

Лабораторная работа №2

Содержание отчета

Текст набирается 12 кеглем, интервал в тексте 1. Отступ 1,25 см.

Разделы нумеруются по порядку, выделяются из текста жирным шрифтом.

Таблицы и рисунки должны содержать номер и название. Комментарии к рисункам обязательны. Если таблица не содержит комментариев и примечаний, то комментарий к таблице обязателен.

После выполнения лабораторной работы студент должен защитить ее, пояснив процесс обработки данных, схемы алгоритмов и тексты программы, а также ответив на ряд контрольных вопросов.

- Титульный лист с указанием темы второй лабораторной работы.

Актуальное оформление титульного листа располагается по адресу:
http://guap.ru/guap/standart/titl_main.shtml

- Постановка задачи

Необходимо написать оба полных задания, которые требуется выполнить во второй лабораторной работе.

- Формализация задачи, представленная двумя разделами

Каждый раздел посвящен отдельному заданию. Формализация (от лат. forma — вид, образ) — отображение результатов мышления в точных понятиях и утверждениях. При формализации изучаемым объектам, их свойствам и отношениям ставятся в соответствие некоторые устойчивые, хорошо обозримые и отождествимые конструкции, дающие возможность выявить и зафиксировать существенные стороны объектов.

- Исходный код, представленный двумя разделами

Привести полный исходный код, который полностью выполняет требования и оба задания второй лабораторной работы.

- Результаты работы программы, представленные двумя разделами

Привести примеры работы программы для каждого задания с описанием действий пользователя и описанием того, что программа выводит на экран.

- Выводы

Сделать выводы о проделанной работе и изученном материале.

Требования к защите

При оценке преподавателем ЛР будут учитываться следующие показатели:

1. Листинг программы и удобство использования программы. То есть будет оценен стиль программирования, выделение и очищение памяти, обоснованность типов возвращаемых данных. Будет оцениваться меню пользователя - насколько реализован интерфейс для пользователя и позволяет ли он полноценно взаимодействовать с программой.
2. Ответы по листингу программы. Обоснованность написания алгоритма и исходного кода при решении задачи.
3. Ответ на теоретические вопросы.

Для защиты лабораторной работы необходимо продемонстрировать работу программы, ответить на вопросы и предоставить корректно оформленный отчет. Лабораторная работа считается успешно сданной при подписанном отчете преподавателем и студентом, и выложенном отчете в личном кабинете студента на сайте ГУАП и подтвержденным преподавателем.

Максимальное число баллов за вторую лабораторную работу - 15. Но студент может получить меньшее число баллов, в зависимости от его ответов и представленной программы.

Лабораторная работа 2. Перегрузка операторов

Важно учитывать при реализации:

- Необходимо выполнить разделение на `h` и `src` файлы для каждого класса. `h` файлы содержат определение, `src` файлы содержат реализацию. функция `main` обязана располагаться в отдельном `src` файле.
- Элемент очереди/стека/списка содержит данные и ссылку на предыдущий/следующий/предыдущий и следующий элемент. Элемент реализовать с помощью класса или структуры. В классе или структуре, реализующей элемент, необходимо поместить функции для установки и извлечения данных. То есть напрямую обращения к данным и ссылке(-ам) быть не должно.
- Необходимо работать с динамическим выделением памяти. Важно выделять и всегда очищать выделенную память.
- Необходимо написать для каждого класса: конструктор с параметрами и без параметров, конструктор копирования, деструктор, необходимо продемонстрировать применение спецификатора `explicit`.
- Требуется реализовать методы/функции с аргументами по умолчанию.
- Реализовать пользовательское меню согласно заданию. В пользовательском меню обязательно предоставить пользователю возможность взаимодействовать со всеми

настраиваемыми параметрами. То есть не должно существовать в программе числовых или буквенных констант, которые могут быть введены пользователем.

- Номер варианта соответствует номеру студента в списке. Каждому студенту необходимо выполнить оба задания.

Вариант 1

- Задание 1 Унарная операция

Создать класс «Циклическая очередь». Оператор ! проверяет ее на пустоту, оператор ++(префикс, метод) добавляет элемент в очередь, оператор --(префикс, дружественная функция) извлекает элемент из очереди, оператор ++(постфикс, дружественная функция) добавляет два элемента в очередь, оператор --(постфикс, метод) удаляет из очереди 2 элемента.

