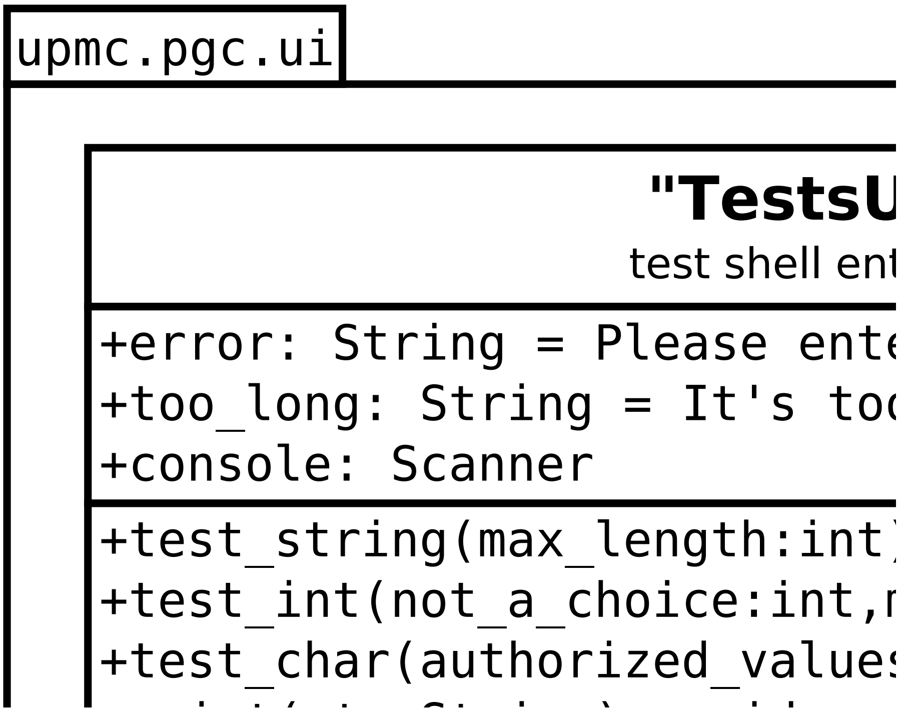


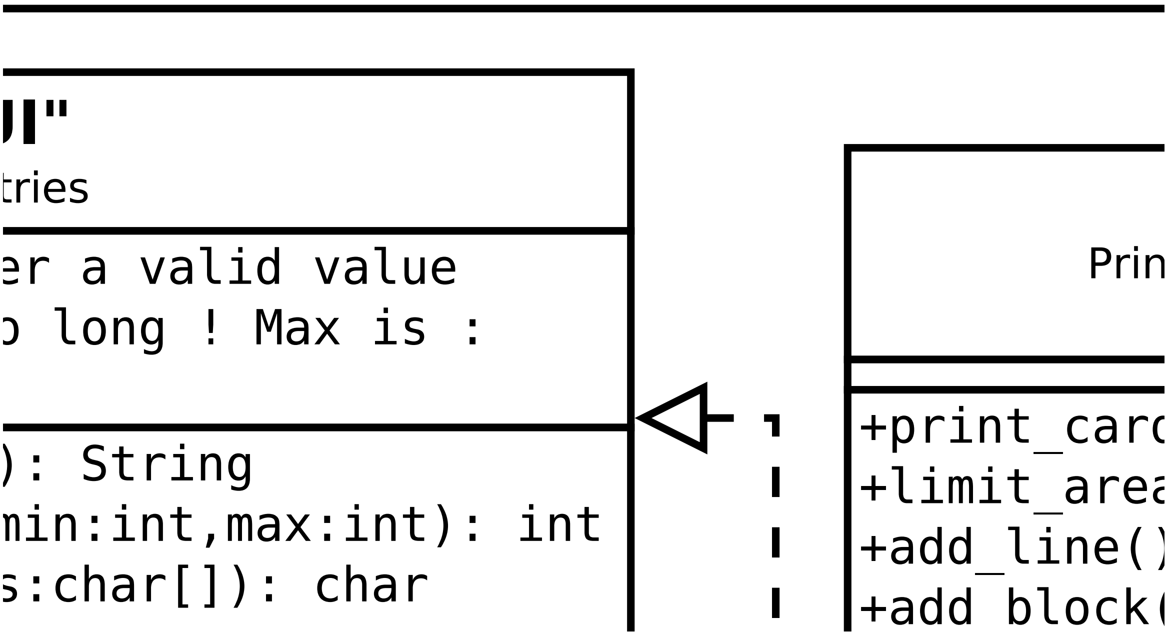
upmc.pgc.ui

"TestsU

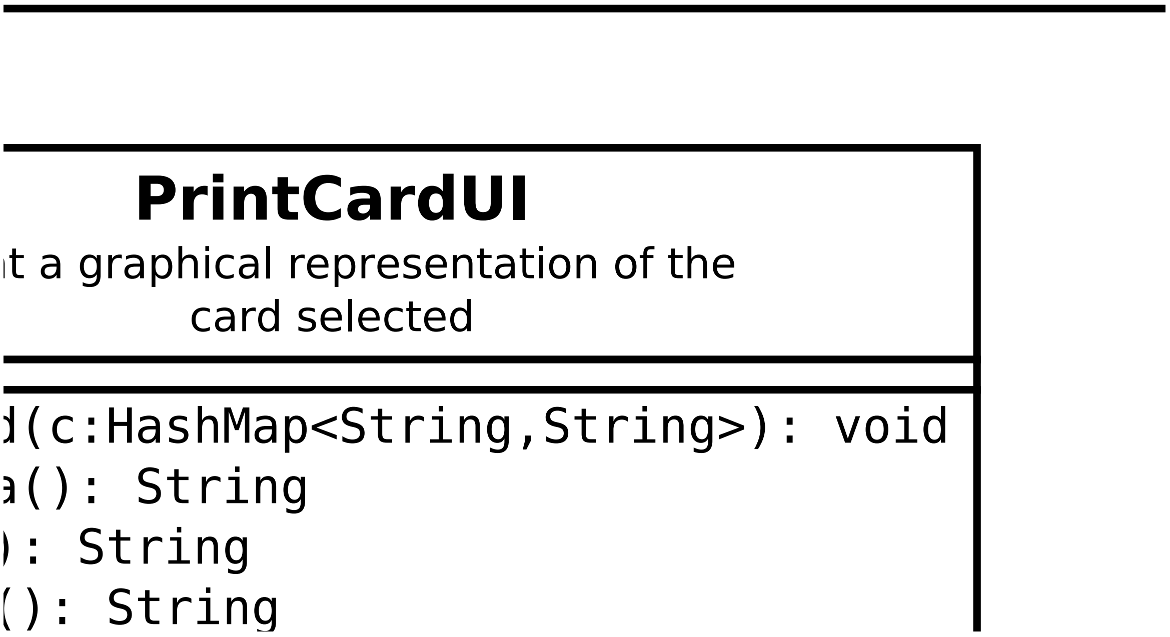
test shell ent

+error: String = Please ente
+too_long: String = It's too
+console: Scanner

+test_string(max_length:int)
