Содержание

[ОБЩАЯ ИНФОРМАЦИЯ 2](#_heading=h.gjdgxs)

[Тема занятия](#_heading=h.30j0zll) 2

[Цели и задачи 2](#_heading=h.1fob9te)

[Ожидаемый результат 2](#_heading=h.3znysh7)

[Структура занятия 3](#_heading=h.tyjcwt)

[ОПИСАНИЕ ЗАНЯТИЯ 4](#_heading=h.3dy6vkm)

# ОБЩАЯ ИНФОРМАЦИЯ

### Тема занятия

### Объекты в С++

**Цели и задачи:**

* Познакомиться с понятием объекта в программировании
* Научиться использовать уже созданными объектами

**По результатам занятия слушатель будет знать:**

* Что такое объект
* Какими атрибутами обладают объекты в программировании

**По результатам занятия слушатель будет уметь:**

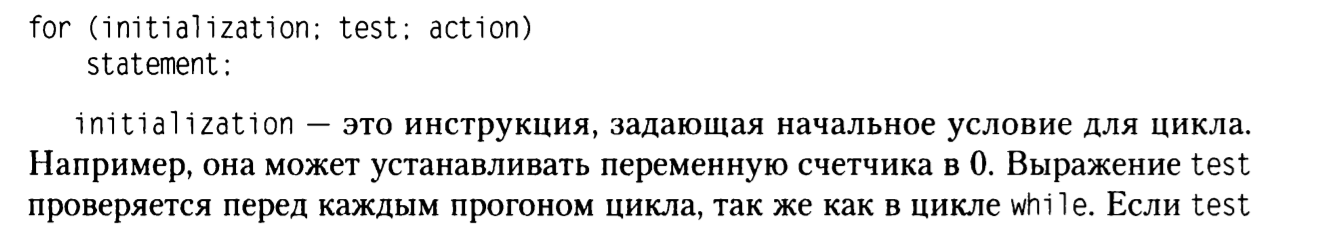
* Работать с объектами классов
* Использовать цикл for (в случаях, когда на это есть необходимость)

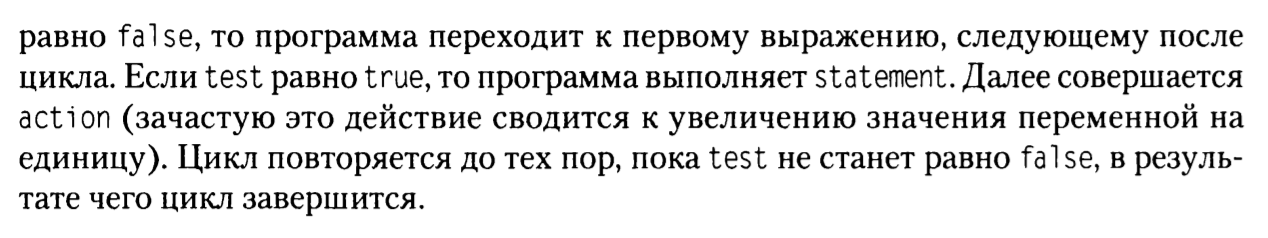
### Структура занятия

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тайминг занятия** | |  |  |  | Таблица 1 |
| № | Этапы | Что делает преподаватель | Что делает Слушатель | Время | Общее время |
| 1 | Приветственное слово преподавателя |  |  | 5 мин. | 5 мин. |
| 2 | Повторение пройденного |  |  | 5 мин. | 40 мин. |
| 3 | Теоретическая часть |  |  | 30 мин. |
| 4 | Вопросы по теоретической части |  |  | 5 мин. |
|  | *Перерыв* |  |  | 15 мин. | 15 мин. |
| 5 | Практическая часть |  |  | 40 мин. | 45 мин. |
| 6 | Рефлексия и вопросы |  |  | 5 мин. |

# ОПИСАНИЕ ЗАНЯТИЯ

**Теоретическая часть**

****



**Использование пустых инструкций в циклах for**

При создании цикла for можно использовать пустые инструкции, как в следующей строке:

