

Итоговый тест №4.

Задание 1.

Создайте перечисление `levelsName`. Оно будет содержать шесть элементов – “названия” этажей. Вам необходимо предложить пользователю “покататься” на лифте.

Названия для этажей:

- Паркинг(Подпись при выборе данного этажа: Вы в паркинге!)
- Банк(Подпись при выборе данного этажа: Здесь находится Креди Агриколь Банк.)
- Страховая компания(Подпись при выборе данного этажа: Страховая компания UNICA.)
- Фондовая биржа(Подпись при выборе данного этажа: Фондовая биржа.)
- Выставочный зал(Подпись при выборе данного этажа: Выставочный зал и зал для презентаций.)
- Ресторан(Подпись при выборе данного этажа: Ресторан Sky.)

Пример результата выполнения программы:

Выберите номер этажа (0,1,2,3,4,5,6): 10

Ошибка!

Выйти из лифта – нажмите 0.

Выбрать другой этаж – нажмите 1:1

Выберите номер этажа (0,1,2,3,4,5,6): 4

Четвертый этаж!

Выставочный зал и зал для презентаций.

Выйти из лифта – нажмите 0.

Выбрать другой этаж – нажмите 1:0

Задание 2.

При разработке игры мы решили, что в ней должны быть монстры, потому что всем нравится сражаться с монстрами. Объявите структуру, которая представляет вашего монстра. Монстр может быть разным: `ogre`, `goblin`, `skeleton`, `orc` и `troll`. Если ваш компилятор поддерживает C++11, то используйте классы `enum`, если нет — обычные перечисления.

Каждый монстр также должен иметь имя (используйте `std::string`) и количество здоровья, которое отображает, сколько урона он может получить, прежде чем умрет. Напишите функцию `printMonster()`, которая выведет все члены структуры. Объявите монстров типа `goblin` и `orc`, инициализируйте их, используя список инициализаторов, и передайте в функцию `printMonster()`.

В данной программе вам необходимо реализовать выбор пользователем ваших монстров.

Пример результата выполнения вашей программы:

Выберите монстра:

1 - ogre

2 - goblin

...

2

This Goblin is named John and has 170 health.

ПРИМЕЧАНИЕ. Обязательно используйте управляющие символы для вывода инструкций пользователю.