быть приведен равновеликим шрифтом (например, Courier new) кегелем не меньше 8. Необходимо также указать, какие проблемы и ошибки возникли в процессе написания кода и отразить, какие действия были предприняты для их решения. В конце расчетно-графической части необходимо вставить снимки экрана с примерами результатов работы вашей программы (минимум 3 теста).

Выводы о проделанной работе должны содержать анализ работы: насколько в процессе разработки программы была уменьшена сложность, обоснование окончательного выбора тех или иных функций и операторов, имеющих значение для решения поставленной задачи. (Например, почему задача реализована с помощью цикла с предусловием for, а не с постусловием while).

Каждый раздел отчета должен иметь свой номер и заголовок, которые напечатаны жирным шрифтом 14 кегелем. Весь отчет (за исключением листинга) оформляется шрифтом Times New Roman.

Блок-схемы оформляются в соответствии с ГОСТ 19.701-90, а отчет должен соответствовать требованиям ГОСТ Р 7.0.97-2016.

В ходе выполнения работы приветствуется творческий подход к решению задачи. Преподавателем оценивается не только возможность программы решить поставленную задачу, но и способность программы обрабатывать ошибки и реагировать на них.

Требования к файлам с исходным кодом и к самому коду. Ваша программа для каждого задания должна быть реализована в виде трех файлов:

- заголовочный .h (.hpp) файл, в котором объявлены ваши классы и интерфейсы работы с ними, а также ваши функции или шаблоны для них;

- файл .cpp с реализацией методов вашего класса и функций;
- файл .cpp с основной программой.

ВАЖНО! Реализация шаблонных классов должна быть в том же файле, в котором объявлен их интерфейс.

К отчету должны быть приложены все файлы исходного кода, а также скомпилированный файл .exe (для Windows) или .out (для NIX). Имена файлов должны быть записаны только строчными латинскими буквами, для разделения можно использовать подчёркивание (_) или дефис (-). Используйте тот разделитель, который используется в проекте. Если единого подхода нет — используйте "_".

Общие правила именования

1. Используйте имена, который будут понятны даже людям из другой группы.
2. Имя должно говорить о цели или применимости объекта.
3. Не экономьте на длине имени, лучше более длинное и более понятное (даже новичкам) имя.
4. Поменьше аббревиатур, особенно если они незнакомы вне вашего проекта.
5. Используйте только общеизвестные аббревиатуры.
6. Не сокращайте слова.

В целом, длина имени должна соответствовать размеру области видимости. Например, **n** — подходящее имя внутри функции в 5 строк, однако при описании класса это может быть коротковато.

В слове первая буква может быть заглавной (зависит от стиля: "camel case" или «Pascal case»), остальные буквы — строчные. Например, предпочтительно `ConnectTcp()`, нежелательно `ConnectTCP()`.

Имена типов начинаются с прописной буквы, каждое новое слово также начинается с прописной буквы. Подчёркивания не используются: **MyTheBestClass**, **MyTheBestStruct**.

Имена всех типов — классов, структур, псевдонимов, перечислений, параметров шаблонов — именуются в одинаковом стиле. Имена типов начинаются с прописной буквы, каждое новое слово также начинается с прописной буквы. Подчёркивания не используются.

Имена переменных (включая параметры функций) и членов данных пишутся строчными буквами с подчёркиванием между словами. Члены данных классов (не структур) дополняются подчёркиванием в конце имени. Например: **a_local_variable**, **a_struct_data_member**, **a_class_data_member**.

Константные объекты объявляются как `constexpr` или `const`, чтобы значение не менялось в процессе выполнения. Имена констант начинаются с символа «k», далее идёт имя в смешанном стиле (прописные и

строчные буквы). Подчёркивание может быть использовано в редких случаях, когда прописные буквы не могут использоваться для разделения.

Обычные функции именуются в смешанном стиле (прописные и строчные буквы); функции доступа к переменным (сеттеры и геттеры) должны иметь стиль, похожий на целевую переменную.

Задание. В соответствии с вашим номером варианта, вам необходимо реализовать класс, а также написать тестовую программу, которая использует ваши объекты вашего класса. При проверке вашей работы преподаватель также может использовать ваш класс в соответствии с объявленными вами возможностями. Постарайтесь предусмотреть как можно больше различных ситуаций в вашей программе для отладки.

