**Задание 5**

**Задачи для пятого задания Практикума по программированию. Общая тема задания «Шахматный симулятор: объектно-ориентированная версия».**

Базовая часть (выполняется всеми самостоятельно!):

На базе собственной реализации Задания 4 создать объектно-ориентированную реализацию программы для игры в шахматы.

Базовые требования к функциональности программы сохраняются прежними:

Реализовать программу, которая позволяет играть в шахматы на компьютере. Взаимодействие с программой производится через консоль (базовый вариант). Игровое поле изображается в виде 8 текстовых строк, плюс строки с буквенным обозначением столбцов (см. пример на Рис. 1) и перерисовывается при каждом изменении состояния поля. При запросе данных от пользователя программа сообщает, что ожидает от пользователя (например, позицию фигуры для следующего хода белыми; целевую позицию выбранной фигуры) и проверяет корректность ввода (допускаются только ходы соответствующие правилам шахмат; поддержка рокировки, сложных правил для пешек и проверки мата вынесена в отдельные пункты). Программа должна считать количество сделанных ходов.

Изображение выглядит как текст, электроника

Автоматически созданное описание

Рис. 1 Пример изображения шахматного поля в текстовом режиме

Сама программа НЕ ходит: т.е. не пытается выполнить ходы за одну из сторон, а предоставляет поочередно вводить ходы за белых и черных.

Требования к реализации:

Основные объекты и абстрактные сущности игры должны быть представлены в виде объектов, представителей соответствующих классов, часть классов должны быть организованы в виде иерархии. В частности: шахматные фигуры – объекты, представители классов, организованных в виде иерархии; доска – объект; ходы фигур – объекты. Вся основная информация должна храниться в атрибутах объектов или классов (например, информация о положении фигур, цвете фигур, символах, используемых для визуализации фигур и т.п.). Основная часть функционала должна программы должна быть организована в виде методов, закрепленных за соответствующими объектами или классами. Например, это касается методов определяющих допустимые ходы фигур. Организация иерархий классов, атрибутов и методов должна позволять гибко расширять возможности программы с минимальными изменениям в уже созданном коде.

**Дополнительные задания:**

1. Придумать 3 новых вида фигур с оригинальными правилами перемещения и реализовать их классы. Создать модификацию шахмат с новыми фигурами с минимальным вмешательством в существующий код.

**Сложность 1**

1. На базе игры в шахматы реализовать игру в шашки. Разработать модификацию шахмат с минимальным вмешательством в существующий код.

**Сложность 2**

1. На базе игры в шахматы на классической доске реализовать игру в гексагональные шахматы ( <https://ru.wikipedia.org/wiki/Гексагональные_шахматы> ). Выбрать один из трех вариантов: шахматы Глинского; шахматы МакКуэя; шахматы Шафрана. Разработать модификацию шахмат с минимальным вмешательством в существующий код для обычных шахмат.

**Сложность 3**

1. На базе игры в шахматы на классической доске реализовать игру в гексагональные шахматы на троих ( <https://ru.wikipedia.org/wiki/Шахматы_для_троих> ). Выбрать один из существующих вариантов. Разработать модификацию шахмат с минимальным вмешательством в существующий код для обычных шахмат.

**Сложность 4**

1. Реализовать возможность «отката» ходов. С помощью специальной команды можно возвращаться на ход (или заданное количество ходов) назад вплоть до начала партии. Информация о ходах в партии должна храниться в объектно-ориентированном виде.

**Сложность 1**

1. Реализовать функцию подсказки выбора новой позиции фигуры: после выбора фигуры для хода функция визуально на поле показывает поля доступные для хода или фигуры соперника, доступные для взятия, выбранной фигурой. Информация о допустимых ходах должна храниться в объектно-ориентированном виде, алгоритм без модификации должен работать при добавлении новых типов фигур (задание берется совместно с Заданием 1).

**Сложность 1**

1. Реализовать функцию подсказки угрожаемых фигур: она возвращает информацию о том, какие фигуры ходящего игрока сейчас находятся под боем (т.е. могут быть взяты соперником на следующий ход) и визуально выделяет их на поле. Функция отдельно указывает на наличие шаха королю. Информация о допустимых ходах должна храниться в объектно-ориентированном виде, алгоритм без модификации должен работать при добавлении новых типов фигур (задание берется совместно с Заданием 1).

**Сложность 1**

1. Реализовать поддержку для пешки сложных правил: «взятие на проходе» и замены на другух фигуру при достижении крайней горизонтали (в базовой версии их поддержка не обязательна, но возможность первого хода на одну или две горизонтали - обязательно). Подробнее о правилах см.: <https://ru.wikipedia.org/wiki/Правила_шахмат> . Информация о допустимых ходах должна храниться в объектно-ориентированном виде, алгоритм без модификации должен работать при добавлении новых типов фигур со сложным поведением (задание берется совместно с Заданием 1 и как минимум одна из новых фигур должна иметь сложное поведение, т.е. изменение правил хода и взятия фигуры в зависимости от дополнительных условий).

**Сложность 1**

**Для каждого студента формируется комплексное задание из сочетания пунктов. Суммарная сложность комплексного задания должна быть не менее 5, как минимум одна задача из комплекта должна стоить дороже 1. Приветствуется выполнение заданий с суммарной сложностью более 5 (рекомендуется отмечать такие решения дополнительными баллами).**

**Сложность комплексного задания может быть скорректирована преподавателем в зависимости от сочетания пунктов между собой. В случае неоднозначности сочетания требований двух пунктов студент предлагает преподавателю свое видение итоговой постановки задачи, а преподаватель согласует его (рекомендуется фиксировать договоренность в письменном виде). Преподаватель имеет право увеличить оценку сложности комплексного задания в случае существенного увеличения трудоемкости при сочетании двух пунктов.**

**Преподаватель имеет право добавить пункты с собственной постановкой задачи и оценкой сложности. Студент может предложить свои пункты и реализовывать их в случае предварительно согласования с преподавателем постановки задачи и оценки сложности пункта. Рекомендуется фиксировать дополнительные пункты письменно. Желательно, чтобы дополнительные пункты составляли не более половины суммарной трудоемкости комплексного задания**