**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ И.РАЗЗАКОВА**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ**

**КУРСОВОЙ ПРОЕКТ**

ПО ДИСЦИПЛИНЕ:

**Функционально-ориентированное проектирование**

НА ТЕМУ: Функционально-ориентированное программирование

**ВЫПОЛНИЛ:** СТУДЕНТ ГР ПИ(англ)-1-16

АБАКИРОВ НУРСУЛТАН

**РУКОВОДИТЕЛЬ**: доц. МАКИЕВА З.Д.

**БИШКЕК 2016**

1. Условие задачи[[1]](#footnote-1).
2. Постановка задачи[[2]](#footnote-2).
3. Графическое представление алгоритма решения в виде блок-схемы (блок-схема для каждой функции, в том числе и для main).
4. Словесный (пошаговый алгоритм решения).
5. Программа на языке С++.
6. Тестовый пример: файл с исходными данными и файл с результатом.
7. Список литературы[[3]](#footnote-3).

Условие задачи

Программа генерирует два набора по 5 карт, определяет их комбинации и сравнивает их по правилам Покера.

Правила игры Покер

В покер играют разными колодами — по 32, 36 или 54 карты, но чаще всего используется стандартная колода из 52 листов с равнозначными мастями. Туз может рассматриваться и как младшая карта для образования последовательности (стрит) до 5 включительно, и как старшая (в комбинации туз- король— дама — валет — 10). Победителем считается тот, чья комбинация из пяти карт окажется лучшей, или тот, кто сможет вытеснить из игры других игроков с помощью ставок или блеф-ставок и останется один до вскрытия карт.

**Роял-флэш**

* ***Роял-флэш*** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *royal flush* — «королевская масть»): не является отдельной комбинацией, а является частным случаем стрит-флэша, старшим из всех возможных, и состоит из 5 старших (туз, король, дама, валет, десять) карт одной масти, например: **Т♥ К♥ Д♥ В♥ 10♥**.

**Стрит-флэш**

* **Стрит-флэш** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *straight flush* — «масть по порядку»): любые пять карт одной масти по порядку, например: **9♠ 8♠ 7♠ 6♠ 5♠**. Туз может как начинать порядок, так и заканчивать его. Самый младший стрит-флэш (от туза до пятёрки) иногда именуют «стальное колесо».

**Каре**

* **Каре**/**Четвёрка**/**Покер** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *four of a kind, quads* — «четыре одинаковых»): четыре карты одного достоинства, например: **3♥ 3♦ 3♣ 3♠**.

**Фулл-хауз**

* **Фулл-хауз**/**Полный дом**/**Три плюс два** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *full house, full boat* — «полный дом», «полная лодка»): одна тройка и одна пара, например: **10♥ 10♦ 10♠ 8♣ 8♥**.

**Флэш**

* **Флэш** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *flush* — «масть»): пять карт одной масти, например: **К♠ В♠ 8♠ 4♠ 3♠**.

**Стрит**

* **Стрит** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *straight* — «порядок») : пять карт по порядку любых мастей, например: **5♦ 4♥ 3♠ 2♦ Т♦**. Туз может как начинать порядок, так и заканчивать его. В данном примере **Т♦** начинает комбинацию и его достоинство оценивается в единицу, а **5♦** считается старшей картой. Самый младший стрит (от туза до пятёрки) иногда именуют «колесо».

**Сет/Триплет/Трипс/Тройка**

* **Сет/Триплет/Трипс/Тройка** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *three of a kind, set* — «три одинаковых», «набор»): три карты одного достоинства, например: **7♣ 7♥ 7♠** .

**Две пары**

* **Две пары** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *two pairs*): две пары карт, например: **8♣ 8♠ 4♥ 4♣**.

**Одна пара**

* **Одна пара** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *one pair*): две карты одного достоинства, например: **9♥ 9♠**.

**Старшая карта**

* **Старшая карта** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *high card*): ни одна из вышеописанных комбинаций, например (комбинация называется «старший туз»): **Т♦ 10♦ 9♠ 5♣ 4♣**.