- Задание 2. Бинарная операция

Создать класс целых чисел. Определить оператор + как метод и - как дружественную функцию. Определить операторы сравнения (<, >, ==, !=) с другими экземплярами вашего класса (половину реализовать как дружественные функции, половину как методы).

Вариант 2

- Задание 1 Унарная операция

Создать класс целых чисел. Оператор --(префиксная форма, метод) уменьшает значение на 1, оператор ++(префиксная форма, метод) увеличивает значение на 1, оператор --(постфиксная форма, дружественная функция) уменьшает значение на 2, оператор ++(постфиксная форма, дружественная функция) увеличивает значение на 2.

- Задание 2. Бинарная операция

Создать класс «двусвязный линейный список», который хранит числа; размер списка вводится с клавиатуры, список заполняется случайными числами. Перегрузить операторы сравнения (<, >, ==, !=) для работы с числами (поэлементно). Половину операторов перегрузить как дружественные функции, половину как методы. Перегрузить оператор () для возвращения подсписка.

Вариант 3

- Задание 1 Унарная операция

Создать класс вещественных чисел. Оператор --(префиксная форма, метод) уменьшает значение на 0,1, оператор ++(префиксная форма, метод) увеличивает значение на 0,5, оператор --(постфиксная форма, дружественная функция) уменьшает значение на

0,2, оператор ++(постфиксная форма, дружественная функция) увеличивает значение на 2,5.

- Задание 2. Бинарная операция

Создать объект "стек", размер стека вводится с клавиатуры, стек заполняется случайными числами. Требуется перегрузить операторы для работы с числами (половину как дружественные функции, половину как методы) +, *, =, +=, -, -=.

Вариант 4

- Задание 1. Унарная операция

Создать объект типа очередь. ++(постфиксная форма, метод) добавляет элемент в очередь, -- (постфиксная форма, метод) вытаскивает элемент из очереди, ++(префиксная форма, дружественная функция) увеличивает все числа в очереди на 1, -- (префиксная форма, дружественная функция) уменьшает все числа в очереди на 1. Оператор ! проверяет очередь на пустоту.

- Задание 2. Бинарная операция

Создать класс координат. Определить оператор + как метод и - как дружественную функцию. Должно быть доступно сложение и вычитание координат с другим экземпляром вашего класса и с числом. Перегрузить оператор сравнения (поэлементное) ==, !=. Перегрузить ввод в поток и вывод из потока.

Вариант 5

- Задание 1 Унарная операция

Создать класс «Дата», который содержит число, месяц и год. Перегрузить операторы: ++ (префиксная форма, метод) увеличивает на выбор пользователя либо дату, либо месяц, либо год на 1. -- (префиксная форма, метод) аналогично уменьшает. ++(постфиксная форма, дружественная функция) увеличивает на выбор пользователя либо дату, либо месяц, либо год на пользовательское число. -- (постфиксная форма, метод) аналогично уменьшает.

- Задание 2. Бинарная операция

Создать объект типа очередь. Перегрузить оператор + как функцию член и * как дружественную функцию. + добавляет элемент в очередь, * умножает элементы в очереди на пользовательское число. Вытаскивает элемент из очереди оператор -. Очереди можно присваивать (=), проверять на равенство (поэлементное) == или !=, вводить и выводить в поток.

Вариант 6

- Задание 1 Унарная операция

Создать объект «циклический односвязный список». Перегруженные операторы: ++ (префиксная форма, метод) добавляет 1 элемент в список. ++(постфиксная форма, метод) добавляет 2 элемента в список. -- (префиксная форма, дружественная функция) сообщает пользователю, что хранится на текущей позиции, затем удаляет элемент и переходит на следующий элемент. -- (постфиксная форма, дружественная функция) сообщает пользователю, что хранится на текущей позиции и на следующей, затем удаляет текущую позицию и следующую за ней, переходит через 2 элемента.

- Задание 2 Бинарная операция

Создать класс «Габариты», который включает в себя высоту, ширину, глубину. Перегрузить все арифметические операторы для работы с числами (половину как дружественные функции, половину как методы).

Вариант 7

Создать объект "циклическая очередь". Перегрузить операции /, /=, +, --, ++ (префиксная форма и постфиксная), +=, -=, =, !=, ==, >=, <=, >, <, ввода в поток, вывода в поток, получить под-очередь (). Операции сравнения могут выполняться поэлементно с числами или с другими экземплярами класса. Половину методов необходимо реализовать как дружественные функции, половину как методы.

