

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное учреждение
высшего профессионального образования
"Казанский (Приволжский) федеральный университет"
Институт физики



УТВЕРЖДАЮ

Проректор
по образовательной деятельности КФУ
Проф. Таюрский Д.А.

"__" _____ 20__ г.

Программа дисциплины
Программирование на C++ Б1.В.ДВ.4

Специальность: 03.05.01 - Астрономия

Специализация: не предусмотрено

Квалификация выпускника: Астроном. Преподаватель

Форма обучения: очное

Язык обучения: русский

Автор(ы):

Фахртдинов Р.Х.

Рецензент(ы):

Мулькаманов Г.Д.

СОГЛАСОВАНО:

Заведующий(ая) кафедрой: Шерстюков О. Н.

Протокол заседания кафедры No ____ от "____" _____ 201__ г

Учебно-методическая комиссия Института физики:

Протокол заседания УМК No ____ от "____" _____ 201__ г

Регистрационный No

Казань
2018

Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) Фахртдинов Р.Х.

1. Цели освоения дисциплины

Целью освоения дисциплины (модуля) является освоение синтаксиса и стандартной библиотеки языка C++, изучение и получения навыков объектно-ориентированного программирования с использованием языка C++

2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел " Б1.В.ДВ.4 Дисциплины (модули)" основной образовательной программы 03.05.01 Астрономия и относится к дисциплинам по выбору. Осваивается на 2 курсе, 4 семестр.

Данная учебная дисциплина включена в раздел "С2.ДВ.1". Осваивается на втором курсе (2 семестр).

Федеральный государственный образовательный стандарт по направлению подготовки "Астрономия", предусматривает изучение дисциплины "Программирование на языке C+". Дисциплина занимает место в системе курсов, ориентированных на изучение принципов построения и разработки информационных систем, применяющихся для построения систем сбора, хранения, передачи и численной обработки данных.

Кроме того, преподавание этого курса обеспечивает студентов современными знаниями, о новых перспективных средствах разработки программного обеспечения, начинающих применяться в современном обществе, потребности которого в обработке информации неуклонно возрастают. В процессе изучения данного курса осуществляется формирование знаний об наиболее востребованной на данный момент парадигмы программирования, начинающей применяться астрономических и физических исследованиях, в том числе и для эффективной обработки экспериментальной информации и численного физического моделирования явлений и систем. Также осуществляется формирование умения на практике применять полученные теоретические и практические знания при выполнении курсовых и дипломных работ, ориентироваться в справочной, учебно-методической литературе, приобретения навыков разработки программ с применением парадигмы объектно ориентированного программирования.

Для освоения данной дисциплины необходимы знания, полученные обучающимися на первом курсе обучения, в частности, они должны иметь общее представление об алгоритмах, структурном и объектно-ориентированном программировании.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-1 (профессиональные компетенции)	способностью ориентироваться в базовых астрономических и физико-математических теориях и применять их в научных исследованиях
ПК-3 (профессиональные компетенции)	способностью и готовностью работать в коллективе исследователей, в том числе в качестве руководителя подразделения или исследовательской группы, формировать цели работы, принимать решения, вести обучение сотрудников
ПК-4 (профессиональные компетенции)	владением навыками самостоятельной работы, самостоятельно оценивать результаты своей деятельности

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-5 (профессиональные компетенции)	способностью и готовностью самостоятельно или в составе группы вести научный поиск
ПСК-1.4	знанием и использованием наземных и космических методов получения астрофизической информации об астрономических объектах и явлениях
ПСК-1.5	умением использовать приобретенные знания для астрофизических исследований

В результате освоения дисциплины студент:

1. должен знать:

принципы объектно-ориентированного программирования, особенности и принципы функционального программирования, синтаксис языка C++, способы применения стандартной библиотек C++ и STL

2. должен уметь:

разрабатывать программы с применением языка программирования C++

3. должен владеть:

практическими навыками составления программ на языке C++, навыки применения среды разработки

4. должен продемонстрировать способность и готовность:

применять знания, навыки, способность и готовность: к практическому применению полученных знаний.

4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 5 зачетных(ые) единиц(ы) 180 часа(ов).

