



C++. Уровень 3

## Урок 7

# Поддержка модульности. Написание игры Blackjack

Единицы компиляции. Разделение на  
файлы заголовков и реализации.  
Написание игры Blackjack.

# План урока

- Термины программиста.
- Разделение текста программы на модули.
- Интерфейс и реализация.
- Разработка игры Blackjack.



# Термины



# Термины программиста

Исходный код

Компилятор

Компиляция

Объектный модуль

Компоновщик

Исполняемый модуль

Препроцессор

IDE

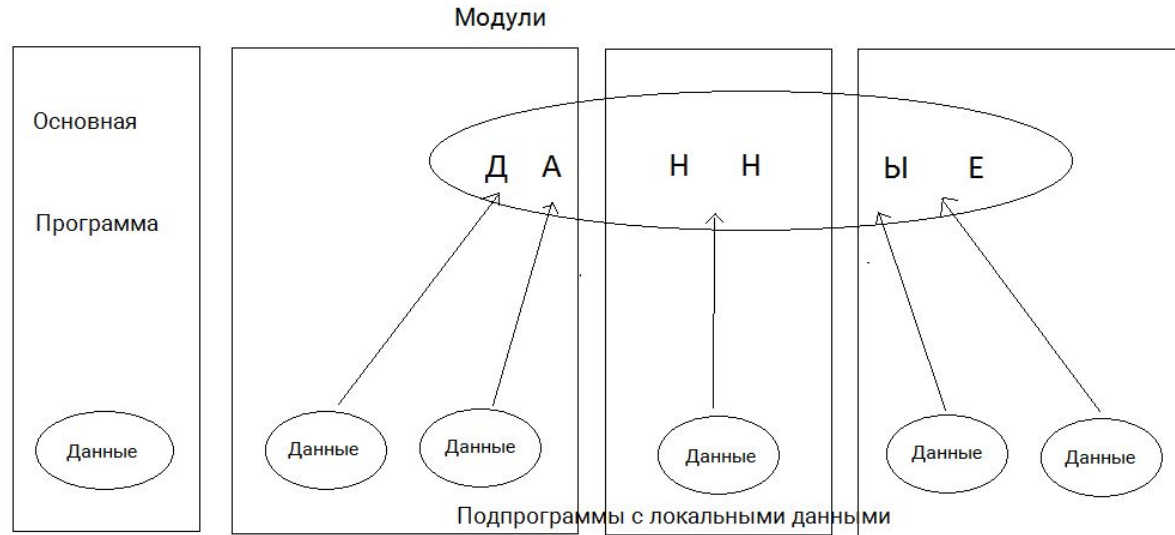
Определение

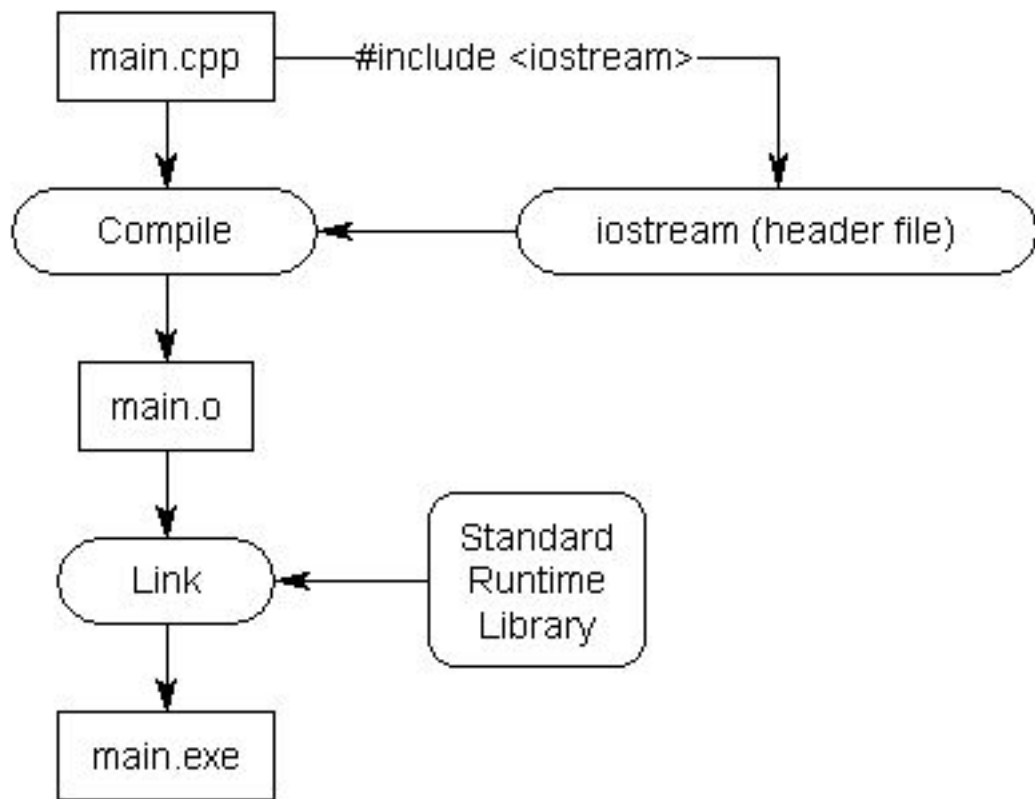
Объявление

Библиотека



# Разделение текста программы на модули





# Понятие интерфейса и реализации

**Интерфейс** — это внешний вид класса, выделяющий его существенные черты и не показывающий внутреннего устройства и секретов поведения.

