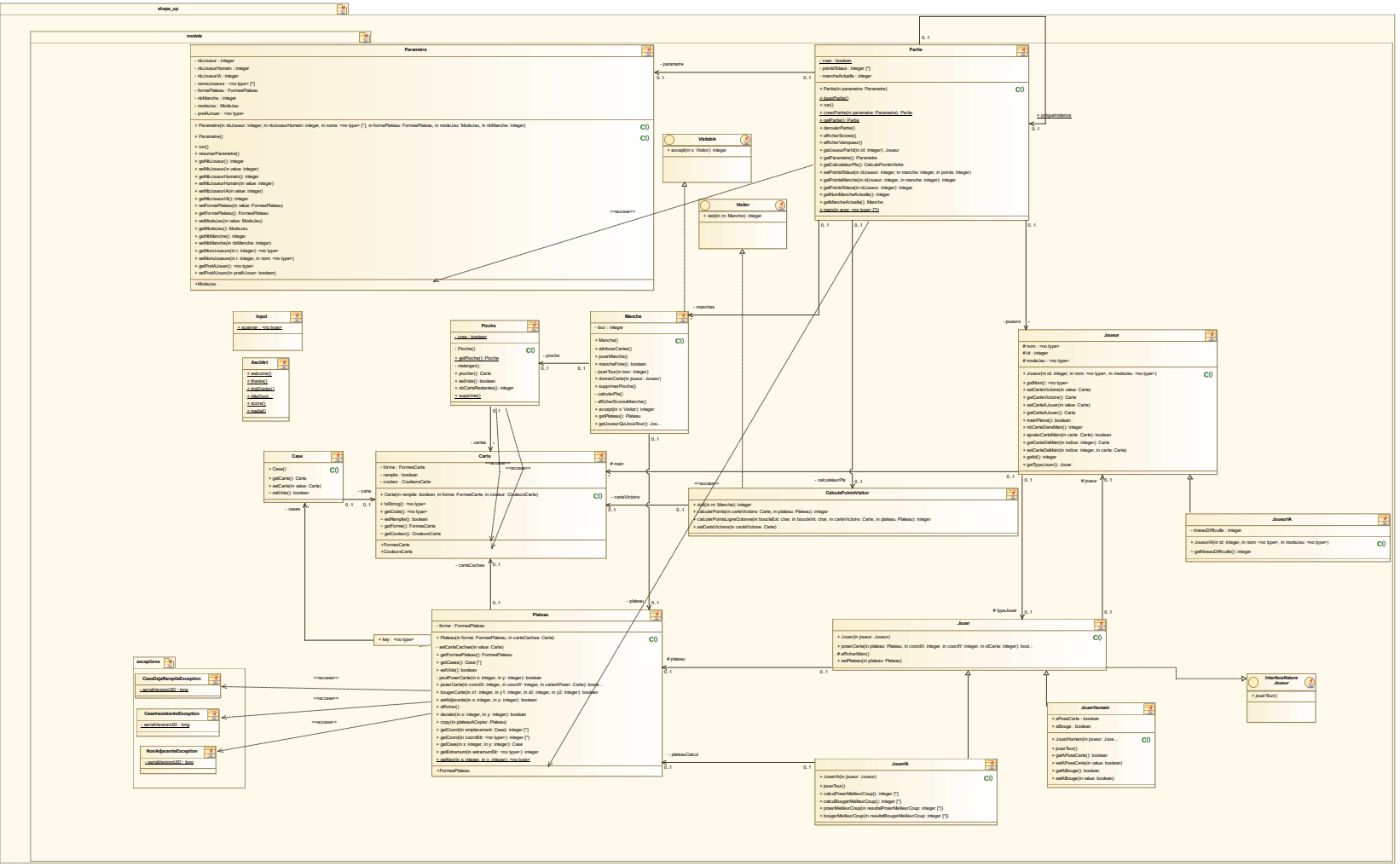
shape_zp
modele
Parametre
- nbJoueur : integer
- nbJoueurHumain : integer
- nbJoueurIA : integer
- noms<<no type>> [*]
- formePlateau : FormesPlateau
- nbManche : integer
- modeJeu : ModeJeu
- prêtAJouer : <<no type>>
+ Parametre(in nbJoueur: integer, in nbJoueurHumain: integer, in noms: <<no type>> [*], in formePlateau: FormesPlateau, in modeJeu: ModeJeu, in nbManche: integer)
+ Parametre()
+ run()
+ resumerParametre()
+ getNbJoueur(): integer
+ setNbJoueur(in value: integer)
+ getNbJoueurHumain(): integer
+ setNbJoueurHumain(in value: integer)
+ setNbJoueurIA(in value: integer)
+ getNbJoueurIA(): integer
+ setFormePlateau(in value: FormesPlateau)
+ getFormePlateau(): FormesPlateau
+ setModeJeu(in value: ModeJeu)
+ getModeJeu(): ModeJeu
+ getNbManche(): integer
+ setNbManche(in nbManche: integer)
+ getNomJoueurs(in i: integer): <<no type>>
+ setNomJoueurs(in i: integer, in nom: <<no type>>)
+ getPretAJouer(): <<no type>>
+ setPretAJouer(in pretAJouer: boolean)
+ModeJeu
Partie
- cree : boolean
- pointsTotaux : integer [*]
- mancheActuelle : integer
+ Partie(in parametre: Parametre)
+ joueParti()
+ run()
+ creerPartie(in parametre: Parametre): Partie
+ setPartie(): Partie
+ dereulerPartie()
+ afficherScores()
+ afficherVainqueur()
+ getJoueurParId(in id: integer): Joueur
+ getParametre(): Parametre
+ getCalculeurPts(): CalculePointsVisitor
+ setPointsTotaux(in idJoueur: integer, in manche: integer, in points: integer)
+ getPointsManche(in idJoueur: integer, in manche: integer): integer
+ getPointsTotaux(in idJoueur: integer): integer
+ getNumMancheActuelle(): integer
+ getMancheActuelle(): Manche
+ main(in args: <<no type>> [*])
Visitable
+ accept(in v: Visitor): integer
Visitor
+ visit(in m: Manche): integer
Input
+ scanner : <<no type>>
AsciiArt
+ welcome()
+ thanks()
+ logChaleur()
+ titleChoix()
+ score()
+ medal()
Pioche
- cree : boolean
+ Pioche()
+ getPioche(): Pioche
- melanger()
+ piocher(): Carte
+ estVide(): boolean
+ nbCarteRestantes(): integer
+ supprimer()
Manche
- tour : integer
+ Manche()
+ attribuerCartes()
+ jouerManche()
+ mancheFinie(): boolean
- jouerTour(in tour: integer)
+ donneeCarte(in joueur: Joueur)
+ supprimerPioche()
- calculePts()
- afficherScoresManche()
+ accept(in v: Visitor): integer
+ getPlateau(): Plateau
+ getJoueurQuiJoueTour(): Jou...