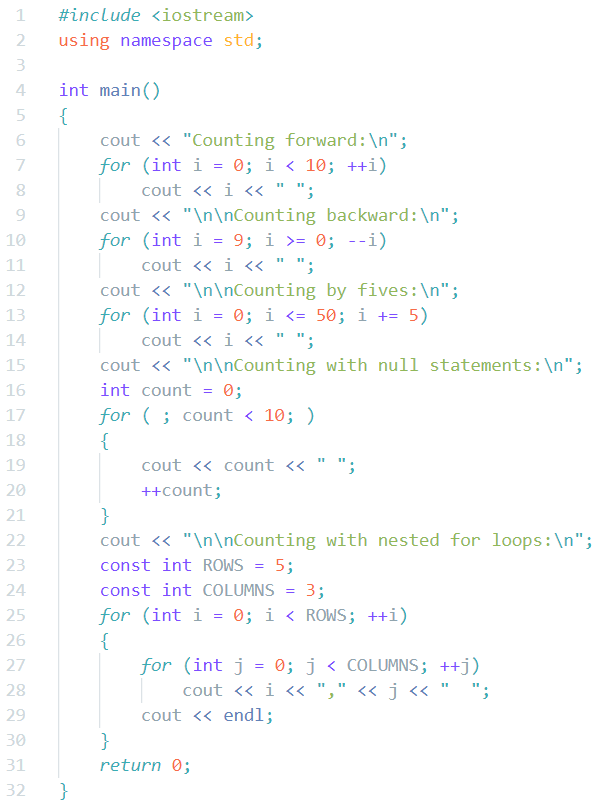


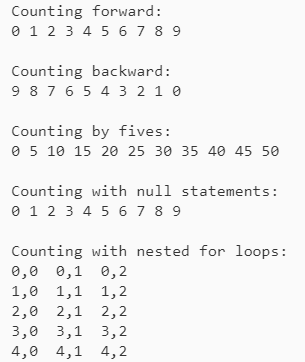
Инструкция инициализации и оператор действия здесь оставлены пустыми, это вполне допустимо. Переменная count была объявлена и инициализирована до начала цикла, а уже в теле цикла увеличивается ее значение на единицу. Хотя этот цикл на первый взгляд может показаться странным, он абсолютно корректен.

**Вложение циклов for**

Использовали ли вы вложенные циклы в своих программах?

\*Приведите пример





**Понятие об объектах**

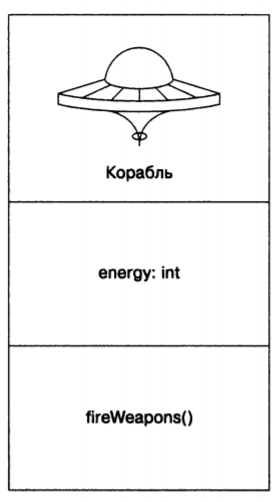
До сих пор мы говорили о том, как сохранять отдельные информационные объекты в переменных и как манипулировать этими переменными с помощью операторов и функций. Но большинство сущностей, которые вам придется представлять в компьютерной игре, - скажем, инопланетный космический корабль - являются объектами. Объекты - это цельные неоднородные сущности, обладающие определенными свойствами (например, уровнем энергии) и возможностями (допустим, могут стрелять из пушек). Зачастую при обсуждении объекта целесообразно рассматривать его свойства отдельно от его возможностей.

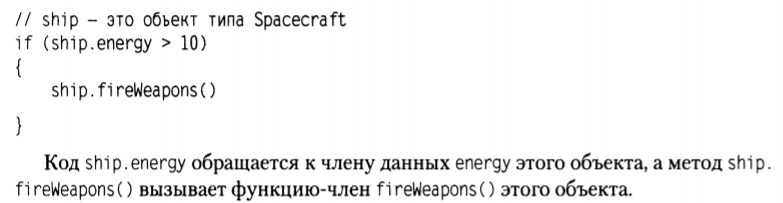
К счастью, в большинстве современных языков программирования можно работать с программными объектами (которые зачастую называются просто объектами ), в которых комбинируются данные и функции. Элемент данных в составе объекта называется -«член данных», а функция объекта называется -«функция-член» (метод).

Давайте в качестве конкретного примера рассмотрим инопланетный космический корабль. Этот объект может относиться к новому типу Spacecraft, определяемому программистом, работающим над данной игрой. В объекте такого типа могут быть член данных, соответствующий уровню энергии, а также функция-член, реализующая стрельбу из пушек. На практике уровень энергии корабля может сохраняться в члене данных energy как переменная типа int, а возможность стрельбы из пушек может быть определена в функции-члене fireWeapons( ). Все объекты, относящиеся к одному и тому же базовому типу, обладают схожей структурой. Иными словами, все объекты одного типа будут содержать схожие наборы членов данных и функций-членов. Правда, конкретные значения членов данных будут различаться от объекта к объекту. Так, если у вас есть эскадра инопланетных космических кораблей, то каждое такое судно будет обладать своим уровнем энергии.

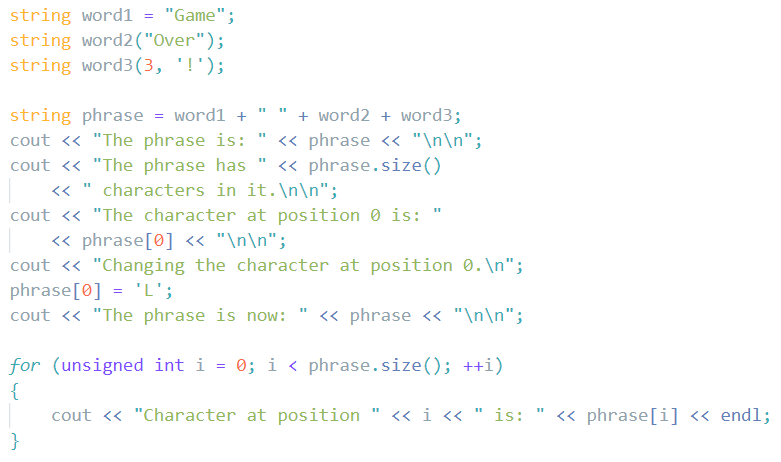
Самая классная черта объектов заключается в том, что работать с ними вы можете, не зная деталей их реализации. Так, вполне можно водить автомобиль, не разбираясь в тонкостях его конструкции. Необходимо знать лишь члены данных объекта и его функции-члены. Продолжая пример с автомобилем: чтобы управлять машиной, достаточно знать, где у нее находится руль, педаль газа и педаль тормоза.