Вариант 8

- Задание 1. Унарная операция

Создать класс «Время», который состоит из секунд, минут и часов. ++ (префиксная форма, дружественная функция) увеличивает на выбор пользователя либо секунды, либо минуты, либо часы на единицу. -- (префиксная форма, дружественная функция) аналогично уменьшает. ++(постфиксная форма, метод) увеличивает на выбор пользователя либо секунды, либо минуты, либо часы на пользовательское значение. -- (постфиксная форма, метод) аналогично уменьшает.

- Задание 2. Бинарная операция

Создать объект очередь с перегруженными: += добавление элемента в очередь (дружественная функция), + сложение двух очередей (метод), -= для извлечения элемента из очереди (метод), - для вычитания очередей друг с другом (дружественная функция), * - для поэлементного умножения очередей друг на друга (метод), / - для деления очереди на число (метод).

Вариант 9

- Задание 1. Унарная операция

Создать класс координаты (x, y, z) с ++ и --. Операторы: ++ (постфиксная форма, метод) прибавляет 1,5 ко всем координатам, -- (постфиксная форма, метод) вычитает 5,9 от всех координат, ++ (префиксная форма, дружественная функция) складывает все три координаты, -- (префиксная форма, метод) вычитает координаты друг из друга.

- Задание 2. Бинарная операция

Создать класс стек, размер вводится с клавиатуры, заполняется стек случайными числами. Описать операторы сравнения !=, <, >, <=, >=, == для работы с другими экземплярами вашего класса, оператор () возвращает под-стек.

Вариант 10

- Задание 1. Унарная операция

Создать объект "стек" с перегруженными операторами - -, ++ и !. Размер стека вводится с клавиатуры, значения стека вводятся с клавиатуры. Перегрузить оператор ! для получения отрицательного значения. Оператор -- (префиксная форма, метод) уменьшает значение всех элементов на 1, оператор ++ (префиксная форма, метод) увеличивает значение всех элементов на 0,5, оператор - -(постфиксная форма, дружественная функция) извлекает из очереди, оператор ++ (постфиксная форма, дружественная функция) добавляет элемент в очередь.

- Задание 2. Бинарная операция

Создать класс целых чисел (long). Определить оператор + как метод (сложение с другими экземплярами класса), - как дружественную функцию, вычитание чисел, / определить как метод для работы с другими экземплярами вашего класса, * определить как дружественную функцию для работы с числами.

Вариант 11

- Задание 1. Унарная операция

Создать класс «односвязный линейный список», для которого оператор ++ (префиксная форма, метод) добавляет в начало списка новый элемент, ++ (постфиксная форма, метод) добавляет новый элемент в конец списка, -- (префиксная форма, дружественная функция) удаляет первый элемент из списка, -- (постфиксная форма, дружественная функция) удаляет последний элемент из списка.

- Задание 2. Бинарная операция

Создать класс «Время», который состоит из секунд, минут и часов. Оператор + добавляет пользовательское значение к одному из параметров (секунды, минуты, часы).

Оператор - вычитает пользовательское значение из одного из параметров (секунды, минуты, часы). Оператор * умножает все параметры (секунды, минуты, часы) на пользовательское значение. Оператор / делит все параметры на пользовательское значение.

Вариант 12

- Задание 1. Унарная операция

Создать объект стек, перегрузив ++ и --. ++ (префиксная форма, дружественная функция) добавляет элемент в стек, ++ (постфиксная форма, дружественная функция) добавляет 2 элемента в стек, -- (префиксная форма, дружественная функция) удаляет элемент из стека, -- (постфиксная форма, метод) удаляет 2 элемента из стека.

- Задание 2. Бинарная операция

Создать класс вещественных чисел (double). Определить операторы + (дружественная функция) для сложения с другим экземпляром вашего класса, - (метод) для вычитания пользовательского значения, / (метод) для деления на пользовательское значение, * (дружественная функция) для умножения на другой экземпляр вашего класса, = для присваивания значения от другого экземпляра вашего класса.

Вариант 13

- Задание 1. Унарная операция

Создать класс целых чисел (long). Определить операторы - - (префиксная форма, метод) для вычитания пользовательского значения, - - (постфиксная форма, метод) для вычитания 1, ++ (префиксная форма, дружественная функция) для прибавления пользовательского значения, ++(постфиксная форма, дружественная функция) для прибавления 1. Оператор ! возвращает отрицательное значение.