Форма промежуточного контроля дисциплины экзамен в 4 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Тема 1. Обзор парадигм программирования. Перспективы развития языка C++	4		2	0	2	Письменное домашнее задание

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
2.	Тема 2. Тема 2. Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных	4		4	0	4	Письменное домашнее задание
3.	Тема 3. Тема 3. Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Потоки C++. Работа с файлами	4		6	2	6	Письменное домашнее задание
4.	Тема 4. Тема 4. Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.	4		4	2	6	Письменное домашнее задание
5.	Тема 5. Тема 5. Препроцессор. Команды препроцессора.	4		4	2	6	Письменное домашнее задание
6.	Тема 6. Тема 6. Обработка ошибок в C++	4		4	0	6	Письменное домашнее задание
7.	Тема 7. Тема 7. Шаблоны классов и функций. Применение шаблонов классов и функций Библиотека STL.	4		4	2	6	Письменное домашнее задание

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
	Тема . Итоговая форма контроля	4		0	0	0	Экзамен
	Итого			28	8	36	

4.2 Содержание дисциплины

Тема 1. Тема 1. Обзор парадигм программирования. Перспективы развития языка C++ лекционное занятие (2 часа(ов)):

Обзор парадигм программирования. Перспективы развития языка C++

лабораторная работа (2 часа(ов)):

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1. Написать функцию, которая записывает данные о сканере из приведенной структуры в требуемую позицию в бинарном файле. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Запись может осуществляться в любую позицию, причем если между вводимой записью и последней (или началом файла) имеются пропуски, они заполняются нулями. Написать функцию, которая "уплотняет" описанный выше бинарный файл путем удаления из него записей, содержащих все нули. Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее шести записей и осуществляющий его уплотнение. Все необходимые данные для функций должны передаваться им в качестве пара метров. Использование глобальных переменных в функциях не допускается. 6 Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1. Написать функцию, которая динамически выделяет память под массив структур (не меньше шести элементов), заполняет его данными в режиме диалога и записывает массив в бинарный файл. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Написать функцию, которая запрашивает данные о сканере в режиме диалога и замещает записи в бинарном файле по заданному номеру. Обязательный параметр - номер замещаемой записи. Функция должна возвращать нулевое значение, если запись прошла успешно, и -1 в противном случае. Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий вставку новых данных о сканере. Все необходимые данные для функций должны передаваться им в качестве пара метров. Использование глобальных переменных в функциях не допускается. 7 Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1. Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Написать функцию, которая вводит данные о сканере с клавиатуры в структуру типа `scan_info`, и если данные об этом сканере отсутствуют в файле, помещает содержимое структуры в конец файла; в противном случае выдает соответствующее сообщение. Привести пример программы, создающей файл с данными о сканерах (данные вводятся из текстового файла) и 6-8 записей и дополняющей файл записями о 2-3 сканерах, вводимых с клавиатуры. Все необходимые данные для функций должны передаваться им в качестве пара метров. Использование глобальных переменных в функциях не допускается.

Тема 2. Тема 2. Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных

лекционное занятие (4 часа(ов)):

Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных

лабораторная работа (4 часа(ов)):

Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных

Тема 3. Тема 3. Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Поток C++. Работа с файлами

лекционное занятие (6 часа(ов)):

Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Поток C++. Работа с файлами

практическое занятие (2 часа(ов)):

Общие сведения - использование массивов и указателей;

лабораторная работа (6 часа(ов)):

Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Поток C++. Работа с файлами

Тема 4. Тема 4. Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.

лекционное занятие (4 часа(ов)):

Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.

практическое занятие (2 часа(ов)):

Расширение языка - использование классов, конструкторы, деструкторы; Расширение языка - использование потоков, файловые потоки;

лабораторная работа (6 часа(ов)):

Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.

Тема 5. Тема 5. Препроцессор. Команды препроцессора.

лекционное занятие (4 часа(ов)):

Препроцессор. Команды препроцессора.

