Card + static std::vector < std::string > ColorCode # std::string name # std::vector< int > manaCost # std::string color # bool isDiscarded # bool is Engaged # std::string typeOfCard + Card(std::string nm, std::vector< int > mnCt, std::string clr) + void setName(const std::string &name) + void engage() + virtual void disengage() + void setManaCost(std ::vector< int > manaCost) + void setColor(const std::string &color) void setIsDiscarded (bool isDiscarded) + const std::string & getName() const + const std::string getColored Name() const + std::vector< int > getManaCost() const + const std::string & getColor() const bool getIsDiscarded () const + bool getIsEngaged() const + ~Card() + void isStillOperational() + virtual void print() + virtual void printLine (int line) + std::string manaToString() + static void print(std ::vector< Card * > v) **EnchantmentCard** # Card * asso LandCard + EnchantmentCard(std ::string name, std::string color) + ~EnchantmentCard() + LandCard(std::string + void effect() nm, std::string clr) + ~LandCard() + void print() + void printLine(int + void printLine(int line) line) + void associate(Card *CardToAssociate) + Card * getAsso()

CreatureCard

std::vector< std::string

+ CreatureCard(std::

string nam, std::vector

< int > mC, std::vector < std::string > capa, std

::string cl, int ap, int hp)

+ std::vector< std::string

> getCapacities() const

+ int getHp() const

+ void setHp(int hp)

(int attackPower)

+ bool isFirstTurn()
+ std::string capacitiesTo

+ void printLine(int

+ void disengage()+ void resetHp()+ bool hasCapacity(std ::string capacity)

+ void setAttackPower

+ void setFirstTurn(int b)

() const

String()

line)

+ int getAttackPower

int hp # int defaultHp # int attackPower # bool firstTurn

> capacities