Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 1

**Классы и объекты. Инкапсуляция.**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

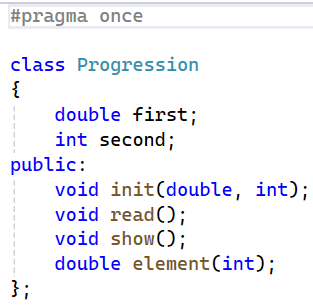
**Постановка задачи**

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:
   1. метод инициализации Init (метод должен контролировать значения аргументов на корректность);
   2. 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.

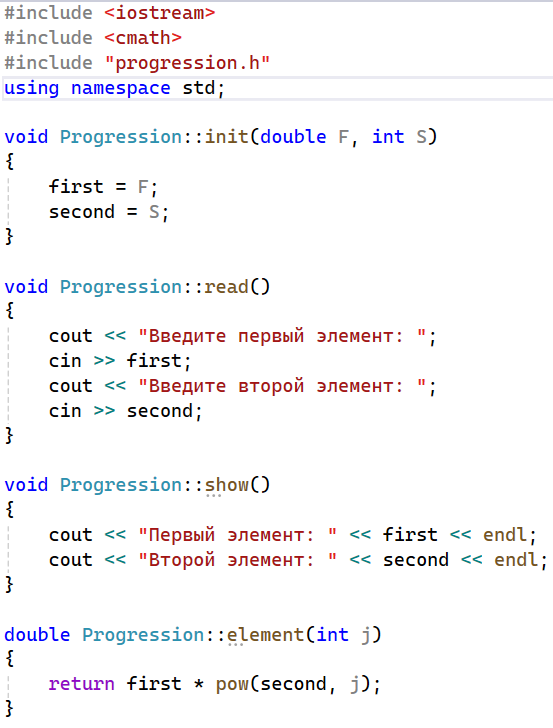
**Вариант 15**

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.

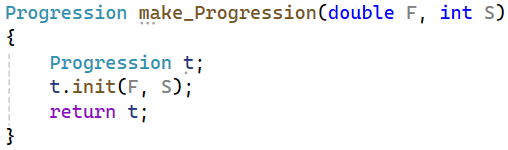
**Описание класса Progression**

****

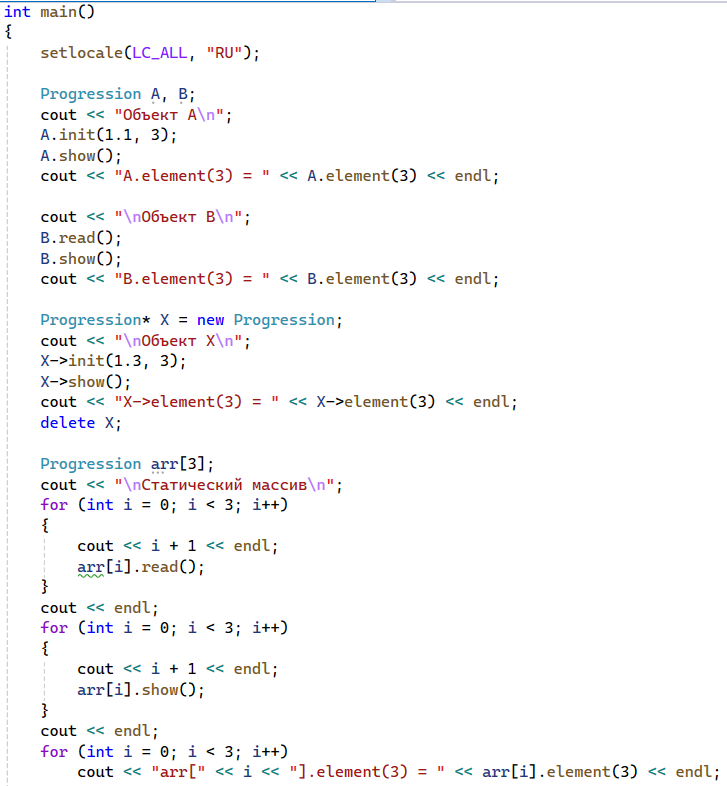
**Определение методов класса Progression**