практическое занятие (2 часа(ов)):

Принципы объектно-ориентированного программирования использование применение инкапсуляции, наследования и полиморфизма; Библиотека STL- использование контейнеров и итераторов

лабораторная работа (6 часа(ов)):

Выполнение практического задания с использованием команд препроцессора в программах на языке C++ Примеры заданий: 1. Гармоническое среднее двух чисел получается путем вычисления среднего от инверсий этих чисел с последующим инвертированием результата. Воспользуйтесь директивой `#define` для определения функционального макроса, который выполняет эту операцию. Напишите простую программу для тестирования этого макроса. 2. В полярной системе координат вектор описывается модулем и углом с осью x в направлении против часовой стрелки. В прямоугольной системе координат тот же вектор описывается составляющими x и y . Напишите программу, которая считывает значения модуля и угла (в градусах) вектора, а затем отображает составляющие x и y . Воспользуйтесь следующими уравнениями: $x = r \cos A$ $y = r \sin A$ Для выполнения преобразования применяйте функцию, которая принимает структуру, содержащую полярные координаты, и возвращает структуру, содержащую прямоугольные координаты (или, если хотите, выберите вариант с указателями на эти структуры) 3. Библиотека ANSI содержит функцию `clock()` со следующим описанием: `#include <time.h> clock_t clock (void);` Здесь `clock_t` это тип данных, определенный в файле `time.h`. Функция возвращает процессорное время, выраженное в единицах, которые зависят от реализации. (Если процессорное время недоступно или не может быть представлено, функция возвращает -1.) Однако в файле `time.h` также определена константа `CLOCKS_PER_SEC`, которая представляет количество единиц процессорного времени в секунде. Следовательно, в результате деления разницы между двумя значениями, возвращаемыми `clock()`, на константу `CLOCKS_PER_SEC` получается количество секунд, прошедшее между двумя вызовами функции. Приведение значений к типу `double` до операции деления позволит получить результат в долях секунды. Напишите функцию, которая принимает аргумент типа `double`, представляющий промежуток времени, а затем выполняет цикл до истечения указанного периода времени. Напишите простую программу для тестирования этой функции. 4. Напишите функцию, которая в качестве аргумента принимает имя массива элементов `int`, размер массива и значение, представляющее количество выборок. Функция должна случайным образом выбирать из массива указанное количество элементов и выводить их значения. Ни один элемент массива не должен выбираться более одного раза. (Это эмулирует выбор чисел в лотерею или членов жюри.) Если в данной реализации доступна функция `time()` или подобная ей функция, то для вывода данных воспользуйтесь функцией `srand()`, чтобы инициализировать генератор случайных чисел `rand()`. Напишите простую программу для тестирования этой функции.

Тема 6. Тема 6. Обработка ошибок в C++

лекционное занятие (4 часа(ов)):

Обработка ошибок в C++

лабораторная работа (6 часа(ов)):

Выполнение практического задания по обработке ошибок в C++ Примеры заданий: 1 Дан код: `#include <iostream> #include<vector> using namespace std; const size_t SIZE = 5; int get_element(const vector<int> & data, int index) { if (index < 0 || index > data.size()) { throw range_error("Incorrect index"); } return data[index]; } int main() { vector<int> data(SIZE); for (int i = 0; i < SIZE; ++i) { data[i] = i; } int index; cin >> index; cout << get_element(data,index); }` При каком значении `index` возникнет исключение Добавьте обработку исключения в программу Найдите ошибку в программе и исправьте ее 2 Напишите функцию-"калькулятор", которая проводит умножение, сложение, вычитание и деление чисел Функция должна принимать три аргумента - `int calc(int a, char c, int b)` с определяет арифметическую операцию, и входит во множество `{'+','-','/','*'}` Калькулятор работает только с целыми числами от -1000 до 1000 создайте класс `math_exception`, отнаследуйте его от `exception` из `stdlib` функция должна бросать такое исключение, если: а) происходит деление на 0 б) результат вычисления или аргумент выходит за границы диапазона [-1000;1000] в коде функции `main` добавьте вызов `calc` и обработку возможного исключения

Тема 7. Тема 7. Шаблоны классов и функций. Применение шаблонов классов и функций Библиотека STL.

лекционное занятие (4 часа(ов)):

Шаблоны классов и функций. Применение шаблонов классов и функций Библиотека STL.

