**Лаба №1**

**Конструктор** - выполняется при создании нового объекта этого класса и используется для инициализации атрибутов ( и настроек) объекта и выполнения других необходимых настроек.

**Деструктор** — Основная задача деструктора - это освобождение ресурсов, которые были выделены объекту класса во время его создания. Например, если объект класса выделял динамическую память, то деструктор должен освободить ее и вернуть в свободное пуле.

**Set и get** - это методы доступа к приватным членам класса в C++. Set используется для установки значения приватного члена класса, а get - для получения значения приватного члена.

**Наследование** в C++ - это механизм, позволяющий создавать новые классы на основе уже существующих. При этом создаваемый класс наследует свойства и методы своего "родительского" класса

**Инкапсуляция** в C++ - это механизм, который позволяет объединить данные и методы, работающие с этими данными, в одном классе и скрыть их от других частей программы. Он обеспечивает доступ к данным только через определенные методы класса, называемые геттерами и сеттерами, что позволяет контролировать изменение этих данных.

**Полиморфизм** в C++ - это свойство, позволяющее объекту использовать один и тот же интерфейс по-разному. Это означает, что можно создавать различные реализации одного и того же метода, которые будут использоваться в зависимости от контекста.

**Пространство имен *–*** это просто некая объявляемая область, необходимая для того, чтобы избежать конфликтов имен идентификаторов. Чтобы пространство имен стало видимым, используем инструкцию

**Манипулятор setw**

Манипуляторы — это особые операции, используемые совместно с операцией вставки << для того, чтобы видоизменять вывод, который делает программа.

**Манипулятор endl**

**В приведенном примере используется манипулятор endl.**

Это слово означает вставку в символьный поток символа окончания строки, поэтому весь последующий текст будет печататься с новой строки.

1) ***абстракция данных***;

2) ***инкапсуляция*** – в классах объединяются данные и методы функции для работы с ними, так как лишь через методы возможен доступ к скрытым данным класса;

3) ***наследование*** – в производных классах наследуются члены базового класса;

4) ***полиморфизм*** – возможность использования одних и тех же методов для работы с различными объектами базового и порожденных им классов.

*Спецификатор* ***public –*** *открывает**доступ к функциями данным класса из других частей программы.*

*Спецификатор* ***protected*** *необходим при наследовании классов.*

**Лаба №2**

**Лаба №3**

**Лаба №4**

**Лаба №5**

**Лаба №6**

**Лаба №7**