+test_int(not_a_choice:int,r
+test_char(authorized_values



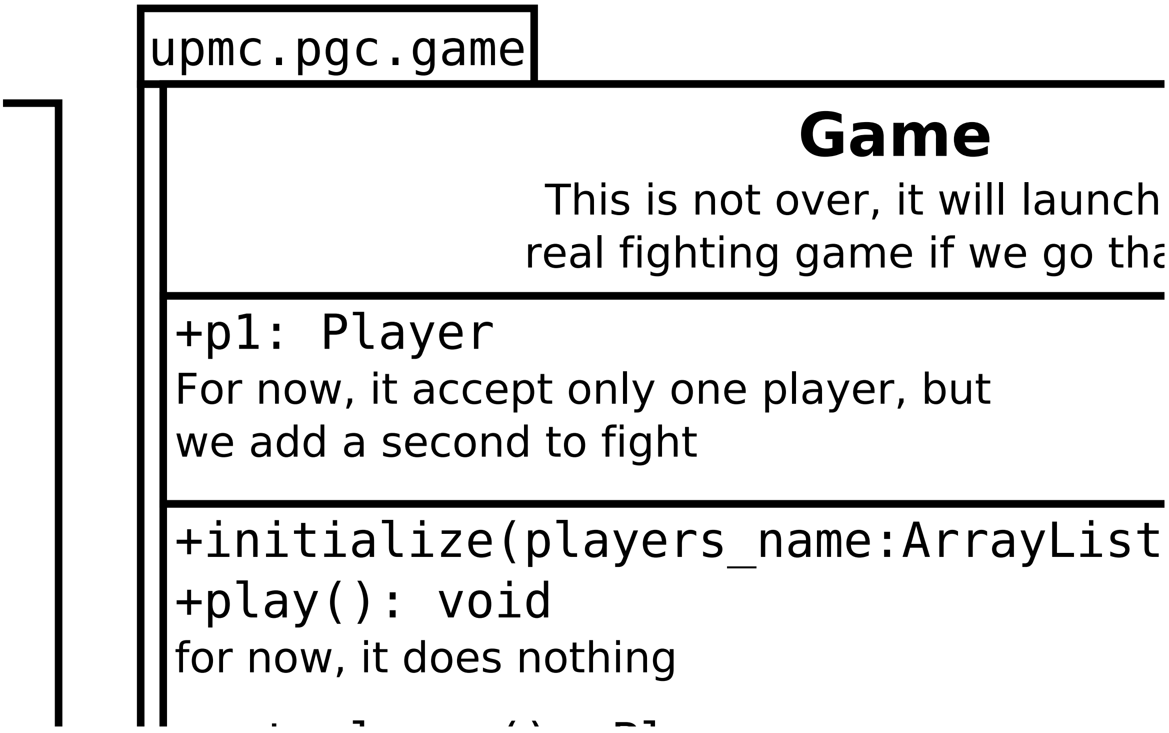
ui"
tries
er a valid value
o long ! Max is :
): String
min:int,max:int): int
s:char[]): char
Prin
+print_card
+limit_area
+add_line()
+add_block(