практическое занятие (2 часа(ов)):

Выполнение практического задания по применению шаблонов в программах на языке C++
Примеры заданий: Все задачи выполняются с использованием контейнеров STL 1. Два выпуклых многоугольника заданы на плоскости перечислением координат вершин в порядке обхода границы. Определить площади многоугольников и проверить, лежит ли один из них строго внутри другого. 2. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга, ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной. 3. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы круг, ограниченный окружностью, проходящей через эти три точки, содержал все точки второго множества и имел минимальную площадь. 4. Даны два множества точек на плоскости. Выбрать четыре различные точки первого множества так, чтобы квадрат с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 5. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы треугольник с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 6. Даны два множества точек на плоскости. Найти радиус и центр окружности, проходящей через n ($n \geq 3$) точек первого множества и содержащей строго внутри себя равное число точек первого и второго множеств. 7. Даны два множества точек на плоскости. Из первого множества выбрать три различные точки так, чтобы треугольник с вершинами в этих точках содержал (строго внутри себя) равное количество точек первого и второго множеств. 8. На плоскости заданы множество точек M и круг. Выбрать из M две различные точки так, чтобы наименьшим образом различались количества точек в круге, лежащие по разные стороны от прямой, проходящей через эти точки. 9. Дано $3n$ точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество n треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга.

лабораторная работа (6 часа(ов)):

Выполнение практического задания по применению шаблонов в программах на языке C++
Примеры заданий: Все задачи выполняются с использованием контейнеров STL 1. Два выпуклых многоугольника заданы на плоскости перечислением координат вершин в порядке обхода границы. Определить площади многоугольников и проверить, лежит ли один из них строго внутри другого. 2. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга, ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной. 3. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы круг, ограниченный окружностью, проходящей через эти три точки, содержал все точки второго множества и имел минимальную площадь. 4. Даны два множества точек на плоскости. Выбрать четыре различные точки первого множества так, чтобы квадрат с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 5. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы треугольник с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 6. Даны два множества точек на плоскости. Найти радиус и центр окружности, проходящей через n ($n \geq 3$) точек первого множества и содержащей строго внутри себя равное число точек первого и второго множеств. 7. Даны два множества точек на плоскости. Из первого множества выбрать три различные точки так, чтобы треугольник с вершинами в этих точках содержал (строго внутри себя) равное количество точек первого и второго множеств. 8. На плоскости заданы множество точек M и круг. Выбрать из M две различные точки так, чтобы наименьшим образом различались количества точек в круге, лежащие по разные стороны от прямой, проходящей через эти точки. 9. Дано $3n$ точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество n треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга.

4.3 Структура и содержание самостоятельной работы дисциплины (модуля)

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Тема 1. Обзор парадигм программирования. Перспективы развития языка C++	4		подготовка домашнего задания	8	Письменное домашнее задание
2.	Тема 2. Тема 2. Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных	4		подготовка домашнего задания	10	Письменное домашнее задание
3.	Тема 3. Тема 3. Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Потоки C++. Работа с файлами	4		подготовка домашнего задания	10	Письменное домашнее задание
4.	Тема 4. Тема 4. Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.	4		подготовка домашнего задания	10	Письменное домашнее задание
5.	Тема 5. Тема 5. Препроцессор. Команды препроцессора.	4		подготовка домашнего задания	10	Письменное домашнее задание
6.	Тема 6. Тема 6. Обработка ошибок в C++	4		подготовка домашнего задания	12	Письменное домашнее задание

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
7.	Тема 7. Тема 7. Шаблоны классов и функций. Применение шаблонов классов и функций Библиотека STL.	4		подготовка домашнего задания	12	Письменное домашнее задание
	Итого				72	

5. Образовательные технологии, включая интерактивные формы обучения

Освоение дисциплины предполагает использование традиционных образовательных технологий: лекционных и практических занятий в компьютерном классе с привлечением мультимедийных технологий при объяснении материала.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

Тема 1. Тема 1. Обзор парадигм программирования. Перспективы развития языка C++

Письменное домашнее задание , примерные вопросы:

Выполнение практического задания по основам языка Примеры заданий: 1 Для хранения данных о планшетных сканерах описать структуру вида: `struct scan_info { char model[25]; // наименование модели int price: // цена double x_size: // горизонтальный размер области сканирования double y_size: // вертикальный размер области сканирования int opt: // оптическое разрешение int grey: // число градаций серого }`; Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Написать функцию, которая извлекает из этого файла данные о сканере в структуру типа `scan_info`. Обязательный параметр - номер требуемой записи. Функция должна возвращать нулевое значение, если чтение прошло успешно, и -1 в противном случае. Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) - 6-8 записей и выводящей на дисплей данные о запрошенной записи. Все необходимые данные для функций должны передаваться им в качестве пара метров. Использование глобальных переменных в функциях не допускается.