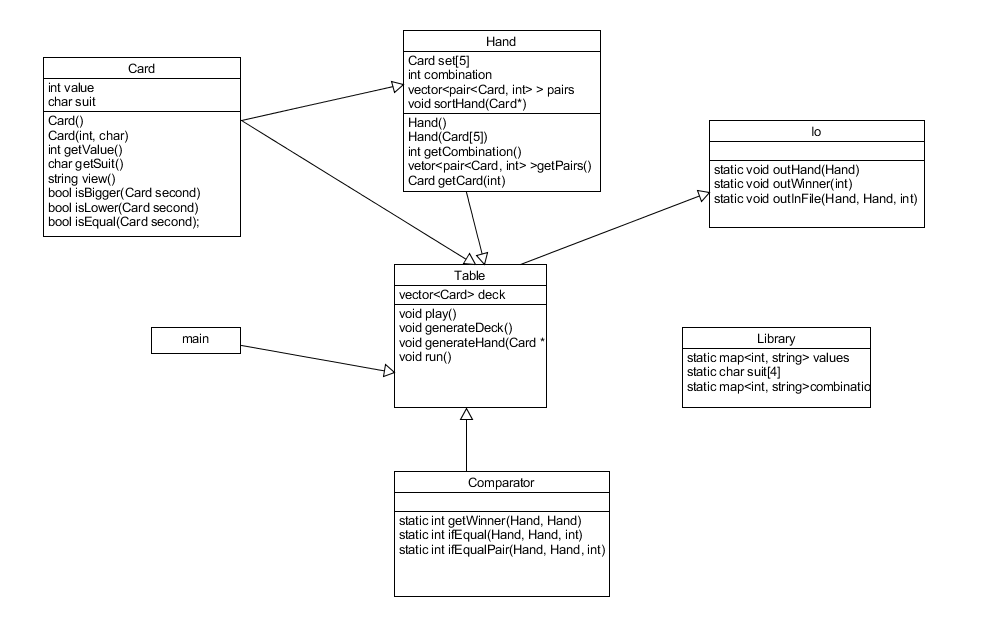
При совпадении комбинаций более сильной является комбинация со старшими картами, например, **8♣ 8♠ 4♥ 4♣ 2♠** старше, чем **7♣ 7♠ 5♥ 5♣ K♠**, а комбинация **6♠ 5♦ 4♥ 3♠ 2♦** старше, чем **5♦ 4♥ 3♠ 2♦ Т♦**.

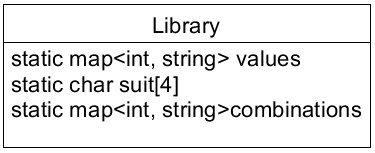
При совпадении комбинаций и старшей карты (если он находится среди 5 открытых карт) выигрыш делится поровну между игроками с одинаковой комбинацией. В случае совпадения первой старшей карты сравнивается с второй или третий (в случае пары и трёх старших карт)[[1]](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D0%B5%D1%80#cite_note-1).

В некоторых разновидностях покера (он называется Хай-Лоу ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Hi-Lo* от *High-Low*) выигрыш может делиться поровну между игроками, имеющими на руках самую сильную комбинацию из карт любого достоинства (рассмотрены выше) и самую слабую комбинацию из непарных карт достоинством не больше 8, например: **Т♣ 7♥ 2♠ 4♦ 8♠** (в слабых комбинациях туз считается картой низшего достоинства). При этом один и тот же игрок может одновременно иметь на руках самую сильную и самую слабую комбинацию карт.