PrintCardUI

nt a graphical representation of the
card selected

```
d(c:HashMap<String,String>): void
a(): String
): String
(): String
```



upmc.pgc.game

Game

This is not over, it will launch
real fighting game if we go tha

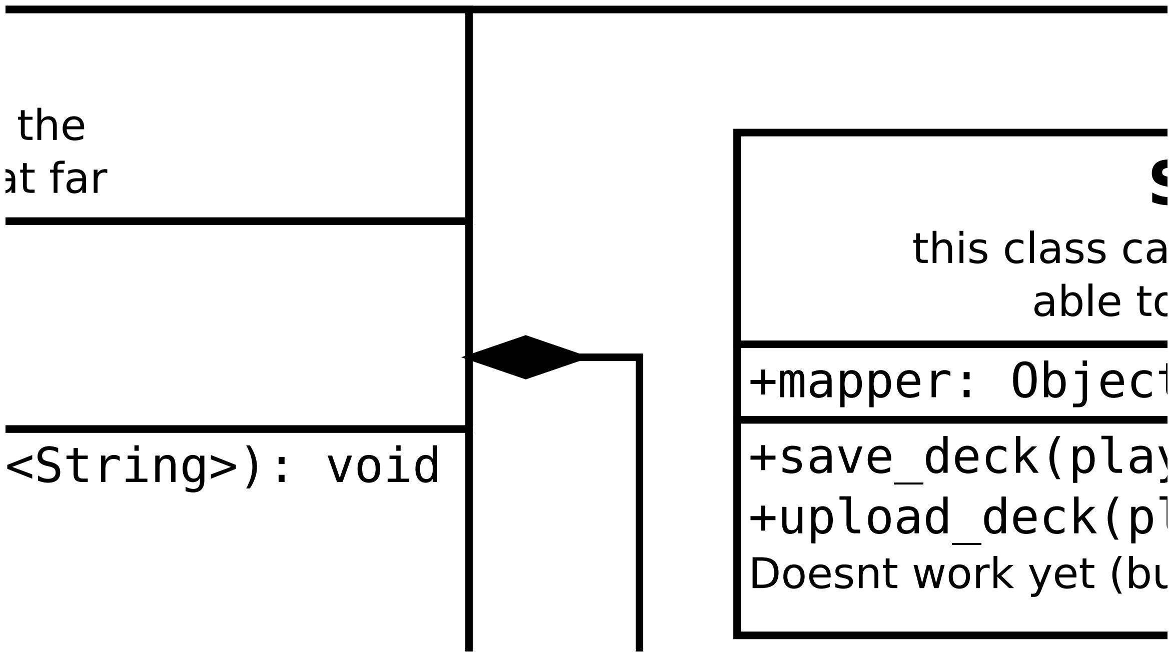
+p1: Player

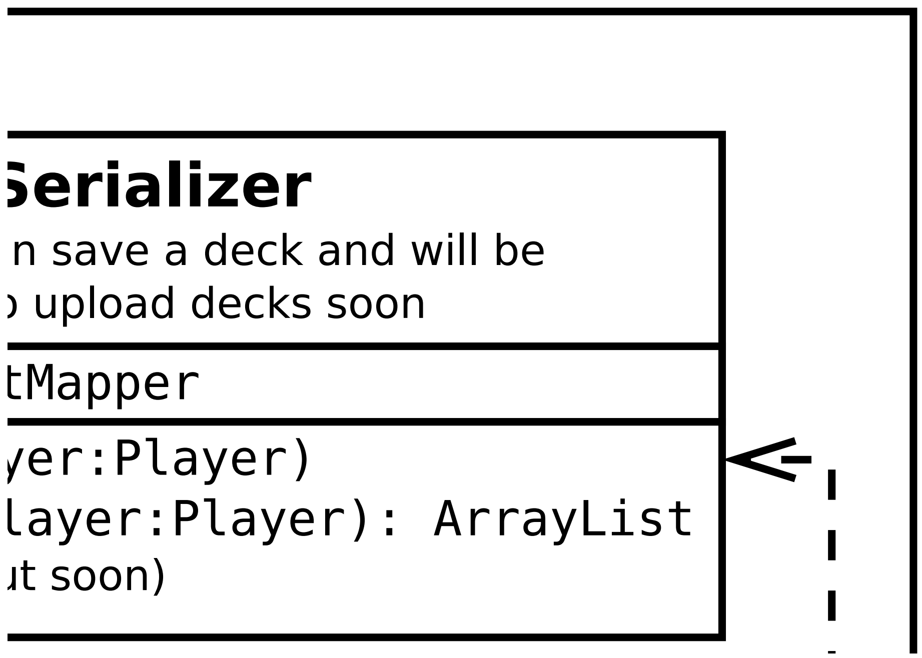
For now, it accept only one player, but
we add a second to fight