Тема 2. Тема 2. Базовые сведения. Типы и описания. Выражения и операторы. Структура программы. Сложные типы данных: указатели, структуры, объединения. Операции над сложными типами данных

Письменное домашнее задание , примерные вопросы:

Примеры заданий: **ОБЩАЯ ЧАСТЬ ЗАДАНИЯ** Реализовать односвязный (для вариантов с 1 по 4)/двусвязный (для вариантов с 5 по 7) списки с хранимым целочисленным значением. Для этих списков реализовать функции: создания списка с первым элементом в качестве аргумента и без него; добавления и удаления элемента с начала с конца и после *i*-го элемента; удаления списка с удалением всех его элементов; Вывод всего списка и значения указанного элемента. Односвязные списки 1. В дополнение к общей части создать функцию копирования списка. Копия исходного списка не должна содержать каких-либо указателей на элементы исходного. 2. В дополнение к общей части создать функцию сортировки списка произвольным методом. 3. В дополнение к общей части создать функцию выделения подсписка из исходного по условию делимости на 2 4. В дополнение к общей части создать функцию объединения списков. присоединение должно быть возможным с начала и с конца исходного списка, а также поочередную вставку(первый элемент первого списка, первый элемент второго списка, второй элемент первого, второй элемент второго и т.д.) Двусвязные списки 5. В дополнение к общей части создать функцию копирования списка. Копия исходного списка не должна содержать каких-либо указателей на элементы исходного. 6. В дополнение к общей части создать функцию сортировки списка произвольным методом. 7. В дополнение к общей части создать функцию выделения подсписка из исходного по условию делимости на 2

Тема 3. Тема 3. Расширение языка Ссылки. Классы. Конструкторы и деструкторы. Использование классов для объектно-ориентированного программирования. Поток С++. Работа с файлами

Письменное домашнее задание , примерные вопросы:

Примеры заданий: 1. Построить систему классов, для описания геометрических фигур (круга, квадрата, треугольника). Предусмотреть методы для сдвига, вращения и изменения размеров фигур. Продемонстрировать возможности соответствующих объектов. 2. Построить описание класса для представления комплексных чисел. Обеспечить операции сложения, вычитания и умножения. Продемонстрировать возможности соответствующих объектов 3. Составить описание класса для объектов-векторов, описываемых координатами концов в трёхмерном пространстве. Обеспечить операции сложения, вычитания векторов, скалярного произведения, расчета длины и косинуса угла между векторами. Продемонстрировать возможности соответствующих объектов 4. Составить описание класса многочленов от одной переменной. Предусмотреть описания операции вычисления значений многочленов для конкретного аргумента, операций сложения и вычитания многочленов и вывода на экран. Продемонстрировать возможности соответствующих объектов 5. Создать описание класса, обеспечивающего представление квадратных матриц до размера 5x5. Реализовать операции перемножения матриц, матрицы на вектор и вектора на матрицу, а также подсчета определителя. Продемонстрировать возможности соответствующих объектов 6. Написать класс-черный ящик для аппроксимации функций. Должен реализовывать аппроксимацию методом наименьших квадратов полиномами 1,2 и 3-го порядков. Продемонстрировать возможности соответствующих объектов. 7. Описать класс "студенческая группа". Предусмотреть поиск студента по фамилии, дате рождения, добавление и удаление студентов в группу, подсчёт статистических данных в группе (по оценкам)

Тема 4. Тема 4. Принципы объектно-ориентированного программирования. Инкапсуляция. Атрибуты членов классов: private, protected, public. Друзья класса. Наследование. Полиморфизм. Виртуальные функции. Перегрузка операций и функций.

