Лабораторная работа №12

**«Классы. Конструкторы и деструкторы, конструктор копии и статические элементы. Доступность компонентов класса: private, public. Дружественные функции (friend)»**

1. Создать класс книг с полями Автор, название книги, издательство, год издания (lab12\_1.cpp)
2. Создать класс книг с полями Автор, название книги, издательство, год издания с конструктором и деструктором (lab12\_2.cpp)
3. Классы могут содержать описание и других функций, которые становятся членами класса. Задание: изменить предыдущий пример таким образом, чтобы типизированные члены класса были скрыты, а для отображения класса существовала специальная функция (lab12\_3.cpp)
4. Идея объектного программирования состоит в том, что детальное программирование объектов осуществляется в одном месте, а их использование в другом. Задание: задать описание описание класса в заголовочном файле .h. Его детальную реализацию произвести в специальном модуле. Для этого использовать специальный элемент описания «::» - который говорит о том, что данный элемент относится к описанию класса. В вызывающей программе создать объекты указанного класса, использующие только описание класса **(lab12\_4.cpp)**
5. **Конструктор копии.** Рассмотрим уже знакомый нам пример, в котором создаются объекты книга. Определим ещё одну функцию, которая на вход будет принимать объект нашего типа. Произойдет ошибка, связанная с тем, что при вызове программы, наш объект должен скопироваться в стек. Если это не оговорено специально, то копирование производится байт за байтом. В том числе и адресные переменные будут скопированы вместе с адресом. Но при выходе из функции сработает деструктор и уничтожит содержимое адреса. Чтобы избежать возникновение этой ошибки, мы должны предпринять дополнительные действия **(lab12\_5.cpp)**
6. Копирование объекта в стек производится при помощи конструктора копии. Избежать указанную проблему можно, закрыв доступ к конструктору копии. В этом случае, мы не сможем передавать указанный объект в другую программу. В примере демонстрируется также, как можно обойти это ограничение, передав на вход адрес объекта **(lab12\_6.cpp)**
7. **Статические элементы.** Создать класс со статической переменной, которая подсчитывает общее количество создаваемых объектов **(lab12\_7.cpp)**
8. Статические функции также создаются один раз для всего класса. Они не связаны с конкретным объектом, но могут получать объекты во входном потоке **(lab12\_8.cpp)**
9. Создать класс point. В указанном классе, помимо координат включить int ID, которое должно содержать уникальный номер объекта. Этот номер должен присваиваться в любом конструкторе объекта. Для присвоения этого номера ввести статическую переменную, которая выдаёт этот номер, а затем увеличивается на 1.
10. Представить поле процент в файле **(lab10\_8.cpp)** как статическое и для разных товаров присвоить разные проценты для розничных цен и вывести те, у которых процент равен 7, а также создать конструктор и деструктор для класса goods.
11. В задании 4 из 10\_2 (про класс point) добавить конструктор со значениями по умолчанию и пустой деструктор.
12. **Функции friend.** Использование функции friend для вычисления суммы двух переменных файле **(lab12\_9.cpp)**
13. Использование friend для перегрузки левой операции (+). Важным способом использования описания **friend** является специальный способ перегрузки операторов. Обычный способ перегрузки предполагает, что перегружаемая функция относится к объекту, который находится слева от функции. Эта необходимость может возникнуть, если слева находится объект другого типа **(lab12\_10.cpp).**
14. Создать класс "Транспортное средство", в нем описать поля: наименование ТС, скорость ТС, пройденное расстояние, статические поля - максимальное расстояние MAX\_DISTANCE, максимальная скорость MAX\_SPEED, методы - получения информации о скорости, получения информации о текущем пробеге, получения информации о текущем времени эксплуатации, установки имени транспортного средства, конструктор с полями по умолчанию, деструктор. Описание класса производить в отдельном cpp файле с внешним определением функций класса.
15. Для класса complex из предыдущего занятия перегрузить (ввести дружественные) операции-функции сложения, вычитания, умножения, деления с некоторым введенным числом.