- Задание 2. Бинарная операция

Создать объект "стек", размеры стека задаются пользователем с клавиатуры, значения также вводятся с клавиатуры. Перегрузить операции половину как дружественные функции, половину как методы +, +=, -, -=, *, *=, /, /= для работы с числами.

Вариант 14

Создать объект "однонаправленный циклический список", в котором определены операции: ++ (префиксная форма, метод) для добавления элемента в конец списка, -- (постфиксная форма, дружественная функция) для удаления первого элемента из списка, = присвоение двух списков друг другу, сравнение списков ==, !=, >, <, >=, <=, [] - получение

элемента списка, () - выдать подсписок от первого до пятого элемента, перегрузить арифметические операции для работы с другим экземпляром вашего класса. Перегрузить ввод и вывод в поток.

Вариант 15

- Задание 1. Унарная операция

Создать объект "однонаправленный линейный список", в котором определены операции: ++ (префиксная форма, метод) добавляет в конец списка, -- (префиксная форма, метод) удаляет элемент из списка, ++ (постфиксная форма, дружественная функция) добавляет элемент в начало списка, -- (постфиксная форма, дружественная функция) удаляет элемент из начала списка.

- Задание 2. Бинарная операция

Создать класс «Время», который состоит из секунд, минут и часов. Определить оператор + как функцию-член и - как дружественную функцию. + увеличивает на выбор пользователя либо секунды, либо минуты, либо часы на число единиц, указанное пользователем. - аналогично уменьшает. Время можно сравнивать (<, >, ==). Перегрузить операторы половину как дружественные функции, половину как методы

Вариант 16

- Задание 1 Унарная операция

Создать класс вещественных чисел (double), перегрузив операторы: ++ (префиксная форма, метод) увеличивает число на пользовательское значение, ++ (постфиксная форма, дружественная функция) увеличивает число на случайное число, -- (префиксная форма, метод) уменьшает число на случайное число, -- (постфиксная форма, дружественная функция) уменьшает число на пользовательское значение. Оператор ! возвращает отрицательное число.

- Задание 2 Бинарная операция

Создать объект "стек" с перегруженными операциями (половину как дружественные функции, половину как методы) +, *, =, /, -, +=, *=, /=, -= для работы с другими экземплярами вашего класса. Размер стека определяется случайным образом (значение от 5 до 15), также случайным образом заполняется.

Вариант 17

- Задание 1 Унарная операция

Создать класс координаты (x, y, z). Перегрузить операторы: ! умножает все координаты на (-1), ++ (префиксная форма, дружественная функция) увеличивает все

координаты на наименьшее значение среди x, y, z; ++ (постфиксная форма, дружественная функция) увеличивает одну из координат на пользовательское число; -- (префиксная форма, метод) уменьшает все координаты на наибольшее значение среди x, y, z; -- (постфиксная форма, метод) уменьшает одну из координат на пользовательское число.

- Задание 2 Бинарная операция

Создать объект "стек" с перегруженными операциями для работы с числами (половину как дружественные функции, половину как методы) -, -=, /=, /, =, ==, <, >, !, ввод и вывод в поток. Размер стека определяется случайным образом (значение от 5 до 10), также случайным образом заполняется.

Вариант 18

- Задание 1 Унарная операция

Создать класс Время, который состоит из секунд, минут и часов. Перегрузить операторы: ++ (префиксная форма, метод) увеличивает все данные на наименьшую разницу между значениями секунд, минут и часов; ++ (постфиксная форма, метод) увеличивает на выбор пользователя один из параметров (секунды, или минуты, или часы) на наибольшее значение среди секунд, минут и часов; -- (префиксная форма, метод) уменьшает все данные на наибольшую разницу между значениями секунд, минут и часов; -- (постфиксная форма, метод) уменьшает на выбор пользователя один из параметров (секунды, или минуты, или часы) на пользовательское значение.

- Задание 2 Бинарная операция

Создать объект типа стек. Размер стека определяется пользователем, заполняется случайным образом. Перегрузить оператор + как метод для работы с числами и * как дружественную функцию для работы с числами. + добавляет элемент в стек, * умножает верхушку стека на параметр от пользователя. Стеки можно присваивать, проверять на равенство == или !=, вводить и выводить в поток.

