

```
1 public interface AI {  
2  
3     /**  
4     @pre: --.  
5     @post: S'ha calculat la següent jugada.  
6     **/  
7     void CalcularJugada();  
8 }  
9
```

```

1 import com.sun.org.apache.xpath.internal.operations.Bool;
2
3 public interface Game {
4     //Pre: ---
5     //Post: Inicialitza el joc, crea la pantalla.
6     public void Init(String sTitle, int iSizeX, int iSizeY
7         , Boolean bFullScreen, String asArgv[]);
8
9     //Pre: ---
10    //Post: Nateja l'estat actual i Guarda i inicialitza el
11    nou estat
12    public void ChangeState(GameState oState);
13
14    //Pre: ---
15    //Post: Pausa l'estat actual i Guarda i inicialitza el
16    nou estat
17    public void PushState(GameState oState);
18
19    //Pre: ---
20    //Post: Nateja l'estat actual i reprèn l'estat anterior
21    public void PopState();
22
23    //Pre: ---
24    //Post: Llegeix els events d'entrada actuals.
25    public void HandleEvents();
26
27    //Pre: ---
28    //Post: Actualitza els elements del joc en funció del
29    evenets de teclat rebuts.
30    public void Update();
31
32    //Pre: ---
33    //Post: Dibuixa l'estat actual per pantalla
34    public void Draw();
35
36    //Pre: ---
37    //Post: True: El joc està en marxa, False: El joc està
38    parat
39    public Boolean Running();
40
41    //Pre: El joc està en marxa
42    //Post: Parla el joc
43    public void Quit();
44
45    //Pre: --
46    //Post: Buida els estats del joc

```

```
42     public void Clean();  
43 }  
44
```



```

1 public interface Peca {
2
3 // "nom": "TORRE",
4 // "simbol": "T",
5 // "imatgeBlanca": "torreB.png",
6 // "imatgeNegra": "torreN.png",
7 // "valor": 5,
8 // "moviments": [[0,"a",1,0], ["a",0,1,0]],
9 // "movimentsInicials": [],
10 // "promocio": false,
11 // "invulnerabilitat": false
12
13 // private String sSimbol;
14 // private String sImatgeBlanca;
15 // private String sImatgeNegra;
16 // private int iValor;
17 // private vector<Moviment> vMoviment;
18 // private vector<Moviment> vMovimentInicial;
19 // private boolean bPromocio;
20 // private boolean bInvulnerabilitat;
21 // private String sPosX;
22 // private String sPosY;
23
24 /**
25  @pre: --
26  @post: Inicialitza la classe (constructor per defecte)
27  */
28 // Peca();
29
30 /**
31  @pre: --
32  @post: Inicialitza la peça;a amb els seus atributs
33  */
34 // Peca(String simbol, String imatgeBlanca, String
    imatgeNegra, int valor, boolean promocio, boolean
    invulnerabilitat);
35
36 /**
37  @pre: La peça es pot moure
38  @post: Modifica la posició de la peça
39  */
40 public void moure(Peca p, Moviments m);
41
42 }

```

```
1 public interface Match {  
2 }  
3
```



```
1 public interface Jugada {
2
3     /**
4     @param sPosOrigen És la posició d'origen.
5     @param sPosDesti És la posició de destí.
6     @pre: --.
7     @post: S'ha fet el moviment.
8     */
9     void ferMoviment(String sPosOrigen, String sPosDesti
10 ); // ???
11 }
```



```
1 public interface Tauler {  
2  
3  
4 }  
5
```



```
1 public interface Jugador {
2
3     /**
4     @pre: --.
5     @post: El jugador s'ha rendit.
6     */
7     void Rendirse();
8
9     /**
10    @pre: --.
11    @post: El jugador ha sol·licitat que s'ajorni la partida.
12    */
13    void Ajornar();
14
15    /**
16    @pre: És el torn d'aquest jugador.
17    @post: El jugador ha fet una jugada.
18    */
19    void FerJugada();
20 }
21
```



```
1 public interface Partida {
2
3     /**
4     @pre: --
5     @post: Inicia una partida
6     */
7     public void iniciar();
8
9     /**
10    @pre: Partida en joc
11    @post: Guarda l'estat de la partida
12    */
13    public void guardar();
14
15    /**
16    @pre: Partida en joc
17    @post: Pausa la partida
18    */
19    public void pausar();
20
21    /**
22    @pre: Partida pausada
23    @post: Reprén la partida
24    */
25    public void rependre();
26
27 }
```



```
1 import java.awt.event.*;
2
3 public interface GameState {
4     //Pre: ---
5     //Post: Inicialitza el GameState
6     public void Init(Game game);
7
8     //Pre: ---
9     //Post: Nateja el GameState
10    public void Clean();
11
12    //Pre: ---
13    //Post: Pausa el GameState
14    public void Pause();
15
16    //Pre: ---
17    //Post: Retorna el GameState
18    public void Resume();
19
20    //Pre: ---
21    //Pos: Llegeix les entrades del GameState
22    public void HandleEvents(Game game);
23
24    //Pre: ---
25    //Post: Actualitza les dades del GameState
26    public void Update(Game game);
27
28    //Pre: ---
29    //Post: Dibuixa el GameState
30    public void Draw(Game game);
31 }
32
```



```
1 public interface Moviments {
2
3 // private String sVertical;
4 // private String sHoritzontal;
5 // private boolean bPotCapturar;
6 // private boolean bPotSaltar;
7
8 /**
9  @pre: --
10  @post: Inicialitza la classe (constructor per defecte)
11  */
12 // Moviments();
13
14 /**
15  @pre: --
16  @post: Inicialitza la classe amb els valors de la peça
17  */
18 // Moviments(String sVertical, String sHoritzontal,
19 // boolean potCapturar, boolean potSaltar);
20
21 /**
22  @pre: --
23  @post: retorna cert si el moviment es pot realitzar per
24  la Peça
25  */
26 public boolean esValid(Moviments m, Peca p);
27 }
```