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

### Standard Template Library

**Цели и задачи:**

* Познакомиться с понятием шаблонных типов в С++
* Познакомиться с понятием STL

**По результатам занятия слушатель будет знать:**

* Что такое объект-коллекция
* Что такое вектор в STL

**По результатам занятия слушатель будет уметь:**

* Работать с векторами и итераторами
* Использовать объекты STL

### 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

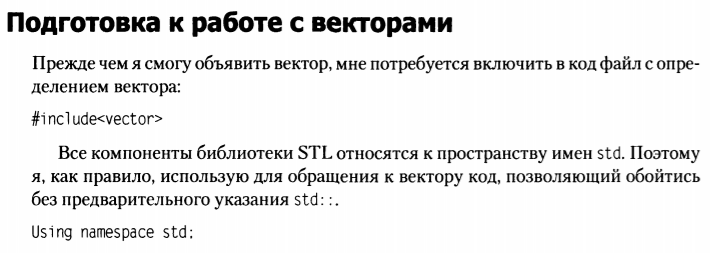
**Теоретическая часть**

Знакомство с библиотекой стандартных шаблонов Хороший программист-игровик знает, что лень - двигатель прогресса. Нет, мы не ленимся трудиться, просто не любим заново выполнять работу, которая когда-то уже была сделана, особенно если она была сделана хорошо. STL (библиотека стандартных шаблонов) - это замечательная коллекция готового качественного кода.

Что такое контейнер и как он может пригодиться при написании игр? В контейнере можно хранить коллекции, объединяющие значения одного типа, а также обращаться к этим значениям. Массивы тоже позволяют выполнять такие операции, но контейнеры STL - более гибкие и мощные сущности, чем старый добрый проверенный массив. В STL определяется целый ряд контейнерных типов; контейнеры разных типов работают немного по-разному.

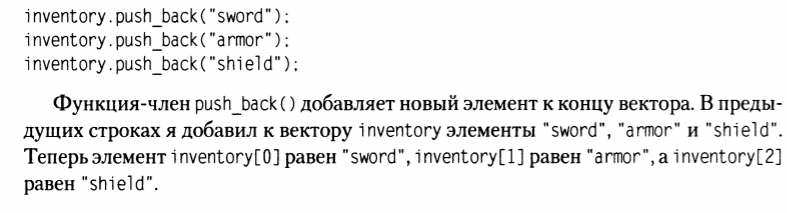
Работа с векторами Класс **vector** определяет один из видов контейнеров, предоставляемых в STL. В принципе, вектор - это динамический массив, то есть массив, способный при необходимости увеличиваться и уменьшаться в размере. Кроме того, в классе **vector** определяются функции-члены, предназначенные для манипуляций с элементами вектора. Таким образом, вектор не только не уступает массиву в функциональном отношении, но и превосходит его.

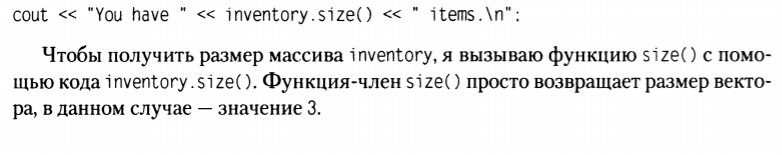
Вектор может расширяться, а массив - нет. Таким образом, если применить вектор для хранения игровых объектов, обозначающих врагов, то размер вектора будет адаптироваться под количество врагов, с которыми приходится сражаться герою. Если бы такая задача решалась с помощью массива, то понадобилось бы создать массив, способный содержать максимальное количество врагов. Но если бы в ходе игры потребовалось увеличить вражескую армию сверх этого размера, то вам пришлось бы туго.