Постановка задачи

Классы, используемые в проекте:





**Library –** объект библиотека, доступная всем модулям программы, с

Параметрами:

**values** = 0 - "2" 1 - "3" **suit** = 'C', 'H', 'S', 'D'

2 - "4" 3 - "5" ('C'- **♣** 'H'**- ♥** 'S'**- ♠** 'D'**- ♦**)

4 - "6" 5 - "7"

6 - "8" 7 - "9"

8 - "10" 9 - "J"

10 - "Q" 11 - "K"

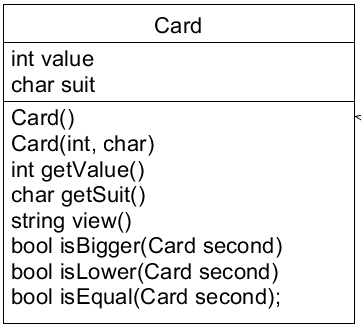
12 - "A"

**Combinations** = 0 - "HIGT CARD" 1 - "PAIR" 2 - "TWO PAIR"

3 - "THREE OF A KIND" 4 - "STRAIGHT" 5 - "FLUSH"

6 - "FULL HOUSE" 7 - "FOUR OF A KIND" 8 - "STRAIGHT FLUSH"

9 - "ROYAL FLUSH"

**Card** – это объект одной карты с

параметрами:

**value**(ключ)

**suit**(масть)

методами:

**Card(int, char)** – конструктор для заполнения объекта

**getValue()** – возвращает value объекта

**getSuit()** – возвращает suit объекта

**view()** – возвращает строку “value”(значение)+“suit”

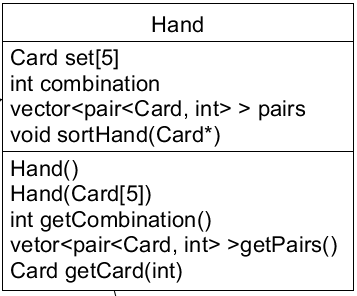
**isBigger(Card)** – возвращает true если ключ текущего объекта больше ключа объекта параметра, иначе false

**isLower(Card)** – возвращает true если ключ текущего объекта меньше

ключа объекта параметра, иначе false

**isEqual(Card)** – возвращает true если ключ текущего объекта равен

ключа объекта параметра, иначе false



**Hand** – это объект руки игрока с

параметрами:

**set[5]** – массив объектов Card

**combination** - ключ комбинации

**pairs** - массив из пар (карта, кол-во повторений этой карты), для определения комбинации

методами:

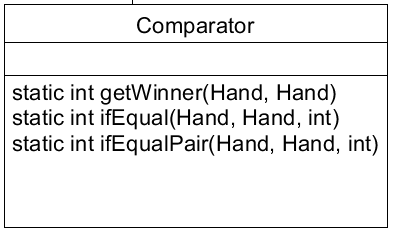
**Hand(Card[5])** – конструктор для заполнения объекта

**getCombination()** – возвращает combination объекта

**getPair()** – возвращает массив pairs объекта

**sortHand(Card\*)** – сортирует массив объектов Card по ключу параметра value

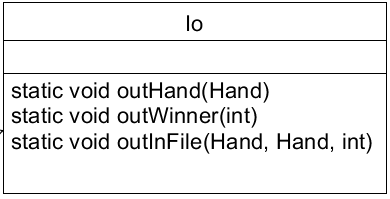
**getCard(int)** - возвращает объект Card из массива set по параметру(индекс)



**Comparator** – это объект сравнения со статическими

методами:

**getWinner(Hand, Hand)** – возвращает 1, если комбинация первого параметра выше комбинации второго параметра объектов Hand. Если же комбинация первого параметра ниже комбинации второго параметра возвращает 2. Если комбинация первого параметра равна комбинации второго параметра возвращает 0.



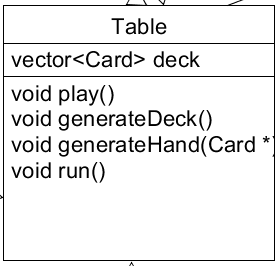
**Io** – это объект вывода со статическими

методами:

**outHand(Hand)** – выводит на экран карты и комбинацию Hand

**outWinner(int)** – выводит победителя. Если параметр равен 1, то победил первый игрок. Если 2, то второй. Если 0, то ничья.