В **Работе №2** требуется создать иерархию классов, используя наследование. Необходимо написать программу, иллюстрирующую применение всех методов ваших классов. Прежде чем приступить к написанию программ, продумайте (или уточните у преподавателя), какие необходимы функции в каждом из классов (может, где-то необходимо считать координаты, где-то площади и объемы, а где-то хранить фамилии и года поступления): как в базовом, так и в классах-наследниках. Также продумайте, что следует поместить в закрытые (а, возможно, защищенные) переменные. Предусмотрите возможность переопределения методов базового класса в производном.

Работа №3 является общей для всех студентов и заключается в написании шаблона класса матрицы, в котором определены базовые операции над матрицами (сложение, вычитание, умножение на число, произведение матриц, транспонирование, нахождение обратной матрицы и вычисление определителя).

ВАРИАНТ 1.

Написать свой класс и реализовать его на языке C++

Работа №1. Написать класс, описывающий параболу и определить для него методы, которые позволят перемещать параболу на заданную величину вдоль оси Ох и оси Оу. Предусмотрите метод, который возвращает точки пересечения двух парабол и метод, возвращающий вершину параболы. Перегрузите оператор битового сдвига так, чтобы информация о параболе выводилась в поток в виде « $ax^2 + bx + c$ »

Работа №2. Полином → Квадратичная функция

ВАРИАНТ 2.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс параллелограмм, который хранит только декартовы координаты его трёх вершин.

Конструктор вызывает набор функций, которые принимают 3 группы координат (или три точки) и проверяют, чтобы каждая из координат x и y находилась в первом квадранте в диапазоне от -100000 до 100000. Это множество функций также должно проверять, что переданные координаты определяют параллелограмм. Должны быть предусмотрены функции-члены, вычисляющие длины сторон параллелограмма, периметр и площадь. Включите функцию, которая определяла бы, не является ли параллелограмм прямоугольником. Напишите метод, который возвращает прямоугольник такой же площади с минимально возможным периметром. Обеспечьте сравнение параллелограммов между собой по их площадям.

Перегрузите операцию вывода в поток для печати всей информации об объекте

Работа №2. Уравнение → Линейное уравнение → Квадратное уравнение

ВАРИАНТ 3.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс прямоугольник со сторонами, параллельными осям координат. Класс имеет атрибуты длина и ширина, каждый из которых по умолчанию равен 1. Он имеет функции-члены, которые вычисляют периметр и площадь прямоугольника. Он имеет функции записи и чтения для длины и ширины. Обеспечьте сравнение прямоугольников между собой по их периметру. Напишите метод класса, который принимает в

качестве аргумента еще один прямоугольник и возвращает новый прямоугольник – пересечение исходных двух.

Перегрузите операцию вывод в поток для печати всей информации о прямоугольнике.

Работа №2. Функция вещественного аргумента \rightarrow Квадратичная функция

ВАРИАНТ 4.

Написать свой класс и реализовать его на языке C++

Работа №1. Напишите класс, который описывает логарифм по произвольному основанию. Перегрузите необходимые арифметические операторы для работы с логарифмом и обеспечьте корректный вывод его в поток и чтение из потока в объект вашего класса.

Работа №2. Точка \rightarrow Правильный треугольник \rightarrow Правильная треугольная призма

ВАРИАНТ 5.

Написать свой класс и реализовать его на языке C++

Работа №1. Напишите класс для перестановок из n объектов по k . Перегрузите помимо необходимых операторы сложения и умножения. Перегрузите оператор битового сдвига для корректного вывода информации об объекте вашего класса в поток.

Для справки:

$$C_n^k = \frac{n!}{(n-k)!k!}$$
$$C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$$
$$C_n^m \cdot C_m^k = C_n^k \cdot C_{n-k}^{m-k}$$

Работа №2. Точка \rightarrow Круг \rightarrow Сфера

ВАРИАНТ 6.

Написать свой класс и реализовать его на языке C++

Работа №1. Реализуйте класс Дата, который будет хранить информацию о дне, месяце и годе, а также о дне недели. Определите и реализуйте методы определения дня недели, прибавления или вычитания дней, недель, месяцев и лет. Реализуйте также метод, который определяет ближайший день встречи выпускников (первая суббота февраля). Перегрузите логические операторы

сравнения, ввода из потока и вывода в поток. Объект вашего класса должен выводиться в формате: ДД.ММ.ГГГГ.

