Контрольная работа

Конструирование простейшего класса

При выполнении контрольной работы студент должен продемонстрировать умение сконструировать класс с заданным набором данных, создать объекты на основе класса и манипулировать ими.

Задание.

В работе требуется сконструировать класс с заданным набором свойств. Набор свойств следует взять в соответствии со своим вариантом задания. В класс также должны быть добавлены методы для просмотра и изменения значений любого из свойств объекта.

Требования к конструированию класса: доступ к свойствам — закрытый, к методам — открытый. В классе следует предусмотреть конструктор по умолчанию, конструктор с параметрами.

Действия, выполняемые программой:

1. создание объекта с помощью конструктора по умолчанию,
2. создание объекта с помощью конструктора с параметрами,
3. создание массива объектов (размерность массива 3 или 4 элемента),
4. инициализация свойств каждого объекта массива(исходные данные вводятся с клавиатуры),
5. просмотр свойств каждого объекта,
6. вычисление заданного параметра для массива объектов в соответствии с вариантом задания (выполнить с помощью глобальной функции).

Требования к структуре программного кода: программа должна иметь модульную структуру, т.е. состоять из нескольких файлов: модуля класса, состоящего из заголовочного файла и файла реализации, и главного модуля, содержащего функцию `main()`.

Представление результата.

Контрольная работа представляется в электронном виде.

Необходимо прислать:

- тексты файлов с исходным кодом программы и комментариями,
- скриншот результата работы программы,
- текстровое описание работы (постановка задачи и пояснения к программе в свободной форме),
- ответы на вопросы.

В комментариях к программе и в текстовом описании следует представить структуру класса: какие разделы имеет класс, содержание каждого раздела, для каждого члена класса указать имя, назначение и

обосновать его доступность. Указать, как в программе обозначены объекты класса и какие конструкторы были использованы при их создании.

Начисление баллов.

За контрольную работу, выполненную в соответствии всем вышеуказанным требованиям, начисляется 15 баллов. За ошибки и недочёты количество начисляемых баллов уменьшается.

Как определить свой вариант задания?

Число, образованное двумя последними цифрами номера зачётной книжки, поделить на 30. Остаток, полученный от деления, и будет номером вашего варианта.

Варианты заданий

В каждом варианте заданы свойства класса и параметр, который должен быть вычислен.

0 вариант

Название страны, площадь страны, количество жителей. Определить страну с наибольшей плотностью населения.

1 вариант

Название программы, Разработчик, Версия, Год выпуска. Определить самую новую программу.

2 вариант

Номер школы, Название школы, Специализация, Количество учащихся. Вычислить общее количество учащихся.

3 вариант

Название товара, Категория товара, Цена, Количество. Вычислить общую стоимость товара.

4 вариант

Номер банковской карты, Фамилия владельца, Год окончания действия, Остаток на счете. Определить владельца карты с минимальным остатком средств.

5 вариант

Фамилия, Количество отработанных дней, Тариф. Вычислить сумму заработной платы.

6 вариант

Марка машины, Мощность двигателя, Объем бака, Цвет кузова . Вычислить машину с самым мощным двигателем.

7 вариант

Марка монитора, Максимальное разрешение, Цена. Вычислить среднюю цену.

8 вариант

Фамилия студента, Предмет, Оценка. Вычислить количество двоек.

9 вариант

Марка принтера, Формат бумаги, Скорость печати, Цена. Определить самый дешевый принтер.

10 вариант

Название турфирмы, Маршрут, Количество оставшихся путевок. Вычислить общее количество оставшихся путевок.

11 вариант

Фамилия, Имя, Должность, Оклад. Определить самого высокооплачиваемого сотрудника.

12 вариант

Станция отправления, Станция прибытия, Время в пути. Определить маршрут с наименьшим временем в пути.

13 вариант

Фамилия спортсмена, Вид спорта, Разряд, Название спортивного клуба. Вычислить количество спортсменов, имеющих первый разряд.

14 вариант

Название книги, Автор, Год издания. Определить самое старое издание.

15 вариант

Фамилия, Отдел, Год поступления на работу, Образование. Определить средний стаж работы.

16 вариант

Фамилия студента, Название вуза, Курс, Факультет. Определить количество студентов второго курса.

