

1. Улучшите программу «Простой босс» (файл `simple_boss_2.cpp`). Добавьте класс `Final_Boss`, который будет унаследован от класса `Boss`, и добавьте к нему дополнительный член данных `m_mega_factor` с закрытым или защищенным уровнем доступа. Добавьте в класс `Final_Boss` метод `mega_attack()`, наносящий урон в 10 раз больший, чем метод `special_attack()`.
2. Улучшите программу «Абстрактное существо» (файл `abstract_creature.cpp`). Добавьте к нему дополнительный класс `Orc_Boss`, который будет унаследован от класса `Orc`. У каждого созданного объекта этого класса здоровье должно быть в 1,5 раза больше, чем у объекта базового класса. Также необходимо переопределить чистый виртуальный метод `greet()`, чтобы отображать следующее сообщение: «The orc boss growls hello.».
3. Улучшите игру «Блэкджек» (файл `blackjack.cpp`), добавив проверку ошибок. Перед началом очередного раунда программа должна проверять, достаточно ли карт в колоде; если нет, колода должна быть вновь наполнена (до 8 колод) и перемешана. Также программа должна перестать запрашивать у игрока, желает ли он взять ещё карту, если у него уже выпал блэкджек.
4. Напишите однокарточную версию игры «Война» с одной колодой, структура раунда в которой такова: все игроки тянут по одной карте, а выигрывает тот, у кого номинал карты оказался наибольшим.
5. Улучшите игру «Блэкджек» (`exercision3.cpp`), добавив возможность игрокам делать ставки. Программа должна следить за капиталом каждого игрока и выводить из игрового стола тех, у кого закончились деньги.
6. Создайте игру «Робот», в которой игрок сможет менять своё местонахождение на игровом поле, перемещаясь каждый раз в одно из мест, ближайших к данному. Игрок может ходить только вверх, вниз, влево, вправо, но не по диагонали. Размер игрового поля может варьироваться от 2x2 до 10x10 в зависимости от очередного запуска программы.
7. Создайте абстрактный класс `Figure` с методами расчета периметра и площади, а также методом вывода информации о фигуре на экран. Создайте производные классы:

- 1) Прямоугольник;
- 2) Круг;
- 3) Треугольник;

Каждый класс должен иметь свой собственный методы вычисления периметра и площади. Создать массив фигур из n элементов и вывести на экран полную информацию (в первую очередь периметр и площадь) о фигурах.

8. Создайте класс Customer со следующими свойствами:

- 1) Имя;
- 2) Отчество;
- 3) Фамилия;
- 4) Адрес;
- 5) Номер кредитной карты;
- 6) Номер банковского счета.

В классе должны быть реализованы следующие методы:

- 1) Конструктор;
- 2) Сеттеры значений свойств;
- 3) Геттеры значений свойств;
- 4) Перегруженный оператор << для отображения информации о клиенте.

Создайте обычный массив объектов данного класса без инициализации. Затем используйте сеттеры для инициализации всех свойств каждого объекта в массиве. Отобразите список покупателей в алфавитном порядке (используйте любой собственный алгоритм сортировки) и список покупателей, номера кредитных карт которых находятся в заданном диапазоне.