Письменное домашнее задание , примерные вопросы:

Примеры заданий: 1. Создать класс CFile, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс CMyDataFile - файл, содержащий в себе данные не которого определенного типа MyData, а также заголовок, облегчающий доступ к этому файлу. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов. 2. Создать класс CPoint - точка. На его основе создать классы CcoloredPoint и CLine. На основе класса CLine создать класс CColoredLine и класс CPolyLine - многоугольник. Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета. Написать демонстрационную программу, в которой будет использоваться список объектов этих классов в динамической памяти. 3. Создать абстрактный класс CVehicle. На его основе реализовать классы CPlane, CCar и CShip. Классы должны иметь возможность задавать и получать координаты, параметры средств передвижения (цена, скорость, год выпуска). Для самолета должна быть определена высота, для самолета и корабля - количество пассажиров. Для корабля - порт приписки. Написать программу, создающую список объектов этих классов в динамической памяти. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Тема 5. Тема 5. Препроцессор. Команды препроцессора.

Письменное домашнее задание , примерные вопросы:

Примеры заданий: 1. Гармоническое среднее двух чисел получается путем вычисления среднего от инверсий этих чисел с последующим инвертированием результата. Воспользуйтесь директивой #define для определения функционального макроса, который выполняет эту операцию. Напишите простую программу для тестирования этого макроса. 2. В полярной системе координат вектор описывается модулем и углом с осью x в направлении против часовой стрелки. В прямоугольной системе координат тот же вектор описывается составляющими x и y. Напишите программу, которая считывает значения модуля и угла (в градусах) вектора, а затем отображает составляющие x и y. Воспользуйтесь следующими уравнениями: $x = r \cos A$ $y = r \sin A$ Для выполнения преобразования применяйте функцию, которая принимает структуру, содержащую полярные координаты, и возвращает структуру, содержащую прямоугольные координаты (или, если хотите, выберите вариант с указателями на эти структуры) 3. Библиотека ANSI содержит функцию clock () со следующим описанием: #include <time.h> clock_t clock (void); Здесь clock_t это тип данных, определенный в файле time. h. Функция возвращает процессорное время, выраженное в единицах, которые зависят от реализации. (Если процессорное время недоступно или не может быть представлено, функция возвращает -1.) Однако в файле time. h также определена константа CLOCKS_PER_SEC, которая представляет количество единиц процессорного времени в секунде. Следовательно, в результате деления разницы между двумя значениями, возвращаемыми clock() , на константу CLOCKS_PER_SEC получается количество секунд, прошедшее между двумя вызовами функции. Приведение значений к типу double до операции деления позволит получить результат в долях секунды. Напишите функцию, которая принимает аргумент типа double, представляющий промежуток времени, а затем выполняет цикл до истечения указанного периода времени. Напишите простую программу для тестирования этой функции. 4. Напишите функцию, которая в качестве аргумента принимает имя массива элементов int, размер массива и значение, представляющее количество выборок. Функция должна случайным образом выбирать из массива указанное количество элементов и выводить их значения. Ни один элемент массива не должен выбираться более одного раза. (Это эмулирует выбор чисел в лотерею или членов жюри.) Если в данной реализации доступна функция time () или подобная ей функция, то для вывода данных воспользуйтесь функцией srand (), чтобы инициализировать генератор случайных чисел rand (). Напишите простую программу для тестирования этой функции.

Тема 6. Тема 6. Обработка ошибок в C++

Письменное домашнее задание , примерные вопросы:

Примеры заданий: 1 Дан код: `#include <iostream> #include<vector> using namespace std; const size_t SIZE = 5; int get_element(const vector<int> & data, int index) { if (index < 0 || index > data.size()) { throw range_error("Incorrect index"); } return data[index]; } int main() { vector<int> data(SIZE); for (int i = 0; i < SIZE; ++i) { data[i] = i; } int index; cin >> index; cout << get_element(data,index); }` При каком значении index возникнет исключение Добавьте обработку исключения в программу Найдите ошибку в программе и исправьте ее 2 Напишите функцию-"калькулятор", которая проводит умножение, сложение, вычитание и деление чисел Функция должна принимать три аргумента - int calc(int a, char c, int b) с определяет арифметическую операцию, и входит во множество {'+', '-', '/', '*'} Калькулятор работает только с целыми числами от -1000 до 1000 создайте класс math_exception, отнаследуйте его от exception из stdlib функция должна бросать такое исключение, если: а) происходит деление на 0 б) результат вычисления или аргумент выходит за границы диапазона [-1000;1000] в коде функции main добавьте вызов calc и обработку возможного исключения

Тема 7. Шаблоны классов и функций. Применение шаблонов классов и функций Библиотека STL.