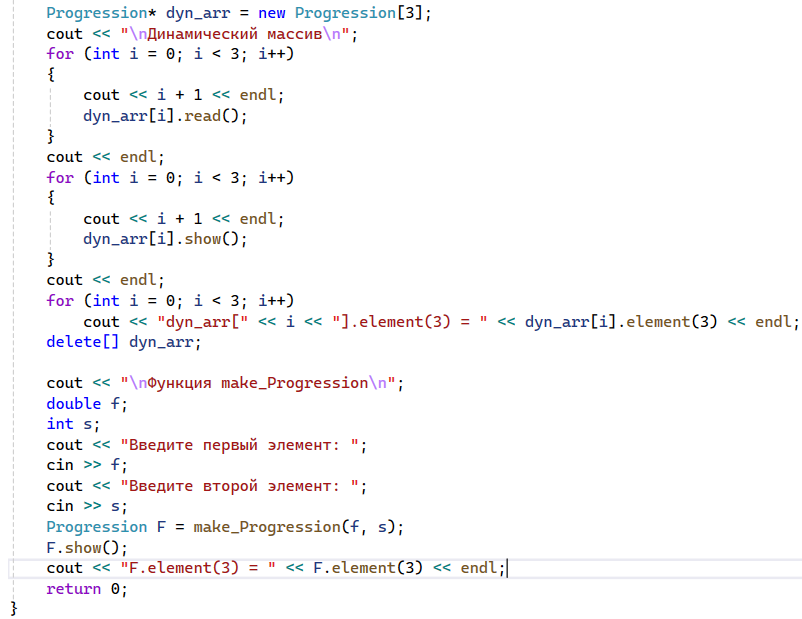
****

**Определение функции make\_Progression()**

****

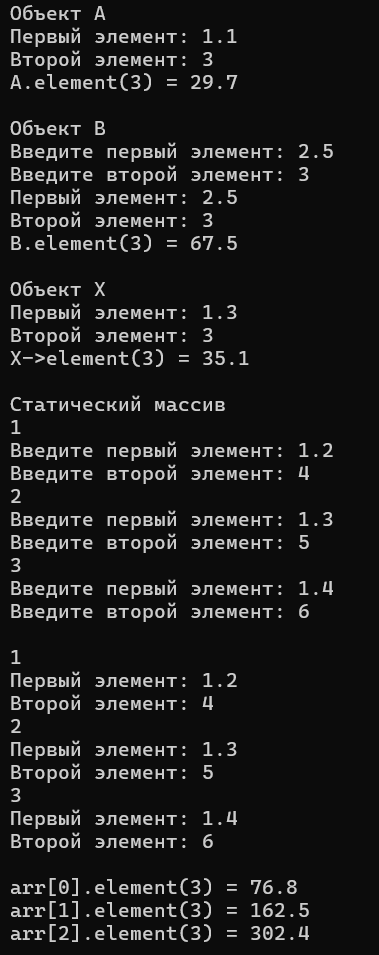
**Функция main**

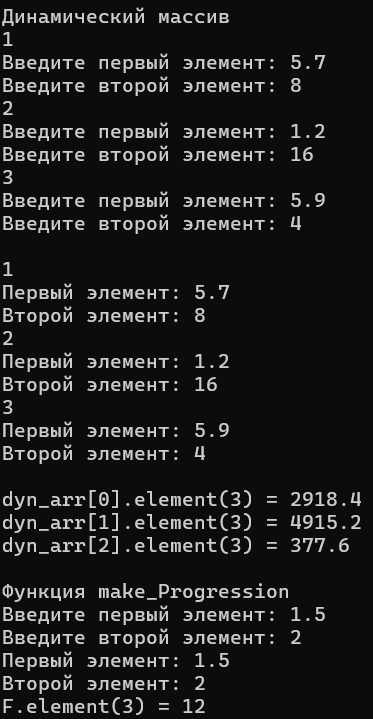
****



\

**Пример выполнения программы**

****

****

**Объяснение результатов программы**

Сначала создаются объекты A и B класса Progression. Данные в объект A заносятся с помощью метода init. Данные в объект B заносятся с помощью метода read, то есть их уже вводит пользователь. С помощью метода element вычисляется 3-й элемент последовательности у каждого из этих двух объектов класса Progression.

Далее создается указатель X на динамически выделенную память под объект класса Progression. Данные заполняются через указатель с помощью метода init. Вычисляется 3-й элемент последовательности, и динамическая память очищается.

Далее создается статический массив arr[3], содержащий в себе 3 объекта класса Progression. Каждый элемент массива arr заполняется данными пользователем с помощью метода read. Вычисляется 3-й элемент последовательности для каждогго элемента массива, где элемента являются объектами класса Progression.

Далее создается указатель на область памяти, выделенную динамически под массив объектов класса Progression длиной 3. Пользователь заносит данные в каждый элемент массива с помощью метода read. Вычисляется 3-й элемент последовательности для каждого элемента массива. Память выделенная под динамический массив очищается.

Далее пользователь вводит значения переменных double f и int s. Затем создается объект F класса Progression с помощью функции make\_Progression(), которая принимает два аргумента f и s. В функции make\_Progression() создается временный объект класса Progression, в который заносятся данные с помощью метода initб из функции фозвращается этот временный объект. Вычисляется 3-й элемент последовательности для объекта F класса Progression.

Функция main завершается.

**Ответы на контрольные вопросы**

1. **Что такое класс?**

Класс – абстрактный пользовательский тип данных, который представляет собой модель реального или придуманного объекта, содержащий данные объекта и методы для обработки этих данных.

