

Тестовая задача “Ханаби”

В этой задаче на объектно-ориентированное проектирование нужно создать систему классов, моделирующую карточную игру.



Правила игры

Мы используем упрощенные правила игры Ханаби. Они отличаются от стандартных.

Два участника играют специальными картами. У каждой карты есть цвет (красный, зеленый, синий, желтый и белый) и достоинство (от 1 до 5).

На вход программе подается список карт колоды. В колоде есть как минимум 11 карт. Первые 5 карт достаются первому игроку. Вторые 5 карт — второму.

Начало новой игры описывается такой входной строкой:

```
Start new game with deck R1 G2 B3 W4 Y5 R1 R1 B1 B2 W1 W2 W1
```

Эта игра начнется со следующими картами:

Первый игрок: R1 G2 B3 W4 Y5 (красная единица, зеленая двойка и т.д)

Второй игрок: R1 R1 B1 B2 W1

Верхняя карта колоды после такой раздачи — W2 (белая двойка)

Игра кооперативная и задача у игроков общая — выложить на стол как можно больше карт, пока игра не закончилась. Игроки держат карты рубашками к себе, то есть не видят своих карт, но видят чужие. Игроки не знают ни состав колоды, ни порядок карт в колоде.

Игроки ходят по очереди и пропускать ход нельзя. Ход — это одно из четырех действий:

1. Разыграть карту. Игрок выбирает одну из своих карт и кладет на стол.

Играть можно только карту, которой еще нет на столе. При этом достоинство карты должно быть единицей или на столе уже должна быть карта того же цвета, но на единицу меньшего достоинства. То есть разыгранная карта должна продолжать возрастающую последовательность одного цвета или начинать последовательность нового цвета.

Сыгранная карта удаляется из руки и помещается на стол. Игрок подбирает карту из колоды себе в руку, в конец списка карт.

Если корректно сыграть карту нельзя, то карта помещается в сброс, а игра немедленно

завершается.

Пример: **Play card 0** — текущий игрок выкладывает первую карту

Если игроку достоверно не известно, можно ли сыграть карту корректно, но он все же *успешно* разыгрывает эту карту, то ход считается *рискованным*. В этой задаче рискованными называются только такие ходы.

2. Сбросить карту. Игрок выбирает одну из своих карт, помещает её в сброс и подбирает одну карту из колоды себе в руку.

Пример: **Drop card 4** — текущий игрок сбрасывает последнюю карту из руки

3. Подсказать цвет. Игрок называет один из цветов и указывает на все карты этого цвета в чужой руке. Цвет, которого нет в чужой руке, называть нельзя.

Пример: **Tell color Red for cards 0 1 2 3 4** — текущий игрок сообщает другому игроку, что у него все карты красные

4. Подсказать достоинство. Игрок называет достоинство и указывает на все карты этого достоинства в чужой руке. Достоинство, которого нет в чужой руке называть нельзя.

Пример: **Tell rank 1 for cards 2 4** — текущий игрок сообщает, что у другого игрока в руке вторая и четвертая карты (нумерация с нуля) — единицы.

Игра завершается в трех случаях:

1. Когда колода заканчивается. То есть, последняя карта колоды никогда не играет.
2. Когда на столе 25 карт. То есть, когда больше ни одну карту выложить нельзя.
3. Когда один из игроков делает некорректный ход: разыгрывает некорректную карту или сообщает другому игроку ложную или неполную информацию о его картах.

Все ходы после завершения игры нужно игнорировать.

Постановка задачи

Программе будут вводить через консоль описание игр одна за другой. Для каждой завершенной игры из входных данных программа должна вывести одну строку, содержащую номер хода, на котором игра завершилась, количество корректно сыгранных карт, а также количество успешных рискованных ходов.

Пример начала входных данных	Пример начала выходных данных
Start new game with deck Y4 R4 Y1 W4 W3 W4 W5 W2 R1 Y1 B1 G3 Tell color Yellow for cards 4 Tell rank 1 for cards 2 Play card 2 Drop card 2 ...	Turn: 101, cards: 9, with risk: 2 Turn: 12, cards: 1, with risk: 0 Turn: 2, cards: 1, with risk: 1 ...

Подробнее правила игры и формат ввода-вывода можно изучить по примерам ввода и вывода [по этой ссылке](#). Настоятельно рекомендуем тестировать программу на этих примерах!

Уровни сложности

Задача разбита на уровни сложности. Не беда, если решишь только первый — идеальные решения этого уровня смогут побороться с остальными!

1. На первом уровне гарантируется, что рискованных ходов нет. То есть каждая строка в выходных данных должна заканчиваться `risk: 0`
2. На втором уровне рискованные ходы есть.

Оформление решения

1. Решение должно целиком содержаться в одном cs-файле с кодом на C#, имя которого составлено по шаблону: `<surname>_<name>.cs` где **surname** и **name** — твои фамилия и имя *латиницей*. Можно использовать C# версий 5 или 6 (Visual Studio 2013 и Visual Studio 2015, соответственно)
2. Должно читать входные данные из консоли (`Console.ReadLine()`) по одной команде в строке и выводить результат работы на консоль (`Console.WriteLine()`) до тех пор пока ввод не закончится (то есть пока `Console.ReadLine()` не вернет `null`).
3. Должно в точности соблюдать формат ввода и вывода. Проверить точное совпадение вывода программы с эталонным выводом можно на предоставленных примерах (см. ссылку выше)
4. Не должно делать потенциально вредоносных действий — обращаться к сети, к файловой системе, запускать и взаимодействовать с другими приложениями, и т.п.
5. Не должно использовать сторонние библиотеки, кроме разрешенных:
 - [System.Collections.Immutable](#)
 - [NUnit](#)

Если нужно еще что-то — пишите на kontur-student@skbkontur.ru, мы подумаем! (Пространства имен `System.Collections.Generic`, `System.Text.RegularExpressions` и подобные являются частью .NET Framework и разрешены к использованию)

Оценка решений

В первую очередь мы оцениваем корректность. За нарушение формата ввода вывода решение дисквалифицируется.

Затем мы оцениваем качество кода. Форматирование, выбор имён для классов, методов и переменных, разбивка кода на классы, отсутствие дублирования кода. Старайся делать код коротким, понятным и логичным.

Полезные ссылки

- Что стоит считать плохим кодом, а что нет? Мы сделали игру-тренажер, по базовым навыкам приведения кода в порядок. <http://cleancodegame.github.io>
А ещё лучше прочитать книгу "[Чистый код](#)" Роберта Мартина или хотя бы пробежаться по её [краткому конспекту](#).
- Набор для игры в Ханаби можно купить в магазине настолок, например, [тут](#).
- Вопросы по тестовому заданию задавайте на почту kontur-student@skbkontur.ru