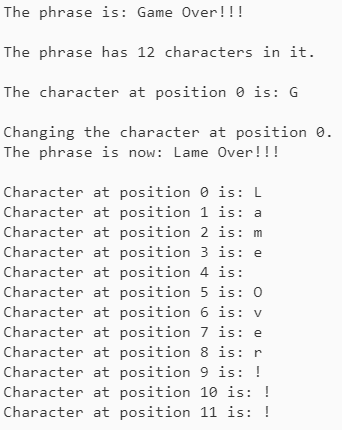
Объекты можно хранить в переменных так же, как встроенные типы. Следовательно, объект, соответствующий космическому кораблю пришельцев, можно сохранить в переменной типа Spacecraft. Можно обращаться к членам данных и функциям-членам с помощью оператора доступа к члену (. ). Этот оператор ставится после имени той переменной, которая соответствует объекту.

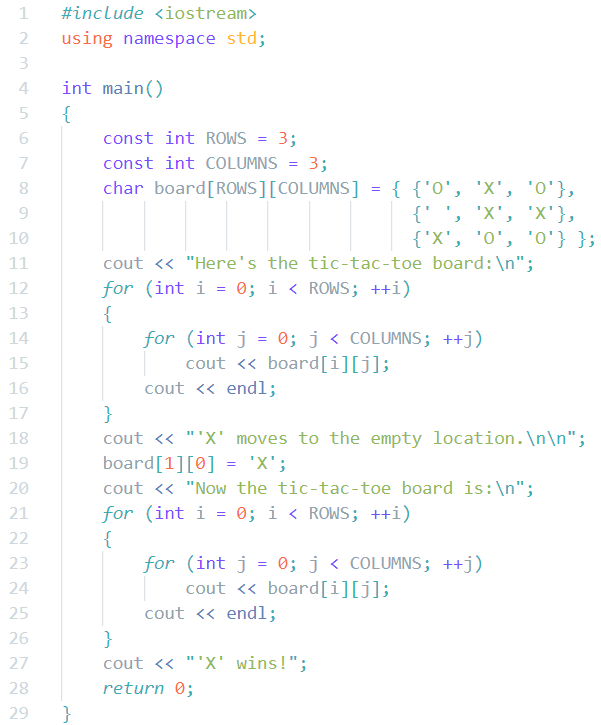


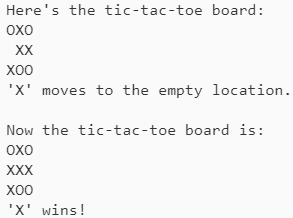


Строка string - это, в сущности, объект. Он предоставляет собственный набор функций-членов. Эти функции позволяют выполнять над объектом string ряд операций, от обычного получения длины конкретной строки до выполнения сложной подстановки символов. Кроме того, строки определяются именно так, что с ними можно легко и интуитивно понятно использовать операторы, с которыми мы уже успели познакомиться.

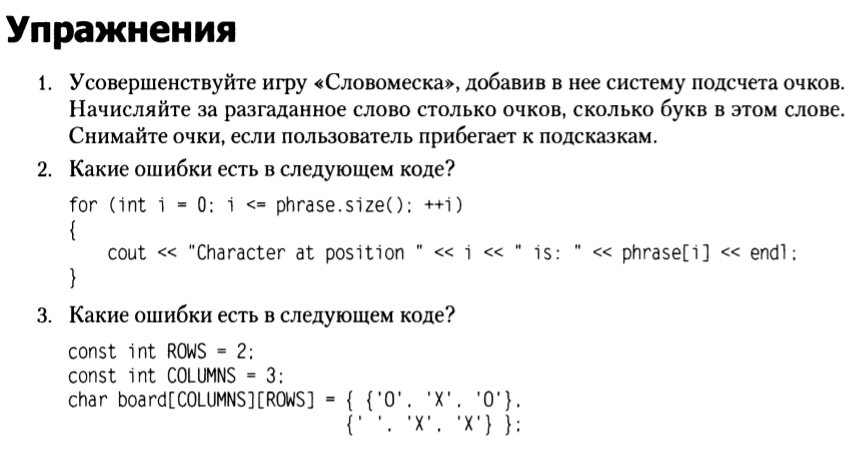


****

****

****

**Практическая часть**

****

(Файл с кодом лежит в папке code/)