**outInFile(Hand, Hand, int)** – записывает карты игроков, их комбинации и победителя в файл.



**Table** – это объект стола с

параметром:

**deck** – колода, массив объектов Card

методами:

**play()** – связывающий элемент, создает колоду, игроков, раздает карты, сравнивает их, выводит победителя.

**generateDeck()** – создает колоду из 52 карт

**generateHand(Card\*)** – берет 5 случайных карт с колоды и записывает их в параметр Card

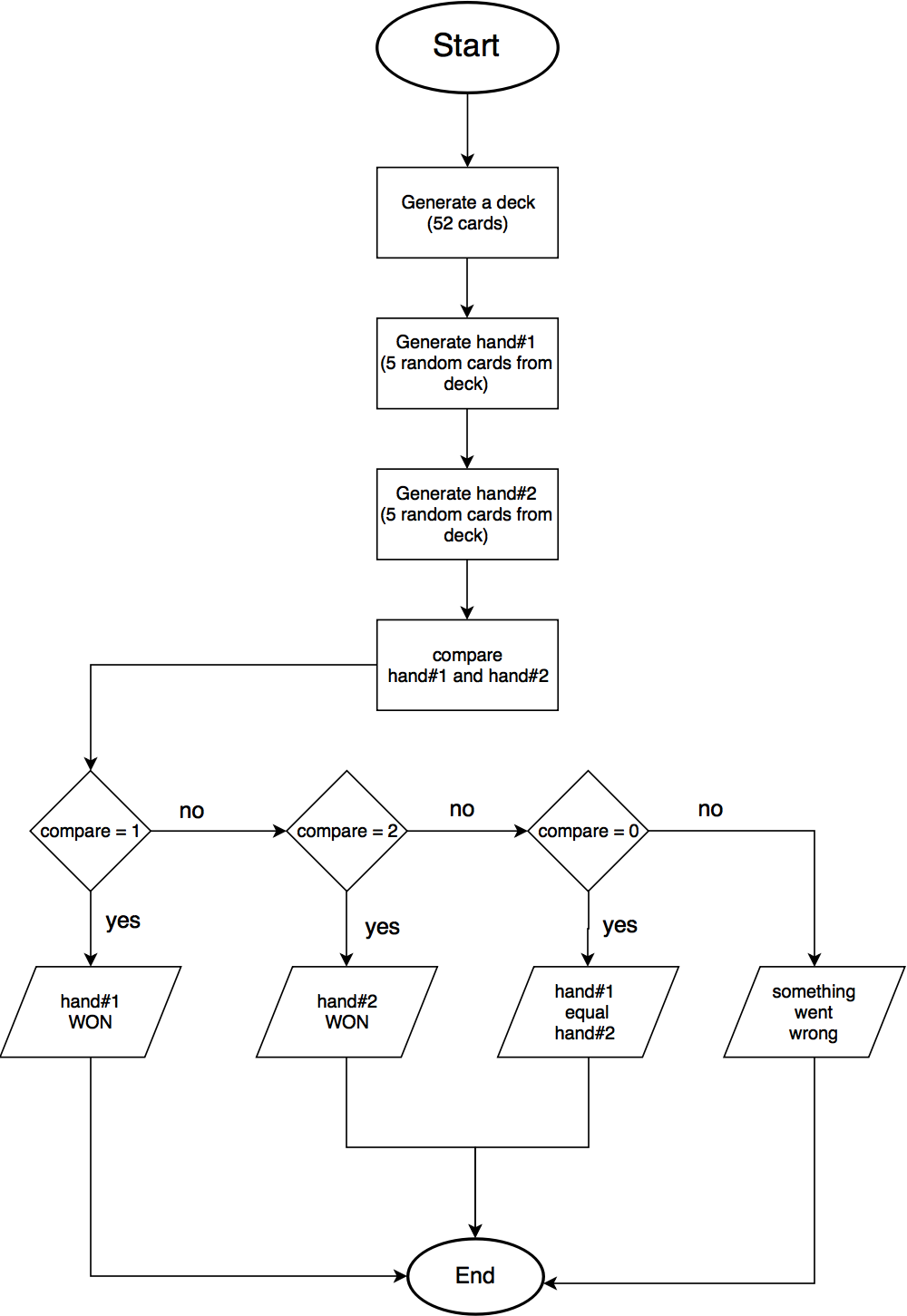
**run()** – дает пользователю выбор: 1 – пройти тесты