Письменное домашнее задание , примерные вопросы:

Примеры заданий: Все задачи выполняются с использованием контейнеров STL 1. Два выпуклых многоугольника заданы на плоскости перечислением координат вершин в порядке обхода границы. Определить площади многоугольников и проверить, лежит ли один из них строго внутри другого. 2. Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга, ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной. 3. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы круг, ограниченный окружностью, проходящей через эти три точки, содержал все точки второго множества и имел минимальную площадь. 4. Даны два множества точек на плоскости. Выбрать четыре различные точки первого множества так, чтобы квадрат с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 5. Даны два множества точек на плоскости. Выбрать три различные точки первого множества так, чтобы треугольник с вершинами в этих точках покрывал все точки второго множества и имел минимальную площадь. 6. Даны два множества точек на плоскости. Найти радиус и центр окружности, проходящей через n ($n \geq 3$) точек первого множества и содержащей строго внутри себя равное число точек первого и второго множеств. 7. Даны два множества точек на плоскости. Из первого множества выбрать три различные точки так, чтобы треугольник с вершинами в этих точках содержал (строго внутри себя) равное количество точек первого и второго множеств. 8. На плоскости заданы множество точек M и круг. Выбрать из M две различные точки так, чтобы наименьшим образом различались количества точек в круге, лежащие по разные стороны от прямой, проходящей через эти точки. 9. Дано $3n$ точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество n треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга.

Тема . Итоговая форма контроля

Примерные вопросы к экзамену:

ТЕМАТИЧЕСКИЙ ПЛАН ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО КУРСУ

Практическое занятие ♦1. Освоение среды разработки.

Практическое занятие ♦2. Написание простейшей программы. Структура программы, запись программы на языке C++. Операторы. Типы данных. Элементарные операции по вводу и выводу. Выполнение практического задания по основам языка. Написание программы

Практическое занятие ♦3. Выполнение практического задания по основам языка. Сдача программы.

Практическое занятие ♦4. Массивы. Указатели. Выполнение практического задания по массивам и динамическим переменным. Написание программы.

Практическое занятие ♦5. Классы. Выполнение практического задания по массивам и динамическим переменным. Написание программы.

Практическое занятие ♦6. Структуры. Применение структур. Члены структуры. Операции со структурами. Списки. Выполнение практического задания по структурам и спискам C++. Написание программы.

Практическое занятие ♦9. Выполнение практического задания по структурам и спискам C++. Сдача программы.

Практическое занятие ♦10. Классы. Инкапсуляция. Закрытые и открытые члены классов. Применение конструкторов и деструкторов. Применение полиморфизма и наследования. Выполнение практического задания по применению конструкторов и деструкторов в классах C++. Написание программы.

Практическое занятие ♦11. Выполнение практического задания по применению конструкторов и деструкторов в классах C++.

Практическое занятие ♦12. Выполнение практического задания по применению конструкторов и деструкторов в классах C++. Сдача программы. Контрольная работа по теме "Принципы объектно ориентированного программирования"

Практическое занятие ♦13. Применение принципов объектно-ориентированного программирования. Применение механизмов наследования и полиморфизма. Виртуальные операции. Перегрузка операций. Выполнение практического задания по принципам объектно ориентированного программирования.

Практическое занятие ♦14. Выполнение практического задания по принципам объектно ориентированного программирования.

Практическое занятие ♦15. Выполнение практического задания по принципам объектно ориентированного программирования. Сдача программы. Контрольная работа по теме Принципы объектно-ориентированного программирования.