**Реализация** — внутреннее представление класса, включая секреты его поведения.



# Практический пример





# Типичные ошибки

**Ошибка 1.** Определение в заголовочном файле.

**Ошибка 2.** Отсутствие защиты от повторного включения заголовочного файла.

**Ошибка 3.** Несовпадение объявления в заголовочном файле и определения в файле реализации.

**Ошибка 4.** Отсутствие необходимой директивы **#include**.

**Ошибка 5.** Отсутствие необходимого модуля в проекте построения программы.

**Ошибка 6.** Зависимость от порядка включения заголовочных файлов.



# Игра Blackjack



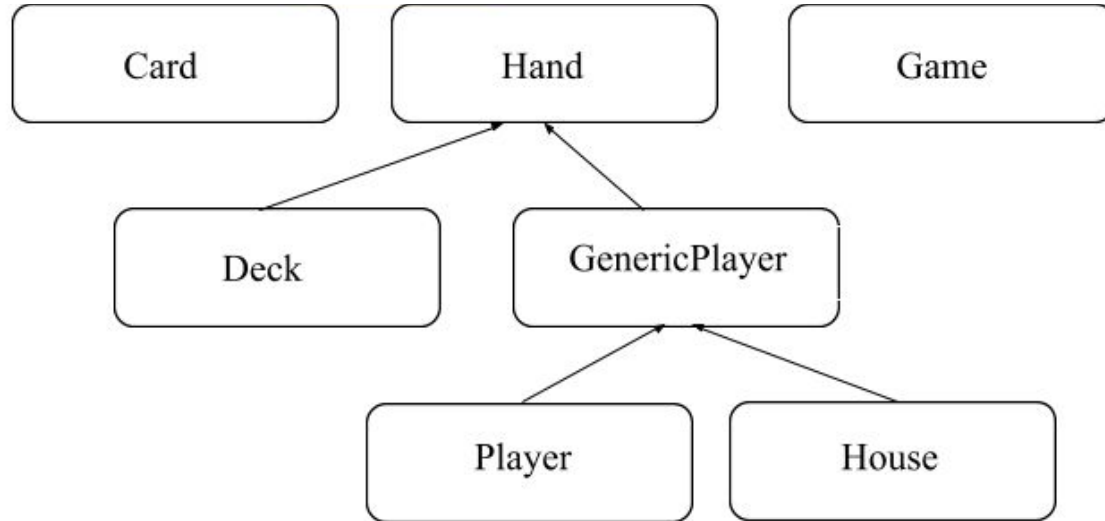


# Классы в программе

Класс	Родительский класс	Описание
Card	нет	Карта
Hand	нет	Набор карт, коллекция объектов класса Card
Deck	Hand	Имеет дополнительную функциональность, которая отсутствует в классе Hand, в частности тасование и раздачу
GenericPlayer	Hand	Обобщенно описывает игрока. Не является полноценным игроком, а лишь содержит элементы, характерные как для игрока-человека, так и для игрока-компьютера
Player	GenericPlayer	Человек-игрок
House	GenericPlayer	Компьютер-игрок
Game	нет	Игра



# Иерархия классов



# Класс Card

Член класса	Описание
<b>rank m_Rank</b>	Значение карты (туз, двойка, тройка и так далее). rank — это перечисление, куда входят все 13 значений
<b>suit m_Suit</b>	Масть карты (трефы, бубны, черви или пики). suit — это перечисление, содержащее все четыре возможные масти
<b>bool m_IsFaceUp</b>	Указывает, как расположена карта — лицом вверх или лицом вниз. Влияет на то, отображается она или нет
<b>int GetValue()</b>	Возвращает значение карты
<b>void Flip()</b>	Переворачивает карту. Может использоваться для того, чтобы перевернуть карту лицом вверх или вниз



# Класс Hand

Член класса	Описание