Работа №2. Треугольник → Треугольная пирамида → Усеченная треугольная пирамида

ВАРИАНТ 7.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс Rectangle (прямоугольник). Класс имеет атрибуты length (длина) и width (ширина), каждый из которых по умолчанию равен 1. Он имеет методы, которые вычисляют периметр и площадь прямоугольника. Он имеет функции записи и чтения для длины и ширины. Функции записи должны проверять, что длина и ширина – числа с плавающей запятой, находящиеся в пределах от 0,0 до 20000,0.

Перегрузите операцию вывод в поток для печати всей информации о прямоугольнике.

Работа №2. Прямолинейное движение делится на равномерное и равноускоренное. Равноускоренное в свою очередь может реализовываться свободным падением по вертикали. Реализуйте иерархию классов, описывающие указанные виды движения.

ВАРИАНТ 8.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс треугольник, хранящий только декартовы координаты вершин. Конструктор вызывает набор функций, которые принимают 3 группы координат и проверяют, чтобы каждая из координат x и y находилась в первом квадранте в диапазоне. Функции также должны проверять, чтобы треугольник не «схлопывался» в прямую линию. Должны быть предусмотрены функции-элементы, вычисляющие длину сторон, периметр и площадь треугольника. Включите функцию, которая определяла бы, не является ли треугольник равнобедренным, равносторонним или прямоугольным.

Перегрузите операцию вывода в поток для печати всей информации о треугольнике.

Работа №2. Спортсмен → Легкоатлет → Спринтер

ВАРИАНТ 9.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс треугольник, хранящий только декартовы координаты вершин. Конструктор вызывает набор функций, которые принимают 3 группы координат и проверяют, чтобы каждая из координат x и y находилась в первом квадранте в диапазоне. Функции также должны проверять, чтобы треугольник не «схлопывался» в прямую линию. Должны быть предусмотрены функции-элементы, вычисляющие радиус вписанной и описанной окружности. Включите функцию, которая определяла бы, находится ли центр описанной окружности внутри треугольника, не является ли треугольник правильным или прямоугольным.

Перегрузите операцию вывода в поток для печати всей информации о треугольнике

Работа №2. Студент → Математик → Математик-программист

ВАРИАНТ 10.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс прямая призма, хранящий только декартовы координаты вершин основания и высоту призмы. Конструктор принимает массив точек, образующих полигон (Точки следует упорядочить, чтобы из них действительно получился полигон) и высоту призмы. Должны быть предусмотрены функции-элементы, вычисляющие длины ребер, периметр и площадь основания, а также площадь боковой поверхности, площадь поверхности и объем призмы.

Перегрузите операцию вывода в поток так, чтобы она печатала, какая фигура лежит в основании, и ее основные характеристики.

Работа №2. Файл → Текстовый файл

ВАРИАНТ 11.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс "Студент", в котором будут храниться данные о ФИО студента, номер группы, специальность, Журнал оценок (максимум 5 предметов, по каждому не более 20 оценок). Предусмотрите методы, которые возвращают средний балл по каждому предмету, среднюю оценку по всем предметам, лучший предмет (с самым высоким средним баллом) и худший предмет (с самым низким средним баллом). Если таких предметов окажется несколько, то метод должен вернуть тот, по которому

больше всего оценок. Предусмотрите также все необходимые методы установки и чтения данных-элементов.

Перегрузите оператор битового сдвига для вывода в поток информации о студенте.

Работа №2. Функция вещественного аргумента → Показательная функция

ВАРИАНТ 12.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс правильный многоугольник, который хранит число вершин и их координаты. Конструктор вызывает набор функций, которые проверяют, чтобы число вершин было не менее 3, чтобы многоугольник был правильным, и в случае ошибки устанавливают значения всех вершин в 0. Должны быть предусмотрены функции-члены, вычисляющие периметр, площадь многоугольника. Напишите метод, который осуществляет поворот в положительном направлении вашей фигуры на заданный угол в радианах.

Перегрузите оператор битового сдвига для вывода в поток всей информации об объекте

Работа №2. Среди накопителей информации можно выделить такие классы, как жесткий диск и флеш-карта. Среди жестких дисков, в свою очередь, можно выделить класс съемных дисков.

ВАРИАНТ 13.

Написать свой класс и реализовать его на языке C++

Работа №1. Создайте класс, описывающий параболу на плоскости. Напишите методы, которые будут возвращать нули параболы при их наличии, интервалы возрастания и убывания, а также координаты вершины параболы. Напишите метод, который позволит переместить параболу по заданному направлению и метод, который обращает направление ветвей.

Перегрузите оператор вывода в поток для печати всей информации об объекте

Работа №2. Массив → Двумерный массив → Журнал оценок студентов