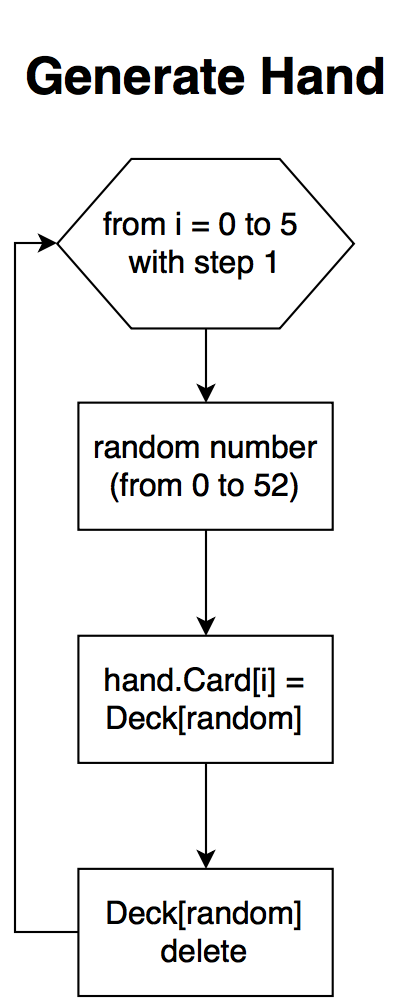
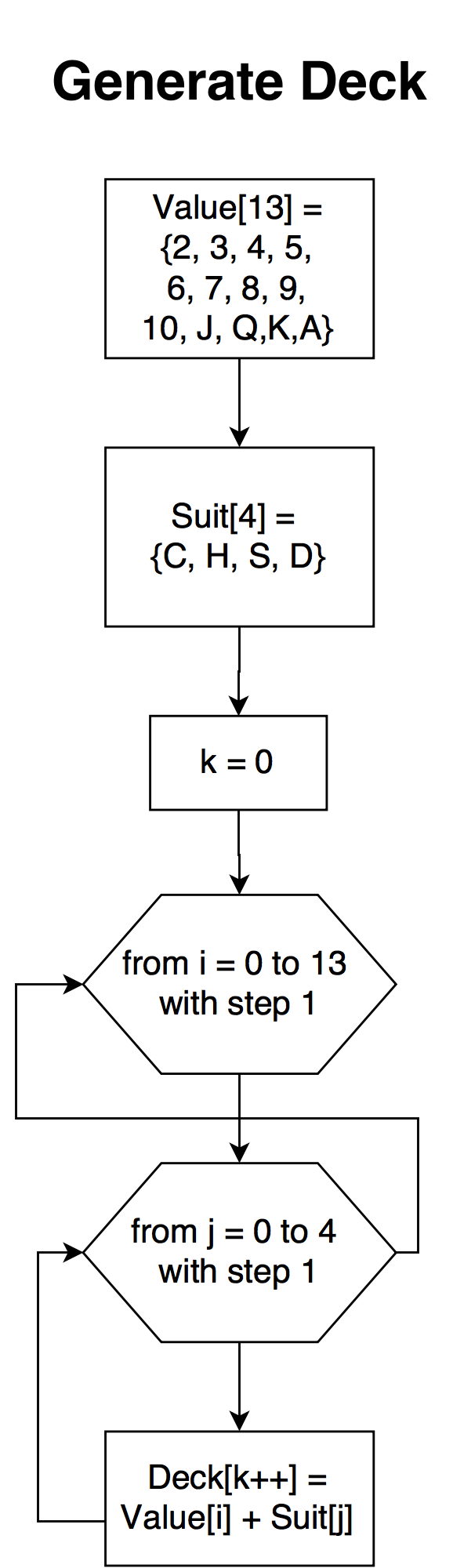