17 вариант

Фамилия абонента, Продолжительность разговора в мин., Стоимость минуты разговора. Вычислить стоимость всех разговоров.

18 вариант

Фамилия, Имя, Род занятий (сотрудник, студент), Год поступления. Вычислить сотрудника, принятого на работу последним.

19 вариант

Название предмета, Преподаватель, Количество лекций, Количество лабораторных работ. Вычислить количество часов занятий по всем предметам (лекции и лабораторные работы имеют продолжительность 2 часа).

20 вариант

Фамилия, Место жительства, Год рождения. Определить средний возраст.

21 вариант

Название фирмы, Адрес, Телефон, Электронный адрес. Вычислить количество фирм, не указавших электронный адрес.

22 вариант

Фамилия, Номер договора, Стоимость заказа, Срок исполнения. Вычислить среднюю стоимость заказа.

23 вариант

Название журнала, Номер, Год выпуска. Вычислить количество журналов, выпущенных в текущем году.

24 вариант

Название группы, Факультет, Количество студентов, Количество успевающих студентов. Вычислить процент успевающих студентов по всем факультетам.

25 вариант

Марка телефона, Фирма изготовитель, Вес, Цена. Определить самый легкий телефон.

26 вариант

Название кинотеатра, Адрес, Количество мест, Средняя цена билетов. Определить кинотеатр с самым большим возможным доходом.

27 вариант

Название музея, Адрес,Год основания, Средняя посещаемость в год. Найти самый посещаемый музей.

28 вариант

Название страны, Название столицы, Количество жителей, Средняя продолжительность жизни. Вычислить общее количество жителей.

29 вариант

Название города, Количество жителей, Год основания, Количество музеев. Определить самый древний город.

Справочный материал

Класс — это пользовательский тип данных, объединяющий данные и алгоритмы для обработки этих данных. Класс моделирует группу каких-либо реальных объектов (студенты, машины), процессов (путешествия), явлений (погода).

Данные класса представлены в виде переменных и называются **свойствами.**

Алгоритмы представлены в виде функций и называются **методами.**

В классе существует разграничение доступа к его членам. Внутреннюю (закрытую) часть класса, доступную только этому классу, составляет раздел `private,` защищенная часть класса доступна классу и его наследникам — раздел `protected,` члены класса из раздела `public` доступны любым объектам без ограничения (открытая часть класса).

Пример объявления класса:

```
class Book
{
public: // открытая часть класса
Book(); // Конструктор
~Book(); // Деструктор
void SetAuthor(std::string); // Метод, устанавливающий новое
//значение свойству
std::string GetAuthor(); // Метод, позволяющий прочитать значение
//свойства. . .
private: // закрытая часть класса
string Author;
```

```
string Title;
int Year; // год издания
} ;
```

Данные для их защиты от воздействий извне помещаются в раздел `private.` Для доступа к таким данным используются методы, которые берут на себя контроль за корректностью использования данных сторонними объектами. Объединение в классе данных и методов с целью защиты данных называется **инкапсуляция.**

Для хранения текстовых данных в примере использовался стандартный тип `string`, объявленный в стандартной библиотеке языка C++. Он удобнее и надежнее для использования, чем массив символов в языке C, т.к. не надо заботиться о распределении памяти, кроме того, для типа `string` определена операция конкатенации строк +.

В класс можно включить несколько функций с одинаковыми именами, но различающихся списками параметров. Такая возможность основана на свойстве языка C++ **перегрузка функций**, разрешающем иметь в программе в пределах одной области видимости несколько функций с одинаковыми именами, но различными списками параметров. Такие функции называются перегруженными. Каждая из перегруженных функции решает свою задачу — в зависимости от переданных параметров. Компилятор различает перегруженные функции по списку параметров. Перегрузить можно глобальную функцию или функцию-член класса. В данной работе перегруженными являются конструкторы.

Объект создается по шаблону, который дает класс, при этом используется специальный метод — **конструктор**. Имя конструктора совпадает с именем класса, он помещается в раздел `public`, для него не указывается тип возвращаемого значения. Конструктор может иметь параметры. Если в класс не включен какой-нибудь конструктор, то компилятор добавит в него конструктор по умолчанию.