Практическое занятие ♦16. Шаблоны классов и функций. Построение шаблонов. Применение шаблонов. Выполнение практического задания по применению шаблонов в программах на языке C++.

Практическое занятие ♦17. Выполнение практического задания по применению шаблонов в программах на языке C++. Контрольная работа по теме "Применение шаблонов в C++"

ВОПРОСЫ К ЗАЧЕТУ:

1. Простые типы данных.
2. Целочисленные типы данных.
3. Вещественные типы данных.
4. Специальные типы данных.
5. Сложные (агрегатные) типы данных.
6. Массивы. Связь между массивами и указателями.
7. Структуры и классы.
8. Указатели. Операции с указателями.
9. Ссылки. Применение ссылок.
10. Структура программы.
11. Работа с функциями в C++. Объявления и описания.
12. Команды препроцессора.
13. Условная компиляция.
14. Правила написания классов, структур, объединений.
15. Конструкторы и деструкторы.
16. Принцип инкапсуляции.
17. Модификаторы доступа к членам класса.
18. Шаблонные функции.
19. Принцип наследования.
20. Принцип полиморфизма. Реализация принципа полиморфизма в C++.
21. Уточнение поведения при наследовании классов.
22. Перегруженные функции.

23. Перегрузка операций.
24. Виртуальные функции-члены классов. Раннее и позднее связывание.
25. Потоки. Стандартные потоки.
26. Файловый вывод с помощью потоков.
27. Правила ввода/вывода с помощью потоков. Преимущества потоков по сравнению со средствами языка C.
28. Статические функции члены класса.
29. Манипуляторы потоков.
30. Дружественные функции.
31. Шаблонные классы.

САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ (СРС) включает следующие виды работ:

- изучение теоретического лекционного материала;
- проработка теоретического материала (конспекты лекций, основная и дополнительная литература);
- доработка текстов программ, незаконченных на практических занятиях;
- подготовка к контрольным работам и к сдаче зачета.

7.1. Основная литература:

1. Серебряков В.А. Теория и реализация языков программирования. "Физматлит", 2012. - 236 с. // http://e.lanbook.com/books/element.php?pl1_id=5294
2. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы. 2-е изд. "Бином. Лаборатория знаний", 2012. - 223 с. // http://e.lanbook.com/books/element.php?pl1_id=8765

7.2. Дополнительная литература:

1. Акулич, Иван Львович. Математическое программирование в примерах и задачах : учебное пособие . 3-е изд. ЛАНЬ, 2011. - 352 с. // http://e.lanbook.com/books/element.php?pl1_id=2027
2. Хохлов Д.Г. Методы программирования на языке C: практикум. Ч.1. "Бином. Лаборатория знаний", 2014. - 336 с. // http://e.lanbook.com/books/element.php?pl1_id=50555
3. Хохлов Д.Г. Методы программирования на языке C: практикум. Ч.1. "Бином. Лаборатория знаний", 2014. - 377 с. // http://e.lanbook.com/books/element.php?pl1_id=50555

7.3. Интернет-ресурсы:

- Литвиненко, Н. А. Технология программирования на C++. Начальный курс - - <http://znanium.com/bookread.php?book=349908>
- Немцова Т. И. Программирование на языке высокого уровня. Программирование на языке C++ - - <http://znanium.com/bookread.php?book=244875>
- Побегайло, А. П. C/C++ для студента - <http://znanium.com/bookread.php?book=489154>
- Полубенцева, М. И. C/C++. Процедурное программирование - <http://znanium.com/bookread.php?book=350407>
- Хабибуллин И.Ш. Программирование на языке высокого уровня C/C++ - <http://znanium.com/bookread.php?book=356906>

8. Материально-техническое обеспечение дисциплины(модуля)

Освоение дисциплины "Программирование на C++" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

1. Лекционная аудитория с мультимедиа проектором, компьютером и экраном.
2. Компьютерный класс на 9 рабочих мест.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по специальности: 03.05.01 "Астрономия" и специализации не предусмотрено .

Автор(ы):

Фахртдинов Р.Х. _____

"__" _____ 201__ г.

Рецензент(ы):

Мулькаманов Г.Д. _____

"__" _____ 201__ г.