Joueur
nom : <<no type>>
id : integer
modeJeu : <<no type>>
+ Joueur(in id: integer, in nom: <<no type>>, in modeJeu: <<no type>>)
+ getNom(): <<no type>>
+ setCarteVictoire(in value: Carte)
+ getCarteVictoire(): Carte
+ setCarteAJouer(in value: Carte)
+ getCarteAJouer(): Carte
+ mainPleine(): boolean
+ nbCarteDansMain(): integer
+ ajouteCarteMain(in carte: Carte): boolean
+ getCarteDeMain(in indice: integer): Carte
+ setCarteDeMain(in indice: integer, in carte: Carte)
+ getId(): integer
+ getTypeJouer(): Jouer
Case
+ Case()
+ getCarte(): Carte
+ setCarte(in value: Carte)
+ estVide(): boolean
Carte
- forme : FormesCarte
- remplie : boolean
- couleur : CouleursCarte
+ Carte(in remplie: boolean, in forme: FormesCarte, in couleur: CouleursCarte)
+ toString(): <<no type>>
+ getCode(): <<no type>>
- estRemplie(): boolean
+ getForme(): FormesCarte
- getCouleur(): CouleursCarte
+FormesCarte
+CouleursCarte
CalculePointsVisitor
+ visit(in m: Manche): integer
+ calculePoints(in carteVictoire: Carte, in plateau: Plateau): integer
- calculePointsLigneColonne(in boucleEnt: char, in boucleInt: char, in carteVictoire: Carte, in plateau: Plateau): integer
+ setCarteVictoire(in carteVictoire: Carte)
JoueurIA
- niveauDifficulte : integer
+ JoueurIA(in id: integer, in nom: <<no type>>, in modeJeu: <<no type>>)
~ getNiveauDifficulte(): integer
Plateau
- forme : FormesPlateau
+ Plateau(in forme: FormesPlateau, in carteCachee: Carte)
- setCarteCachee(in value: Carte)
- getFormePlateau(): FormesPlateau
+ getCases(): Case [*]
+ estVide(): boolean
- peutPoserCarte(in x: integer, in y: integer): boolean
+ poserCarte(in coordX: integer, in coordY: integer, in carteAPoser: Carte): bool...
+ bougerCarte(in x1: integer, in y1: integer, in x2: integer, in y2: integer): boolean
+ estAdjacente(in x: integer, in y: integer): boolean
+ afficher()
+ decaler(in x: integer, in y: integer): boolean
+ copy(in plateauACopier: Plateau)
+ getCoord(in emplacement: Case): integer [*]
+ getCoordX(in coordStr: <<no type>>): integer [*]
+ getCase(in x: integer, in y: integer): Case
+ getExtremum(in extremumStr: <<no type>>): integer
+ getNext(in x: integer, in y: integer): <<no type>>
+FormesPlateau
Jouer
+ Jouer(in joueur: Joueur)
+ poserCarte(in plateau: Plateau, in coordX: integer, in coordY: integer, in idCarte: integer): bool...
# afficherMain()
+ setPlateau(in plateau: Plateau)
InterfaceNature Joueur
+ jouerTour()
JouerHumain
+ aPoseCarte : boolean
+ aBouge : boolean
+ JouerHumain(in joueur: Joue...
+ jouerTour()
+ getAPoseCarte(): boolean
+ setAPoseCarte(in value: boolean)
+ getABouge(): boolean
+ setABouge(in value: boolean)
JouerIA
+ JouerIA(in joueur: Joueur)
+ jouerTour()
+ calculPoserMeilleurCoup(): integer [*]
+ calculBougerMeilleurCoup(): integer [*]
+ poserMeilleurCoup(in resultatPoserMeilleurCoup: integer [*])
+ bougerMeilleurCoup(in resultatBougerMeilleurCoup: integer [*])
exceptions
CaseDejaRempliaException
- serialVersionUID : long
CaseInexistanteException
- serialVersionUID : long
NonAdjacenteException
- serialVersionUID : long
- parametre
- pioche
- cartes
- carte
- case
- carteCachee
- plateau
- plateauCalcul
- manches
- joueurs
- calculateurPts
- carteVictoire
- instanceUnique
main
joueur
typeJouer
plateau
<<access>>
+ key : <<no type>>