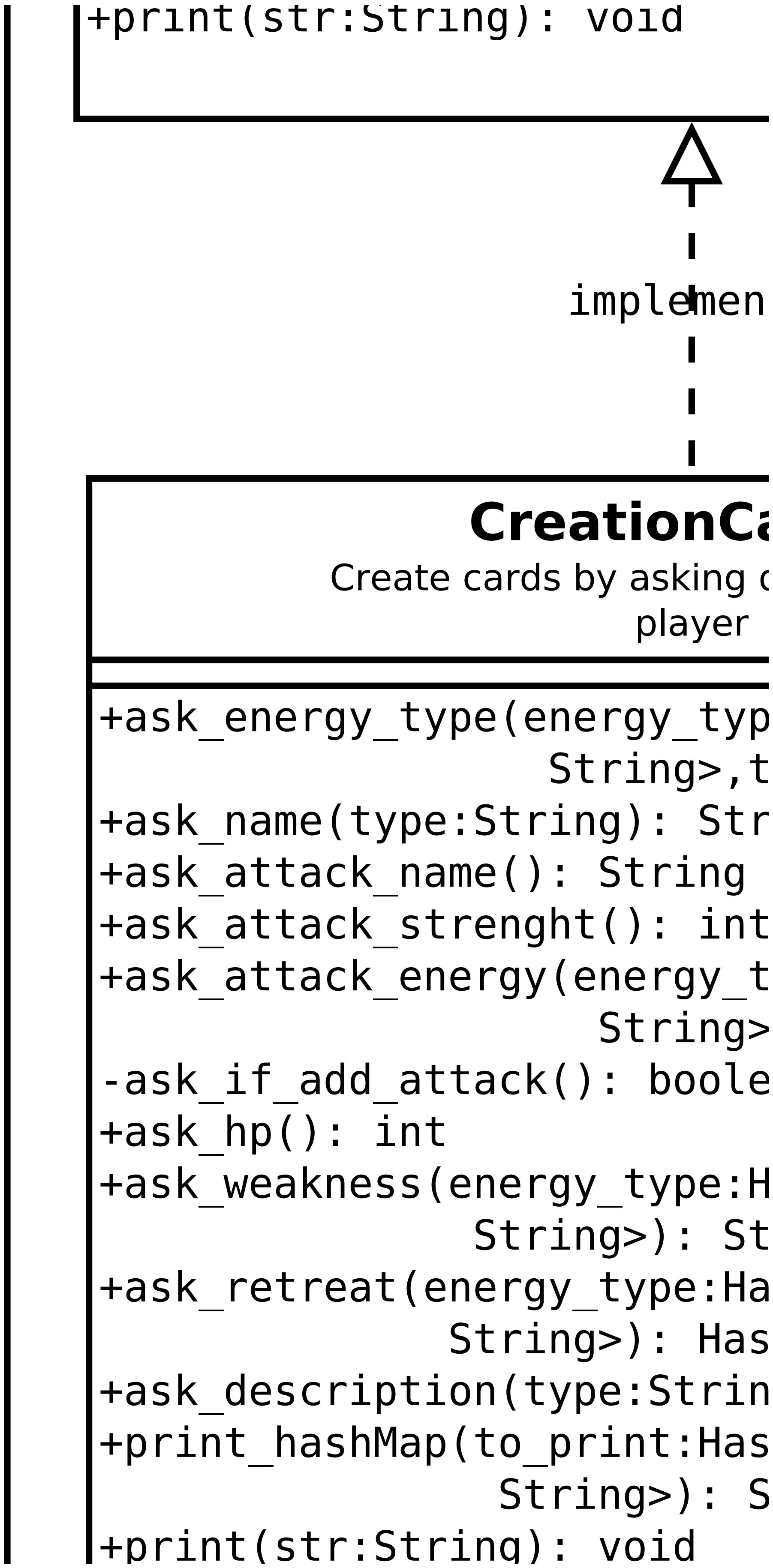
+initialize(players_name:ArrayList

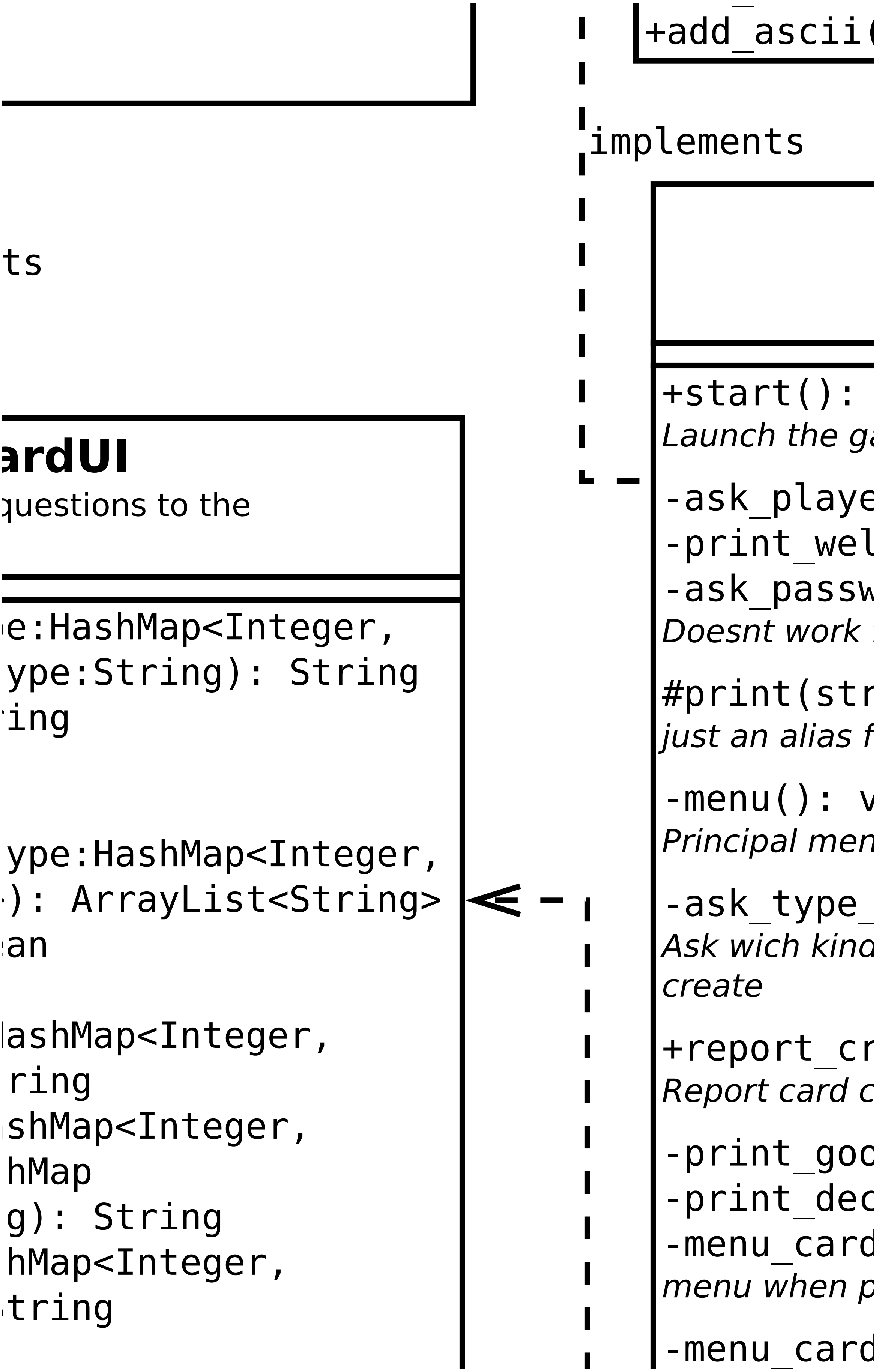
+play(): void

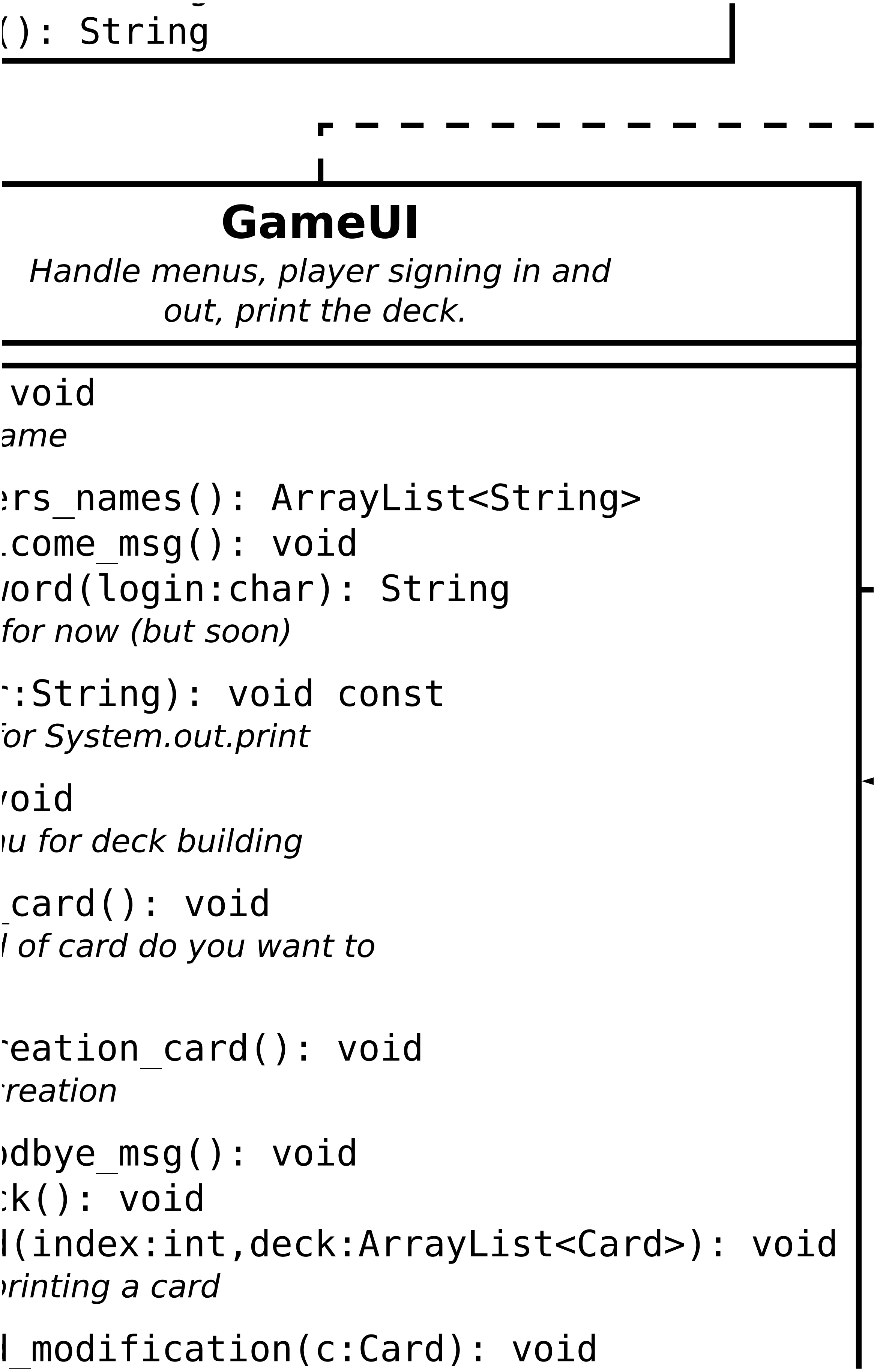
for now, it does nothing


the
at far
<String>): void
S
this class ca
able to
+mapper: Object
+save_deck(play
+upload_deck(pl
Doesnt work yet (bu


Serializer
n save a deck and will be
o upload decks soon
tMapper
yer:Player)
layer:Player): ArrayList
ut soon)


+print(str:String): void
implemen
CreationCa
Create cards by asking c
player
+ask_energy_type(energy_typ
String>,t
+ask_name(type:String): Str
+ask_attack_name(): String
+ask_attack_strenght(): int
+ask_attack_energy(energy_t
String>
-ask_if_add_attack(): boole
+ask_hp(): int
+ask_weakness(energy_type:H
String>): St
+ask_retreat(energy_type:Ha
String>): Has
+ask_description(type:Strin
+print_hashMap(to_print:Has
String>): S
+print(str:String): void


+add_ascii(
implements
ts
+start():
Launch the ga
-ask_playe
-print_wel
-ask_passw
Doesnt work
#print(str
just an alias f
-menu(): v
Principal men
-ask_type_
Ask wich kind
create
+report_cr
Report card c
-print_goo
-print_dec
-menu_card
menu when p
-menu_card
ardUI
questions to the
e:HashMap<Integer,
ype:String): String
ing
ype:HashMap<Integer,
): ArrayList<String>
an
lashMap<Integer,
ring
shMap<Integer,
hMap
g): String
hMap<Integer,
tring



(): String

GameUI

Handle menus, player signing in and out, print the deck.

```
 void
```
ame

```
ers_names(): ArrayList<String>
.come_msg(): void
vord(login:char): String
```
for now (but soon)

```
:String): void const
```
or System.out.print