Конструктор по умолчанию не имеет параметров и создает объект с неинициализированными свойствами:

```
Book::Book()
{

}
```

или всегда с одним и тем же набором значений:

```
Book::Book()
{
Author = “Noname”;
Title = “Noname”;
Year = 0;
}
```

Конструктор с параметрами создает объект с заранее определенным набором свойств:

```
Book::Book(std::string Auth, std:: Ttl, int
y)
{
Author = Auth;
Title = Ttl;
Year = y;
}
```

Кроме конструктора, в составе класса есть еще один служебный метод — деструктор. **Деструктор** выполняет разрушение объекта, он не имеет параметров, находится в разделе `public`, не имеет типа возвращаемого значения, а имя отличается от имени конструктора одним символом: знаком ~ (тильда) в начале.

```
Book::~Book()
{
// Деструктор по умолчанию
}
```

Явный вызов деструктора вставлять в программу не требуется — он вызывается автоматически. Если для данных класса не выделяется динамическая память, деструктор можно не включать в класс — он будет добавлен автоматически компилятором.

В описании конструктора и любого другого метода класса (в реализации) используется оператор разрешения области видимости `::`. Этот оператор позволяет включить идентификаторы в заданное пространство имен `namespace`. Для функций-членов класса пространством имен будет класс. Если не использовать оператор `::`, получится глобальная функция, не связанная с классом.

Примеры создания объектов

```
Book B1; // Создание объекта с помощью конструктора по
//умолчанию
Book B2(“Pushkin”,”Evgeny Onegin”,2003);
//Создание объекта с помощью конструктора с параметрами
Book B3[3]; // Создание массива объектов с помощью
//конструктора  по умолчанию
```

Для доступа к свойствам и методам в функциях-членах данного класса используется только имя свойства или метода

```
Book::Book(std::string Auth, std::Ttl, int y)
{
Author = Auth;
. . .
}
```

Также возможно обратиться к члену того же класса с помощью указателя на текущий объект `this`

```
Book::Book(std::string Auth, std::Ttl, int y)
```

```
{
this->Author = Auth;
. . .
}
```

Для доступа сторонних объектов к свойствам и методам объекта какого-либо класса используется имя объекта и операция точка (.)

```
P1.SetAuthor(”Gogol”);
```

Сторонние объекты также могут обращаться к свойствам и методам какого-либо объекта с помощью указателя на объект и операции стрелка (→):

```
Book* ptr = &P1; // инициализация указателя на объект
ptr->GetAuthor(); // обращение к методу объекта через
//указатель
```

Пример простейшей реализации методов класса Book:

```
void Book::SetAuthor(std::string Auth)
{
Author = Auth;
}

std::string Book::GetAuthor()
{
return Author;
}
```

Работа с переменными типа string.

В языке С строки задаются как массив символов с нуль-символом в конце массива (`chat c[10]`) и имеются функции :

`strcpy(c1,c2)` – копирование строк;
`strcat(c1,c2)` – объединение строк;
`strcmp(c1,c2)` – сравнение строк;
и другие.

Используя эти функции, необходимо проверять, достаточно ли выделено места для строки, куда будет записана результирующая строка.

В языке С++ имеется класс string, он освобождает программиста от заботы о выделении памяти. В этом классе определены операции присваивания (=), конкатенации (+), сравнения (= =, <, >,....) и другие. Это позволяет записывать действия со строками как с числовыми переменными.

```
string s1, s2,s3; //конструктор создаёт три объекта string:
s1,s2,s3
s1="new"; //s1 присваивается значение new
s2=s1;    //строка s1 копируется в s2
s3=s1+s2; // строки s1 и s2 объединяются и записываются в строку
s3
```

`if (s1<s2) cout<<s1;` //при сравнении строк меньшей считается та, у которой код первого несовпадающего символа меньше.

Для использования класса необходимо подключить заголовочный файл `<string>`

Более подробные сведения о классах вы найдёте в лекциях и книгах, указанных в списке литературы.

Вопросы к контрольной работе.

1. Что такое класс в объектно-ориентированном программировании?
2. Какую структуру имеет модуль в C++?
3. Какими средствами осуществляется консольный ввод данных в языке Си, C++?
4. Какие свойства (принципы) объектно-ориентированного программирования вы знаете?
5. Сконструируйте простейший класс с конструктором по умолчанию и конструктором с параметрами. Покажите, как с помощью этих конструкторов можно создать объекты.