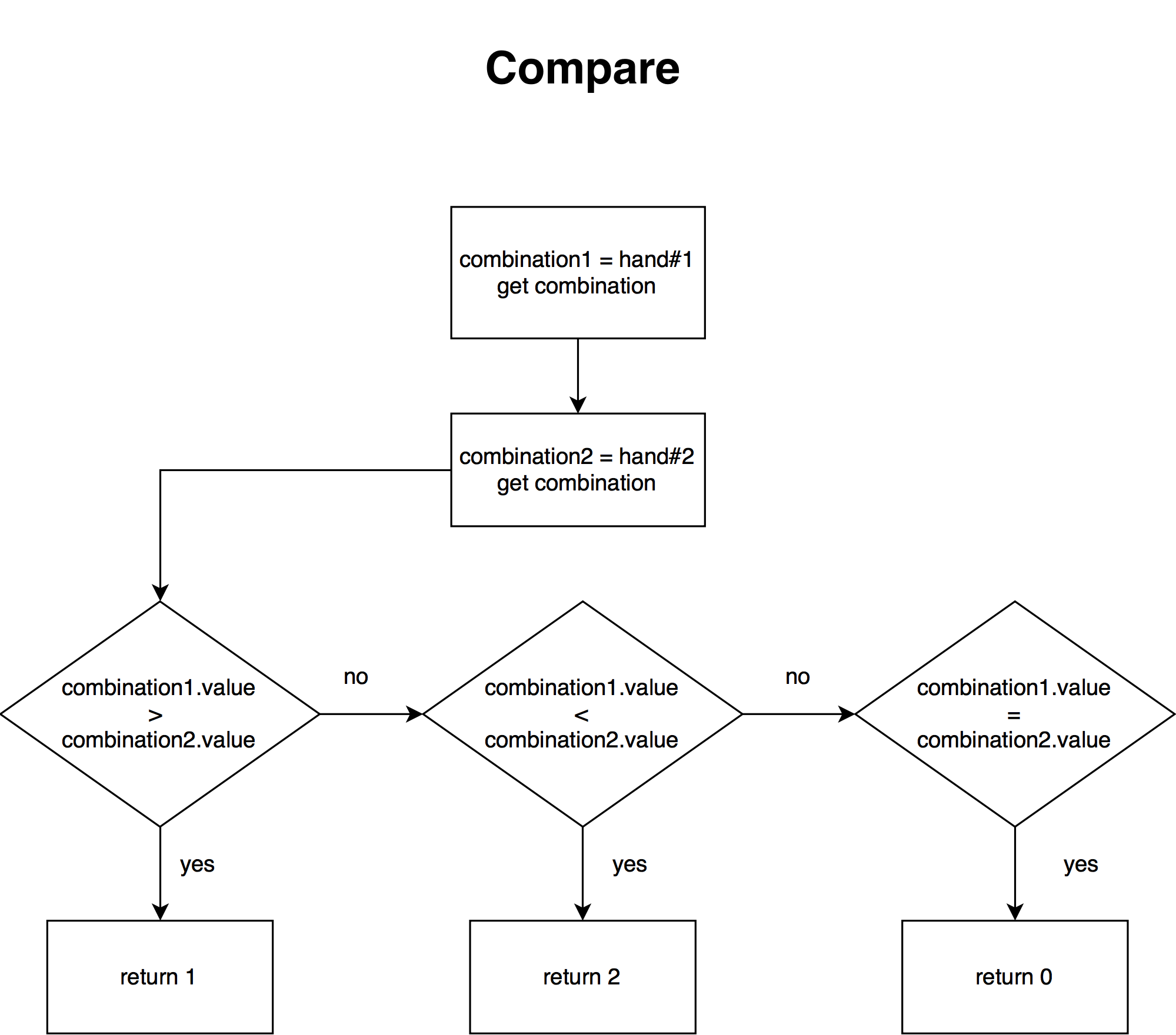
2 – начать игру