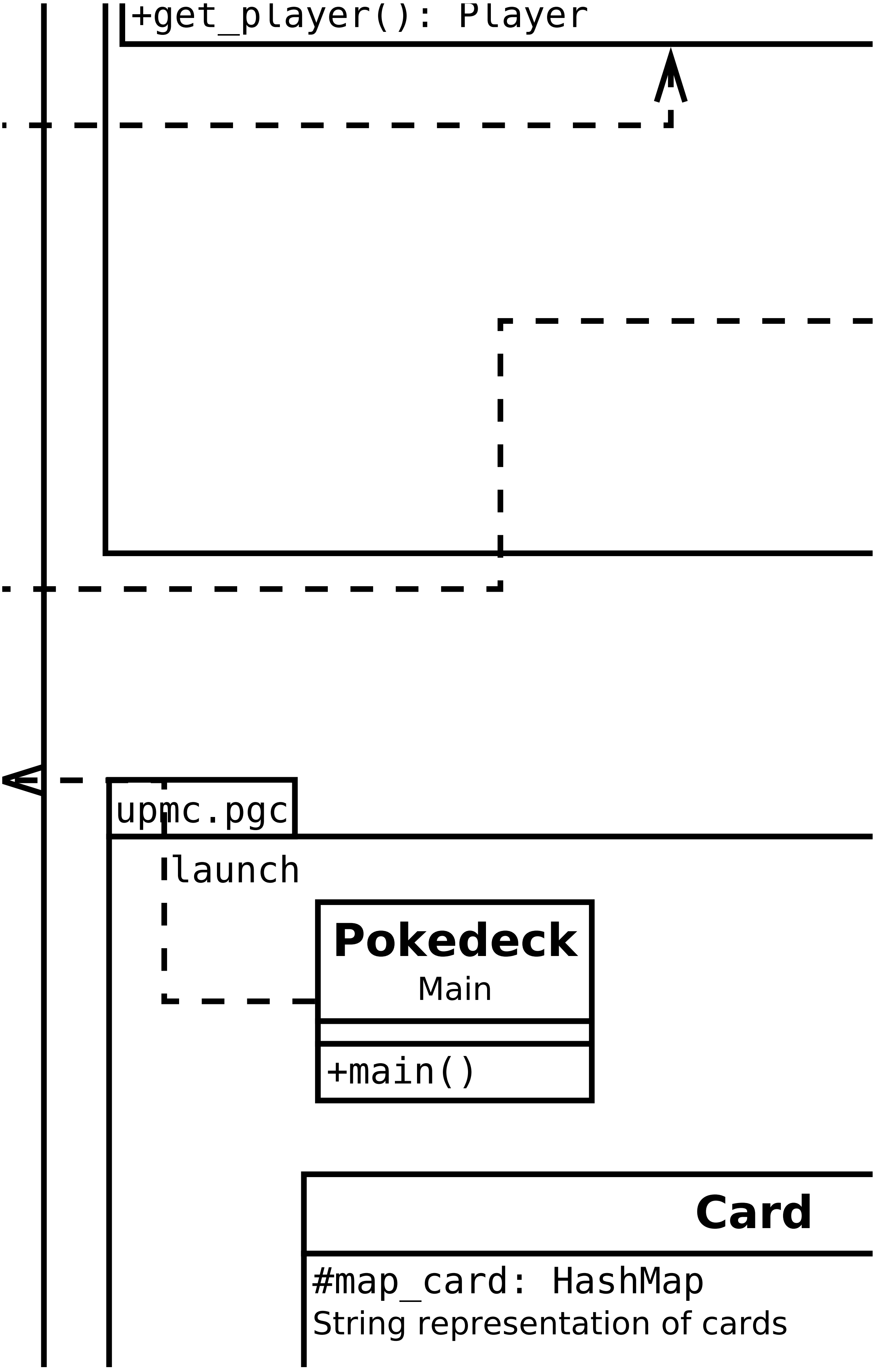
```
void
```
u for deck building

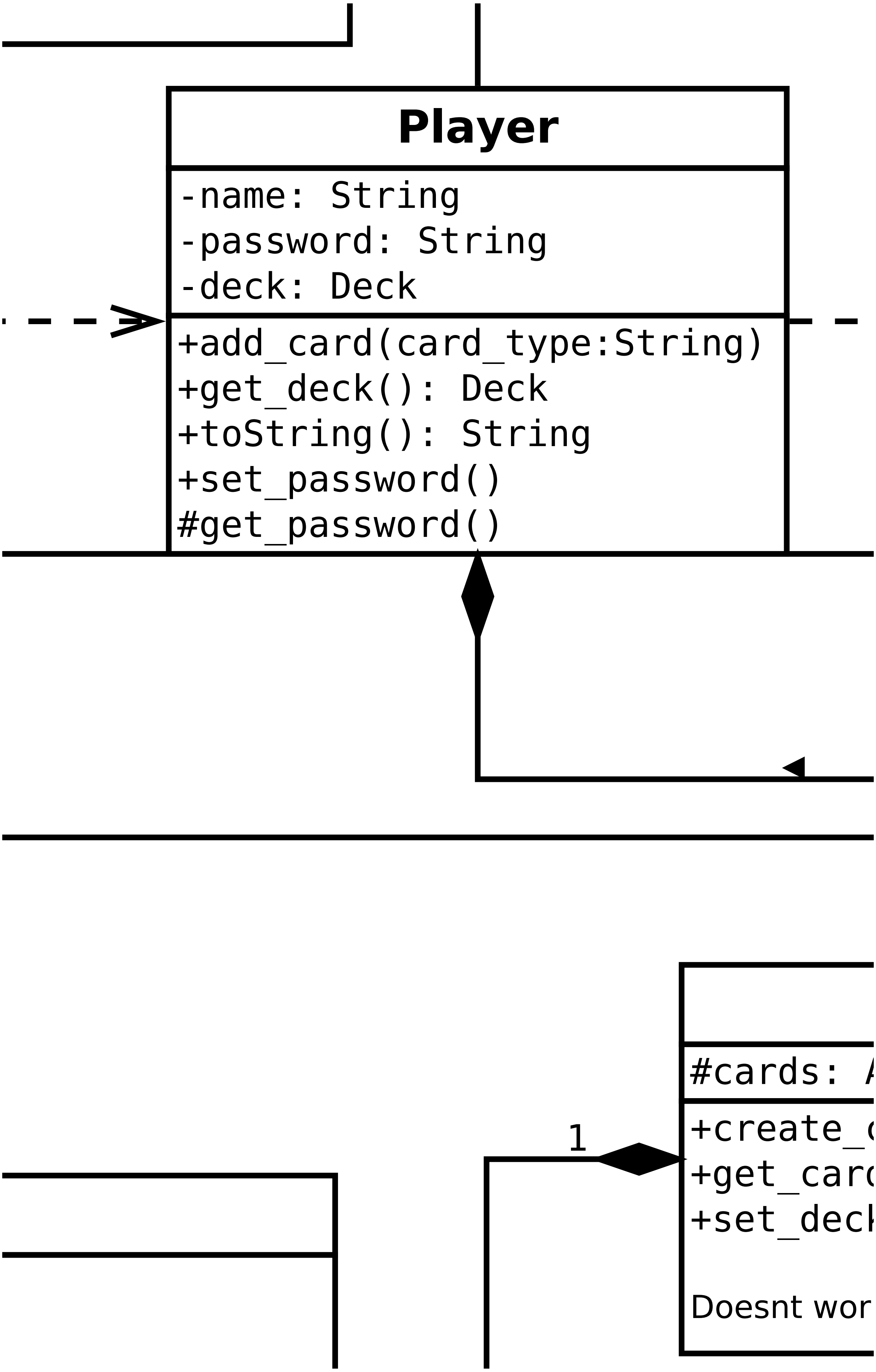
```
_card(): void
```
l of card do you want to

```
reation_card(): void
```
reation

```
odbye_msg(): void
ck(): void
l(index:int,deck:ArrayList<Card>): void
```
printing a card

```
l_modification(c:Card): void
```


