Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

Лабораторная работа №1.

" Классы и объекты. Инкапсуляция. "

15 вариант

Выполнил студент гр. РИС-24-2б

Иванова Елена Олеговна

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

Цель:

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Использование классов и объектов в ОО программе.

Задача: Элемент арифметической прогрессии aj вычисляется по формуле: aj=a0·r j. Поле first – дробное число, первый элемент прогрессии a0, поле second – положительное целое число, постоянное отношение r. Реализовать метод element(int j) –вычисление j-го элемента прогрессии.

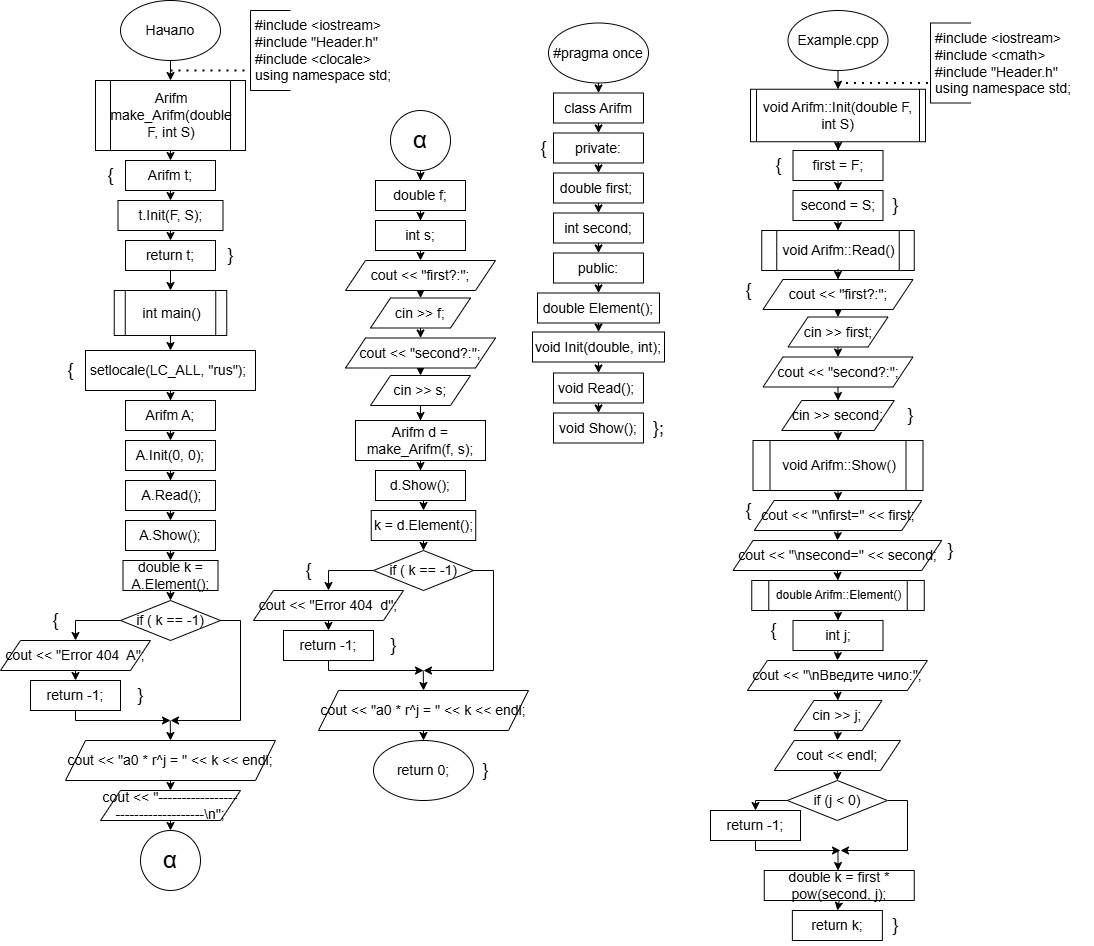
Постановка задачи:

* Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.
* Структура-пара – структура с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутствовать:
  + метод инициализации Init (метод должен контролировать значения аргументов на корректность);
  + ввод с клавиатуры Read;
  + вывод на экран Show.
* Реализовать внешнюю функцию make\_тип(), где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

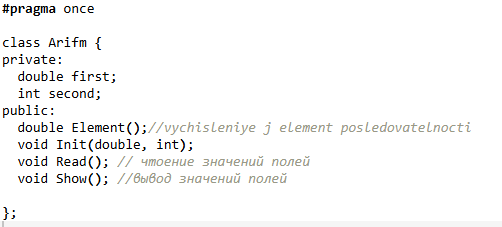
Анализ задачи.

