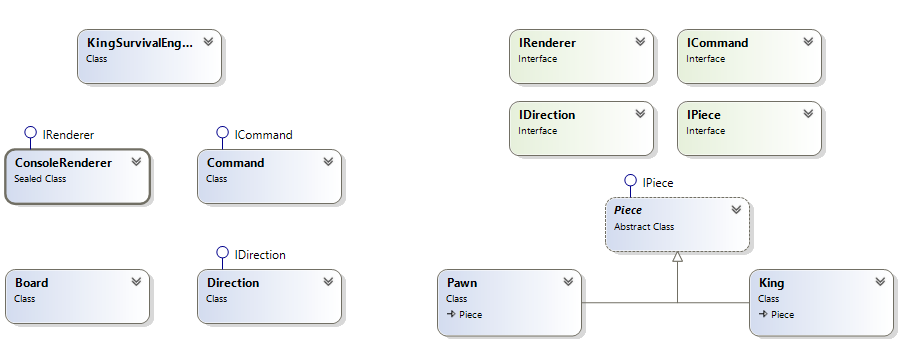
**Отбор King Survival 1**

****

**Н**ашият отбор има за задача да рефакторира и допълни дадения код за създаване на играта King Survival. В нея на стандартна шахматна дъска 8x8 полета имаме цар King и четири пешки PawnA, PawnB, PawnC и PawnD. Играчът задава команди за движение на царя и на пешките, като целта е или царят да стигне до row 0 на дъската и да победи, или да бъде убит от пешките. Играчът движи фигурите и на царя, и на пешките чрез команди в конзолата.

**Д**аденият ни код беше рефакторирен, преименуван, променен и допълнен, като в резултат целият код е организиран в King Survival Solution. В него имаме обособени три папки – Engine, GamePlay Classes, Interfaces и отделно файловете KingSurvival.Tests и StartGame.cs. Диаграмата на проекта изглежда по следния начин.



**Файлове:**

**KingSurvival.Tests.** – Съдържа unit тестове на кода, разделени логически в отделни файлове – Pawn Tests, King Tests, Direction Tests, Board Tests.

**Games**.cs – тук се съдържа Main.cs класът на проекта и от него се стартира Engin-a.

Папка **Engine**:

**KingSurvivalEngine**.cs – В метода Run() се прави смяната на ходовете на двете страни – царят и пешките и се извършва разпознаване на зададените команди за движение и изпълнението им чрез метода ProcessCommand(). Тук се прави и проверка за край на играта и кой е победителят и се дават упътвания на конзолата за случващото се в процеса на игра.

**Constants**.cs – съдържа константите в играта – първоначалните позиции на фигурите и техния брой.

**Console.Renderer**.cs – наследява интерфейса IRenderer. Служи за изчистване и изрисуване на игралната дъска след всеки ход.

**Command.cs** – наследява интерфейса ICommand и превръща зададения вход в команда за движение.

**PieceFactory.cs –** тук се създават фигурите и им се задават координати. Наследява интерфейса IPieceFactory.

**Папка Formatters** – в нея има форматиращи методи, наследяващи IFormatter.

Папка **GamePlay** **Classes**:

**Board**.cs – Класът представя игралната дъска, представляваща матрица от бели и черни полета.

**Direction.cs –** Задава и ъпдейтва посоките на движение на фигурите.

**King**.cs – Класът на царя, в който се съдържат конструктор за King и се имплементира метода Move(), който изпълнява движението на царя и брояч за направените ходове.

**Pawn**.cs – Класът на пешките, в който се съдържат конструктор за пешка и се прави проверка за валидността на зададения символ на пешката.

**Piece**.cs – наследява интерфейса IPiece и съдържа конструктор за създаване на фигурите.

**Validations**.cs – класът служи за правене на проверки по отношение на командите – дали са правилно зададени и могат ли да бъдат изпълнени.

Папка **Interfaces**:

**ICommand.cs –** свързва се с командите за движение.

**IDirection.cs –** задава координатите при движението.

**IFormatter.cs –** използва се за правилното форматиране на символите.

**IPiece.cs –** представя движението на фигурите по дъската.

**IPieceFactory.cs –** свързва се със създаването на фигурите.

**IRenderer**.cs – съдържа метода Render() и се отнася до прерисуването на дъската след всеки ход.

***Линк към GitHub на проекта*:** <https://github.com/TeamKingSurvival1/KingSurvival/>

**Участници в проекта:**

Александър Димитров – **a.dimitrov**

Иван Бангиев – **bangiev**

Крум Тюкенов – **Overon**

Даниела Пенкова – **Regina05**

Мирослав Цаков – **miroslav.tsakov**

Ангел Великов – **angel\_rw**

Росица Бойкова – **rosica.boykova**