<b>vector&lt;Card*&gt; m_Cards</b>	Коллекция карт. Хранит указатели на объекты типа Card
<b>void Add(Card* pCard)</b>	Добавляет карту в руку. Добавляет указатель на объект типа Card в вектор m_Cards
<b>void Clear()</b>	Очищает руку от карт. Удаляет все указатели из вектора m_Cards, удаляя все связанные с ними объекты в куче
<b>int GetTotal()</b>	Возвращает сумму очков карт руки



# Класс GenericPlayer

Член класса	Описание
<b>string m_Name</b>	Имя игрока
<b>virtual bool IsHitting() const = 0</b>	Указывает, нужна ли игроку еще одна карта. Чистая виртуальная функция
<b>bool IsBoosted() const</b>	Указывает, что у игрока перебор
<b>void Bust() const</b>	Объявляет, что у игрока перебор





# Класс Player

Член класса	Описание
<b>virtual bool IsHitting() const</b>	Указывает, нужна ли игроку еще одна карта
<b>void Win() const</b>	Объявляет, что игрок выиграл
<b>void Lose() const</b>	Объявляет, что игрок проиграл
<b>void Push() const</b>	Объявляет, что игрок сыграл вничью



# Класс House

Член класса	Описание
<code>virtual bool IsHitting() const</code>	Указывает, нужна ли игроку еще одна карта
<code>void FlipFirstCard()</code>	Переворачивает первую карту



# Класс Deck

Член класса	Описание
<b>void Populate()</b>	Создает стандартную колоду из 52 карт
<b>void Shuffle()</b>	Тасует карты
<b>void Deal (Hand&amp; aHand)</b>	Раздает в руку одну карту
<b>void AdditionalCards (GenericPlayer&amp; aGenericPlayer)</b>	Раздает игроку дополнительные карты до тех пор, пока игрок может и хочет их получать



# Класс Game

Член класса	Описание
<b>Deck m_Deck</b>	Колода карт
<b>House m_House</b>	Рука дилера
<b>vector&lt;Player&gt; m_Players</b>	Группа игроков-людей. Вектор, содержащий объекты типа Player
<b>void Play()</b>	Проводит кон игры Blackjack



# Вопросы участников