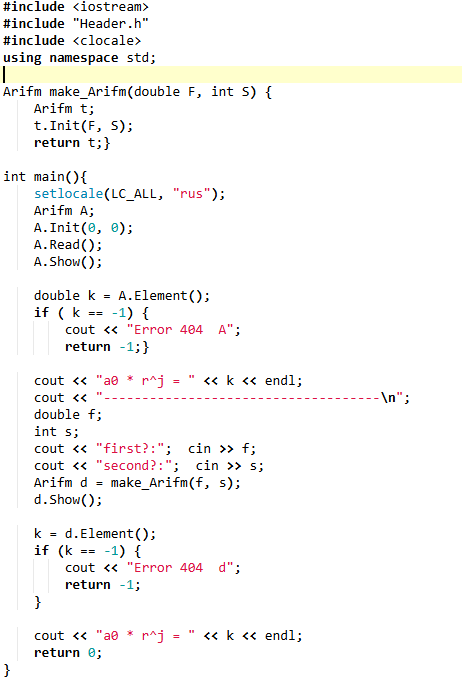
1. В main.cpp написать основной код.
2. В заголовочный файл написать основу класса.
3. В Еxample.cpp расписать методы класса.

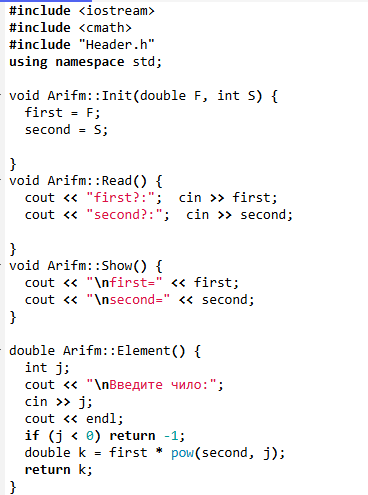
Блок схема.



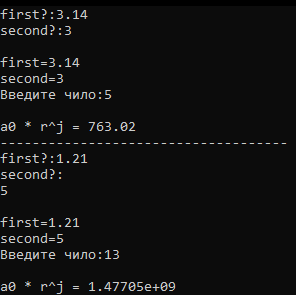
Код.







Вывод.



Ответы на контрольные вопросы:

1. Что такое класс?

Класс — шаблон или чертёж для создания объектов. Он определяет состояние (поля, атрибуты) и поведение (методы, функции) объекта.

1. Что такое объект (экземпляр) класса?

Объект (экземпляр) класса — конкретная реализация класса, созданная на основе его "шаблона". Он содержит реальные данные и может выполнять методы, определённые в классе.

1. Как называются поля класса?

Поля класса называются атрибутами или свойствами.

1. Как называются функции класса?

Функции класса называются методами.

1. Для чего используются спецификаторы доступа?

Спецификаторы доступа используются для контроля видимости и изменения полей и методов класса.

1. Для чего используется спецификатор public?

public используется для объявления полей и методов класса, которые доступны везде.

1. Для чего используется спецификатор private?

private используется для строгого ограничения доступа к полям и методам класса, делая их доступными только внутри этого класса.

1. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

В с++ для class по умолчанию используется спецификатор доступа private.

1. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?

В с++ для struct по умолчанию используется спецификатор доступа public.

1. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

При проектировании интерфейса класса в C++ все методы и поля, входящие в публичный API, должны быть объявлены как public. Это ключевой принцип инкапсуляции: предоставить пользователю класса только то, что необходимо, скрыв внутреннюю реализацию.

1. Каким образом можно изменить значения атрибутов экземпляра класса?

Прямое изменение (если поле public). //Person p; p.name = "Alice";

1. Каким образом можно получить значения атрибутов экземпляра класса?

Прямой доступ (если поле public) //cout << p.name;

1. Класс описан следующим образом

struct Student {

string name; int group; ……… };

Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

s->name;

1. Класс описан следующим образом

struct Student {

string name; int group; …….. };

Объект класса определен следующим образом Student s; Как можно обратиться к полю name объекта s?

s.name;

1. Класс описан следующим образом

class Student {

string name; int group; ……. };

Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

В public класса вписываем функцию.

void setName(const string& newName) {

name = newName; }

Обращаемся к объекту как s->setName("Анна")

1. Класс описан следующим образом

class Student { string name; int group; public: ….. };

Объект класса определен следующим образом Student s; Как можно обратиться к полю name объекта s?

В public класса вписываем функцию.

void setName(const string& newName) {

name = newName; }

Обращаемся к объекту как s.setName("Анна")

1. Класс описан следующим образом

class Student { public: char\* name; int group; ……… };

Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

s->name.