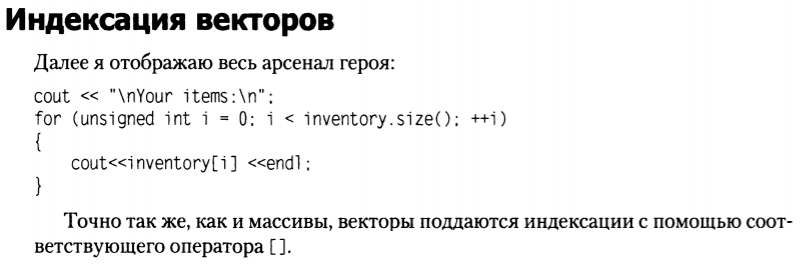


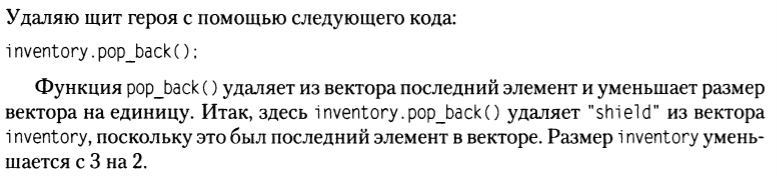
Изображение выглядит как снимок экрана

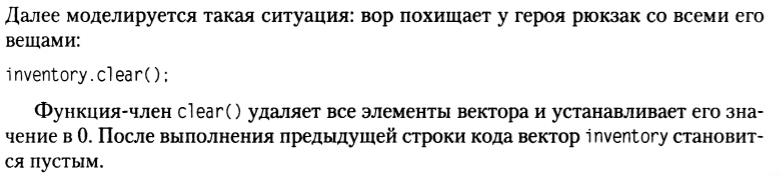
Автоматически созданное описание

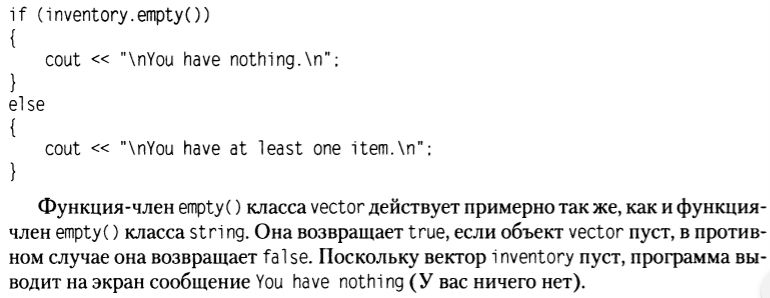
****

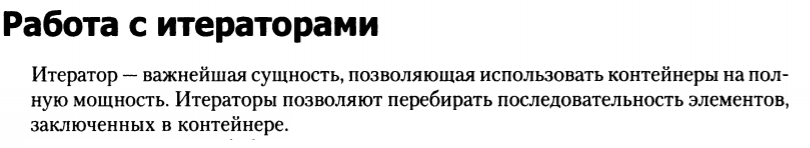
****

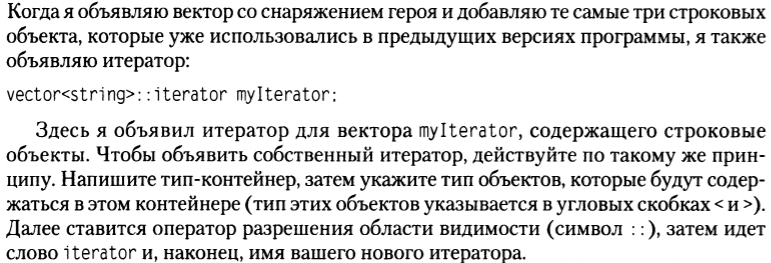
****

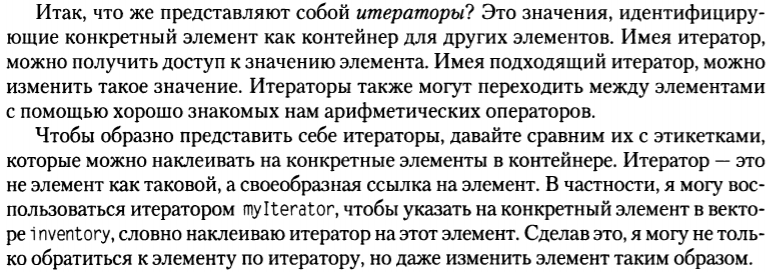
****

****

****

****

****

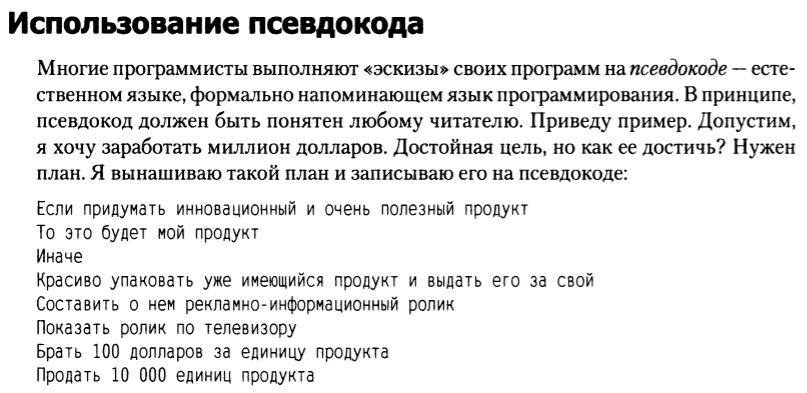
****

**Изображение выглядит как снимок экрана

Автоматически созданное описание**

**Изображение выглядит как снимок экрана

Автоматически созданное описание**

****

****

**Изображение выглядит как снимок экрана

Автоматически созданное описание**

**Практическая часть**