1. **Что такое объект (экземпляр) класса?**

Объектом или экземпляром класса называется переменная класса. Класс объявляется один раз, а объектов этого класса может быть сколько угодно.

1. **Как называются данные класса?**

Данные класса называются полями класса.

1. **Как называются функции класса?**

Функции класса называются методами класса.

1. **Для чего используются спецификаторы доступа?**

Спецификаторы доступа используются для управления видимостью элементов класса в разных частях программы.

1. **Для чего используется спецификатор public?**

Спецификатор досупа public используется для построения интерфейса класса, то есть для той части класса, которая отвечает за взаимодействие пользователя с этим классом. Элементы с спецификатором public видны во всех модулях программы.

1. **Для чего используется спецификатор private?**

Спецификатор доступа private используется для тех элементов класса, которые нужны для его реализации. Элементы с этим спецификатором доступа видны только внутри этого класса.

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

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

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

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

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

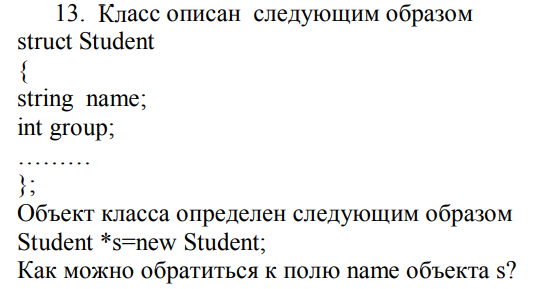
Для описания инттерфейса класса должен использоваться спецификатор доступа public, чтобы с классом можно было взаимодействовать в любом модуле программы.

1. **Каким образом можно изменить значения атрибутов экземпляра класса?**

Изменить значения атрибутов экземпляра класса можно с помощью методов-модификаторов.

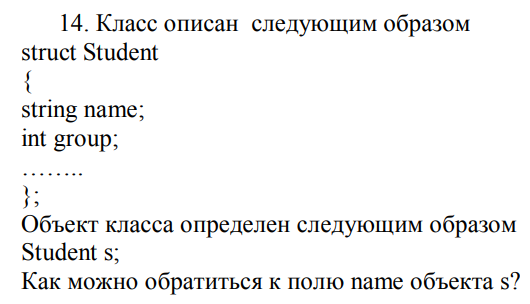
1. **Каким образом можно получить значения атрибутов экземпляра класса?**

Получить значения атрибутов экземпляра класса можно с помощью методов-селекторов.



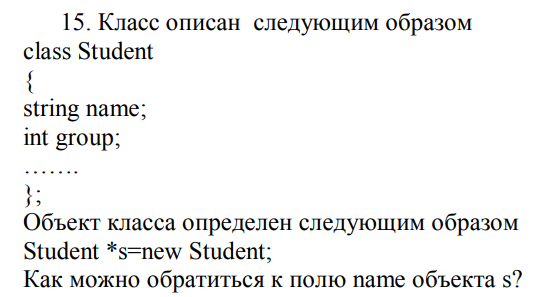
Обратиться к полю name объекта s можно следующим образом:

s->name

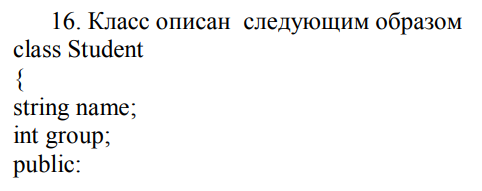


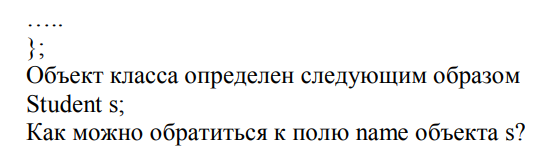
Обратиться к полю name объекта s можно следующим образом:

s.name

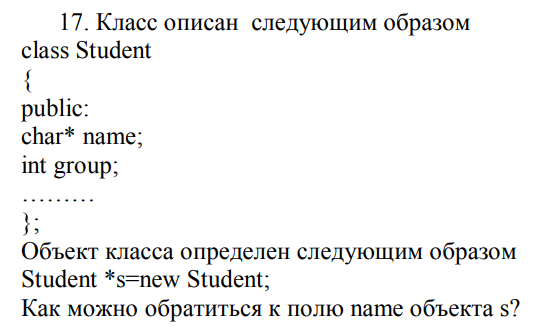


К полю name объекта s обратиться невозможно, поскольку это поле класса имеет спецификатор доступа private.





К полю name объекта s обратиться невозможно, поскольку это поле класса имеет спецификатор доступа private.



Обратиться к полю name объекта s можно следующим образом:

s->name

Ссылка на github: <https://github.com/RuntovIvan/Informatika>