Графическое представление алгоритма решения в виде блок-схемы

Словесный (пошаговый алгоритм решения):

1. Создать колоду из 52 карт.
2. Раздать двум игрокам по пять случайных карт из колоды.
3. Сравнить карты двух игроков по правилам игры «Покер».
4. Вывести карты двух игроков и победителя или ничью.





Main.cpp

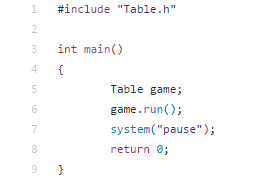
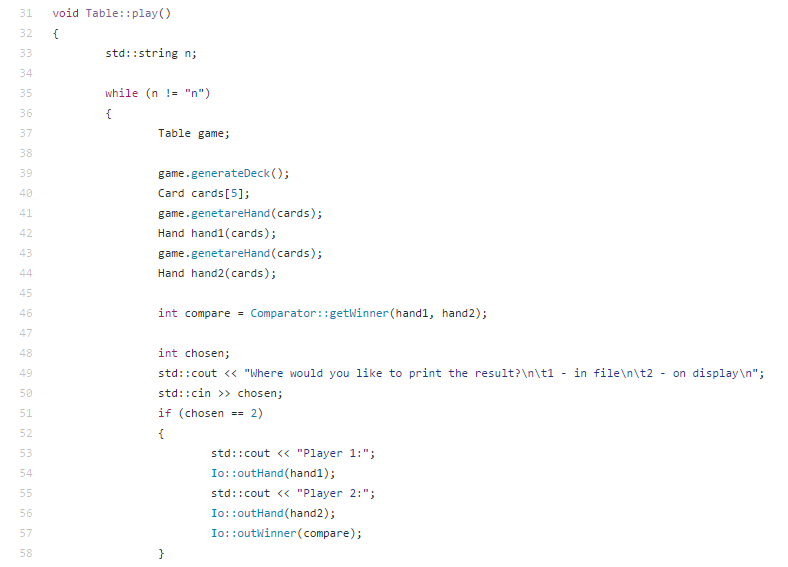
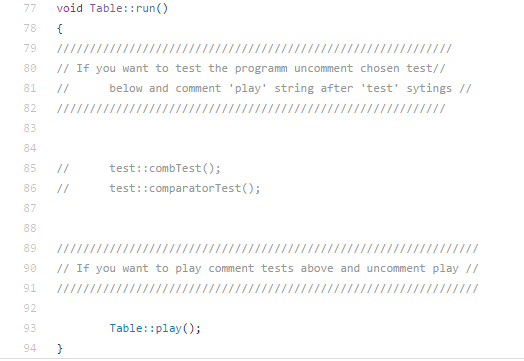
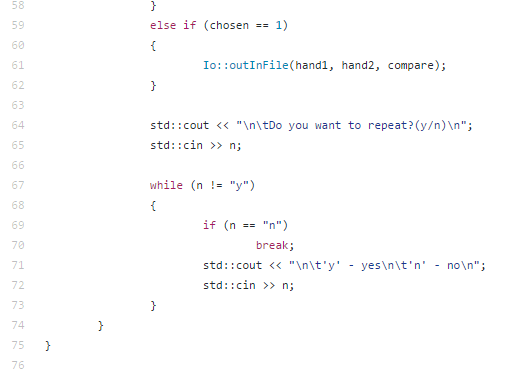
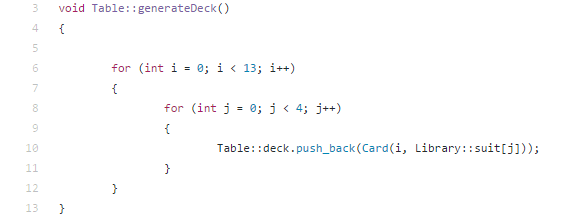
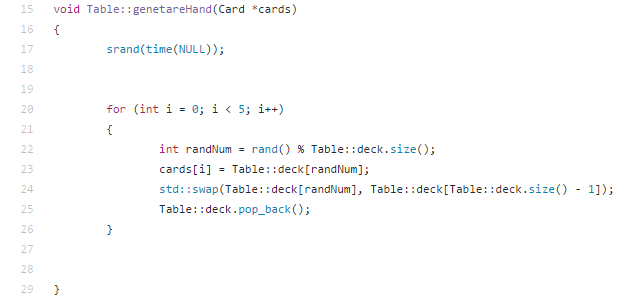


