

<<enumeration>>	
Wizard	
YELLOW	
BLUE	
PINK	
GREEN	

<<enumeration>>	
Color	
Black	
White	
Grey	

<<enumeration>>	
Creature	
Yellow Gnomes	
Red Dragons	
Blue Uniforms	
Green Frogs	
Pink Fairies	

<<enumeration>>	
Name	
Herald: 3	
Knight: 2	
Centaur: 2	
Fungus: 3	
Isler: 1	
Thief: 3	
Monk: 1	
Herbalist: 2	
Magpostman: 1	
Princess: 2	

<<class>>	
Assistant	
+ value: Values	
+ Assistant () void	
<<enumeration>>	
Values	
Cheetha1:1	
Octopus2:1	
Cat3:2	
Lion4:3	
Octopus7:4	
Dog4:4	
Elephant5	
Turtle1:5	

<<class>>	
Entrance	
+ Entrance (in capacity: int) void	
<<class>>	
Dining Room	
+ diners: +map: Creature, Students[]	
+ getNumberOfCreatures (in creature: Creature) int	
+ DiningRoom () void	
+ addStudents (in student: Student)	

<<abstract class>>	
StudentContainer	
+ students: +Students[]	
+ addStudents (in newStudents: +Student[])	
+ removeStudents (in creature: Creature)	
+ addStudent (in student: Student)	
+ addStudent (in student: Student)	

<<class>>	
Student	
+ creature: Creature	
+ Student (in creature: Creature): Student	
<<class>>	
StudentBucket	
+ generated_students: Map: Creature, Integer	
+ max_students: int	

<<class>>	
Player	
+ username: String	
+ my_color: Color	
+ last_played_card: Assistant	
+ my_coins: int	
+ assistant: Professor Assistant[]	
+ professor: Professor (in creature: Creature)	
+ wizard: Wizard	
+ towers: int	
+ entrance: Entrance	
+ dining_room: DiningRoom	
+ Player (in username: String, in color: Color, in wizard: Wizard, in numberof towers: int, in entrance: Entrance) void	
+ addCoin () void	
+ removeCoins (in characterCost: int) void	
+ playAssistantCard (in assistant: Assistant) void	
+ changeNumberofTowers (in numberof towers: int)	

<<class>>	
Table	
+ islands: Island[]	
+ clouds: Cloud[]	
+ mother_nature: MotherNature	
+ coin_reserve: int	
+ islandFusion (in islands: Island[]) void	
+ table (in islands: Island[], in clouds: Cloud[]) in characters: Character[]	
+ removeCoin () void	

<<class>>	
Mother Nature	
+ curr_island: int	
+ postmanSteps: int	
+ getCurIsland () int	
+ setCurIsland (in curIsland: int)	
<<class>>	
Island	
+ towers: int	
+ colorofTowers: Color	
+ money: int	
+ changeTower (in color: Color)	
+ island (in students: Student[], in numberof towers: int, in colorof tower: Color) void	
+ addIsland () void	
+ removeIsland () void	
+ addStudent (in student: Student)	

<<class>>	
Cloud	
+ extracted_students: Student[]	
+ extractStudent () void	
+ Cloud (in max_students: int) void	

<<class>>	
Student	
+ creature: Creature	
+ Student (in creature: Creature): Student	

<<class>>	
StudentBucket	
+ generated_students: Map: Creature, Integer	
+ max_students: int	

<<class>>	
ConcreteCharacterCreator	
+ createCharacter (in name: Name): Character	

<<class>>	
CharacterCreator	
+ createCharacter (in name: Name): Character	

<<class>>	
GameModel	
+ players: Player[]	
+ table: Table	
+ numberofPlayers: int	
+ characters: Character[]	
+ played_character: Character	
+ currentPlayer: Player	
+ updatedCharacterCost: int	
+ GameModel (in advanced_rules: boolean, in players: String[]) in numberofPlayer: int void	
+ fillClouds () void	
+ playAssistant (in indexofAssistant: int) void	
+ establishIslandOrder () void	
+ moveMotherNature (in jumps: int) boolean	
+ findWinner () Player	
+ checkProfessor () void	
+ evaluateTurn () void	
+ playCharacter (in characterIndex: int) void	
+ checkNeighboringIsland () void	

<<interface>>	
Playable	
+ addNewEntry (in indexofIsland: int) void	
+ evaluateInfluence () void	
+ setStartPositionMovements (in numberofSteps: int)	
+ theEffect (in creature: Creature)	
+ moveStudents (in source: StudentContainer, in destination: StudentContainer, in creature: Creature)	
+ setInfluenceCharacter (in typeofInfluenceCharacter: int)	

<<class>>	
MoverCharacter	
+ effect () void	

<<class>>	
Herald	
+ name: Name	
+ model: Playable	
+ effect () void	

<<class>>	
Herbalist	
+ destructor: int	
+ name: Name	
+ model: Playable	
+ effect () void	

<<class>>	
Postman	
+ name: Name	
+ model: Playable	
+ effect () void	

<<class>>	
Thief	
+ name: Name	
+ model: Playable	
+ effect () void	

<<class>>	
InfluenceCharacter	
+ model: Playable	
+ name: Name	
+ effect: void	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	

<<class>>	
Character	
+ effect () void	
+ getName () Name	