Вариант 19

- Задание 1 Унарная операция

Создать объект типа стек, размер стека задается пользователем, данные вводятся с клавиатуры. Оператор ! проверяет стек на пустоту. Оператор ++ (префиксная форма, дружественная функция) увеличивает все значения в стеке на максимальное значение, которое в нем хранится, ++ (постфиксная форма, дружественная функция) увеличивает все значения в стеке на минимальное значение, которое в нем хранится, -- (префиксная форма, метод) уменьшает все значения в стеке на максимальное значение, которое в нем

хранится, -- (постфиксная форма, метод) уменьшает все значения в стеке на минимальное значение, которое в нем хранится.

- Задание 2 Бинарная операция

Создать класс вещественных чисел. Определить оператор - как метод и + как дружественную функцию для работы с другими экземплярами вашего класса. Перегрузить операторы сравнения <, >, != и == для работы с другими экземплярами вашего класса.

Вариант 20

- Задание 1 Унарная операция

Создать объект "очередь". Размер очереди определяется случайным образом (значение от 3 до 10), также случайным образом заполняется. Оператор ! умножает все данные в очереди на (-1). Оператор ++ (префиксная форма, метод) увеличивает все данные очереди на пользовательское значение, ++ (постфиксная форма, метод) увеличивает все данные очереди на значение размера очереди, -- (префиксная форма, метод) уменьшает все данные очереди на пользовательское значение, -- (постфиксная форма, метод) уменьшает все данные очереди на значение (размер очереди * 2).

- Задание 2 Бинарная операция

Создать класс «Габариты», который включает в себя высоту, ширину, глубину. Перегрузить все операторы сравнения для работы, как с числом, так и с другим экземпляром вашего класса (половину как дружественные функции, половину как методы).

Вариант 21

- Задание 1 Унарная операция

Создать класс «Габариты», который включает в себя высоту, ширину, глубину. Перегрузить операторы: ++ (префиксная форма, дружественная функция) увеличивает все данные на максимальное значение среди высоты, ширины, глубины, ++ (постфиксная форма, дружественная функция) увеличивает на выбор пользователя один из параметров (высота, ширина, глубина) на пользовательское значение, -- (постфиксная форма, дружественная функция) уменьшает все данные на минимальное значение среди высоты, ширины, глубины, -- (префиксная форма, метод) уменьшает на выбор пользователя один из параметров (высота, ширина, глубина) на пользовательское значение.

- Задание 2 Бинарная операция

Создать циклический односвязный список, размер его задается пользователем с клавиатуры, значения задаются случайным образом (максимальное значение 100).

Перегрузить операторы +, -, /, * для работы с числами, и операторы == и != для работы с другими экземплярами вашего класса.

Вариант 22

- Задание 1 Унарная операция

Создать очередь, размер очереди задается пользователем с клавиатуры, значения задаются случайным образом. Перегрузить операторы: ++ (префиксная форма, метод) увеличивает размер очереди на пользовательское значение, данные вносятся случайным образом, ++ (постфиксная форма, дружественная функция) увеличивает все данные очереди на минимальное число, которое в ней хранится, -- (префиксная форма, метод) уменьшает размер очереди на пользовательское значение, -- (постфиксная форма, дружественная функция) уменьшает все данные очереди на максимальное число, которое в ней хранится.

- Задание 2 Бинарная операция

Создать класс целых чисел (long). Перегрузить все операторы сравнения (половину как дружественные функции, половину как методы) для работы, как с числами, так и с другими экземплярами класса.

Вариант 23

- Задание 1 Унарная операция

Создать класс координаты (x, y, z). Перегрузить операторы: ++ (префиксная форма, метод) увеличивает на выбор пользователя x или y или z на пользовательское число, ++(постфиксная форма, дружественная функция) увеличивает на выбор пользователя x или y или z на максимальное число среди x, y, z; -- (префиксная форма, дружественная функция) уменьшает на выбор пользователя x или y или z на пользовательское число, -- (постфиксная форма, метод) увеличивает на выбор пользователя x или y или z на минимальное число среди x, y, z.

- Задание 2 Бинарная операция

Создать класс циклическая очередь, размер которой задается случайным образом (от 5 до 30). Необходимо перегрузить арифметические операторы +, -, /, * для работы с другими экземплярами вашего класса; и +=, -=, /=, *= для работы с числами.