Table.cpp

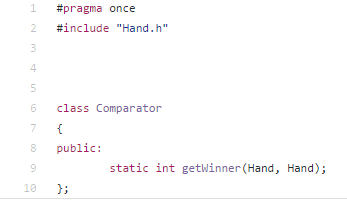




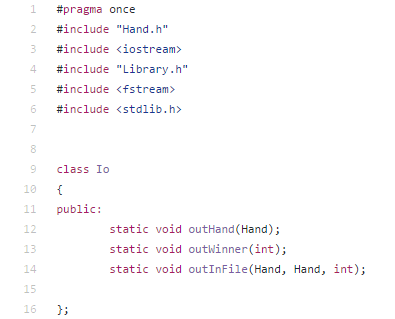




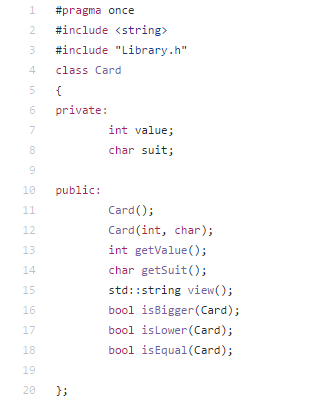
Comparator.h



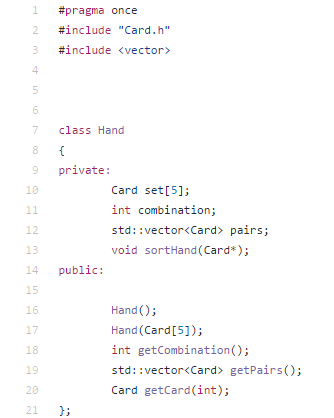
IO.h



Card.h



Hand.h



Library.h

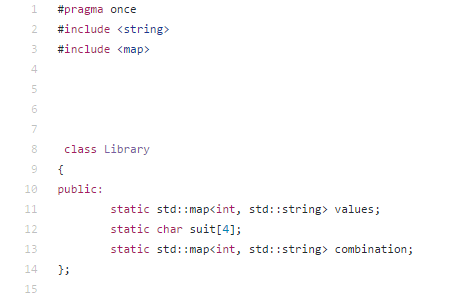
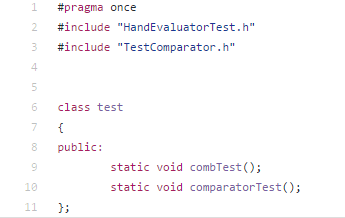


Table.h



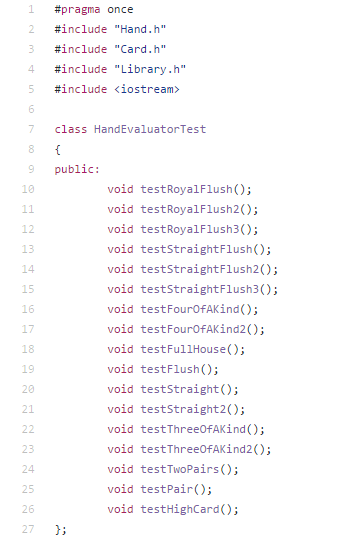
Test.h



ComparatorTest.h



EvaluatorTest.h



1. [1] Условие задачи в том виде, который приведен в учебнике или задании [↑](#footnote-ref-1)
2. [2] Постановка задачи включает подробное описание задачи: приводятся имена исходных данных (массивов, переменных), имена файлов, файлов, имена промежуточных данных и результата. Если необходимо по условию – формулы, которые будут использованы в вычислениях. [↑](#footnote-ref-2)
3. [3] Необязательный пункт [↑](#footnote-ref-3)