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.»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

1. Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.

2. 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:

a. метод инициализации Init (метод должен контролировать значения аргументов на корректность);

b. ввод с клавиатуры Read;

c. вывод на экран Show.

3. 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.

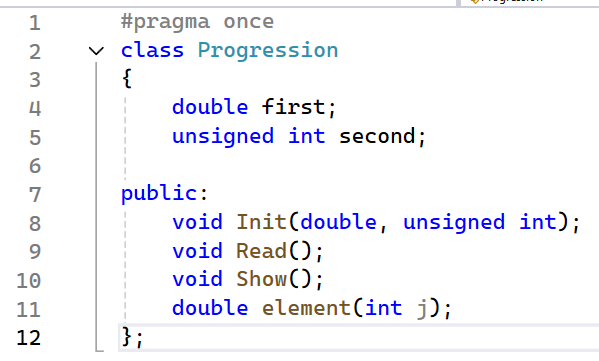
Элемент арифметической прогрессии *aj*​ вычисляется по формуле: aj=*a*0​⋅*r^j*. Поле first – дробное число, первый элемент прогрессии *a*0​, поле second – положительное целое число, постоянное отношение *r*. Реализовать метод element(int j) – вычисление j-го элемента прогрессии.

**Анализ задачи**

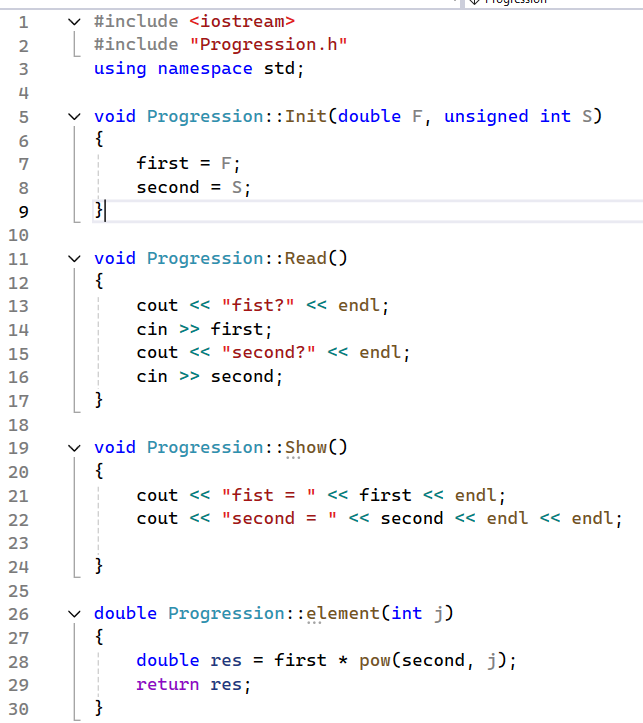
* Класс Progression содержит два поля: first (первый элемент прогрессии) и second (постоянное отношение). Эти поля используются для вычисления элементов прогрессии.
* Методы класса (Init, Read, Show, element) обеспечивают инициализацию, ввод, вывод и вычисление элементов прогрессии.
* В главной функции демонстрируется создание объектов, массивов объектов и динамических массивов, а также использование методов класса.

**Код программы**

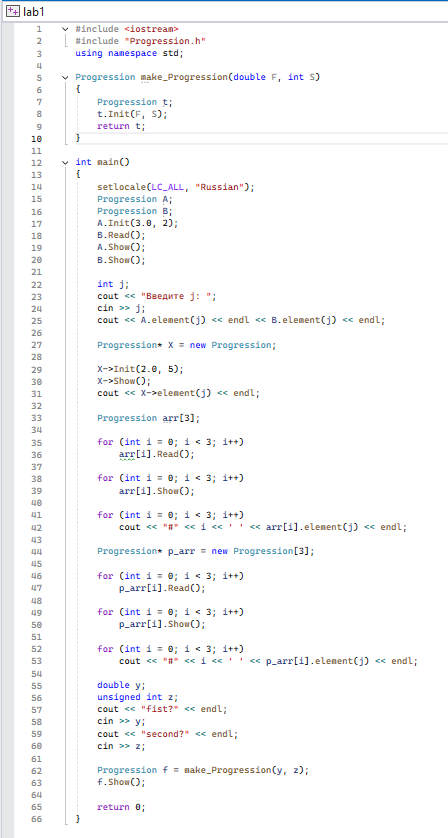
Progression.h

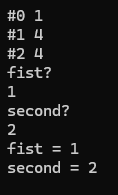
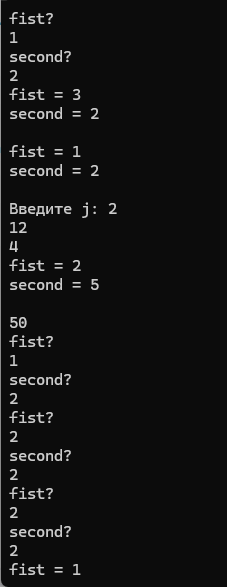
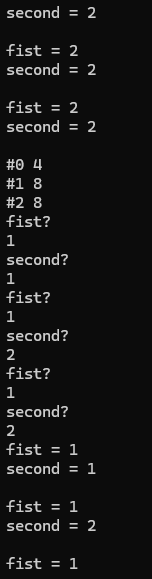


Progression.cpp



Lab1\_main.cpp





**Контрольные вопросы**

1. **Что такое класс?**  
   Класс — это пользовательский тип данных, который объединяет данные (поля) и функции (методы) для работы с этими данными.
2. **Что такое объект (экземпляр) класса?**  
   Объект (экземпляр) класса — это конкретная переменная, созданная на основе класса, которая содержит данные и может использовать методы класса.
3. **Как называются поля класса?**  
   Поля класса называются атрибутами или членами данных.
4. **Как называются функции класса?**  
   Функции класса называются методами.
5. **Для чего используются спецификаторы доступа?**  
   Спецификаторы доступа определяют уровень доступа к членам класса (полям и методам).
6. **Для чего используется спецификатор public?**  
   Спецификатор public позволяет доступ к членам класса извне, то есть из других функций и классов.
7. **Для чего используется спецификатор private?**  
   Спецификатор private ограничивает доступ к членам класса, делая их доступными только внутри самого класса.
8. **Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?**  
   По умолчанию используется спецификатор private.
9. **Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?**  
   По умолчанию используется спецификатор public.
10. **Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?**  
    При описании интерфейса класса должен использоваться спецификатор public, чтобы методы и поля были доступны извне класса.
11. **Каким образом можно изменить значения атрибутов экземпляра класса?**  
    Значения атрибутов можно изменить через методы класса или напрямую, если атрибуты имеют спецификатор доступа public.
12. **Каким образом можно получить значения атрибутов экземпляра класса?**  
    Значения атрибутов можно получить через методы класса или напрямую, если атрибуты имеют спецификатор доступа public.
13. **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;

15. **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?**

Поле name является приватным, поэтому напрямую обратиться нельзя. Нужно использовать публичный метод, если он предусмотрен.

16. **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?**

Поле name является приватным, поэтому напрямую обратиться нельзя. Нужно использовать публичный метод, если он предусмотрен.

17. **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;