+get_player(): Player
upmc.pgc
launch
Pokedeck
Main
+main()
Card
#map_card: HashMap
String representation of cards



Player

-name: String
-password: String
-deck: Deck

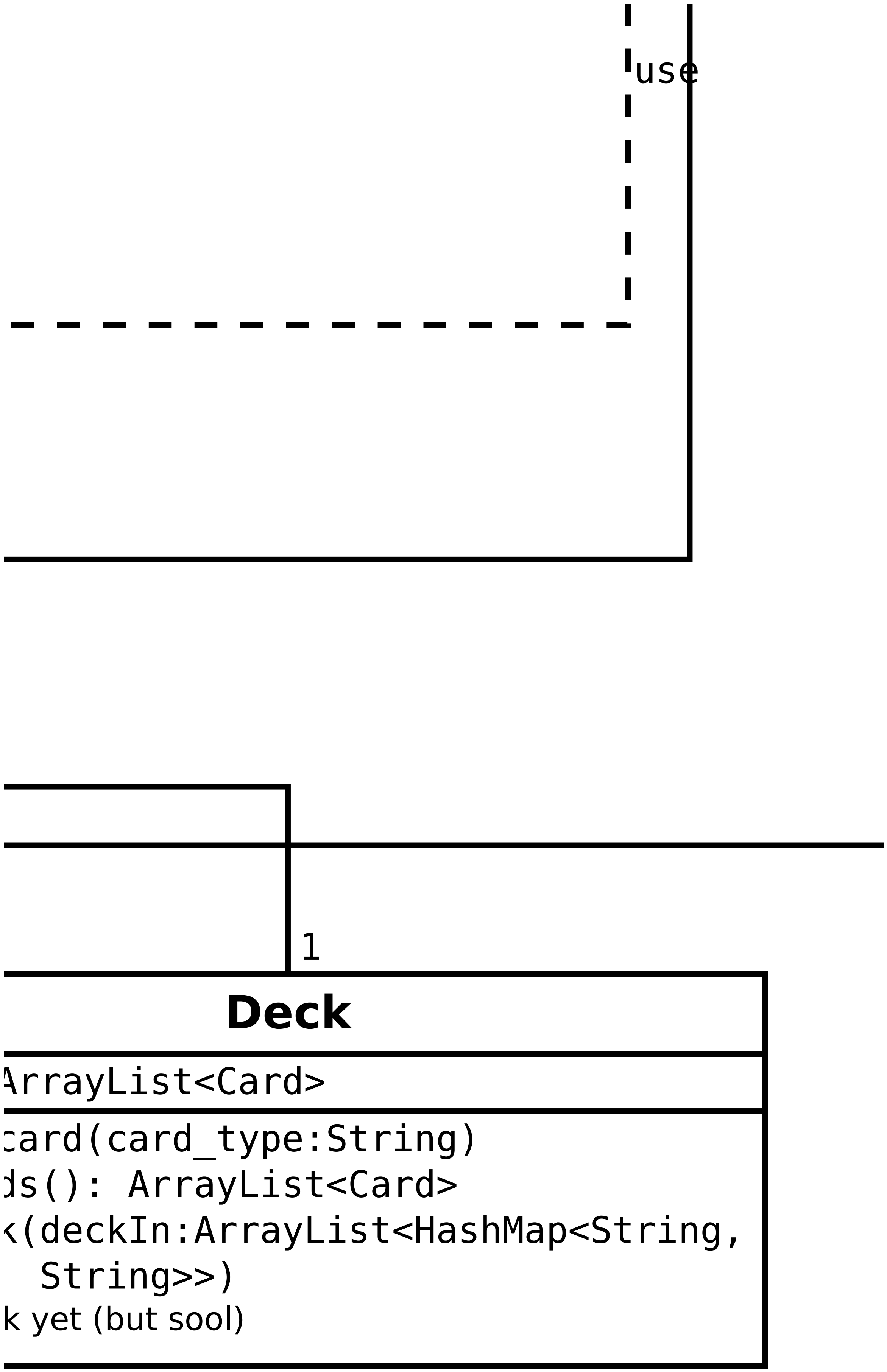
+add_card(card_type:String)
+get_deck(): Deck
+toString(): String
+set_password()
#get_password()

1

#cards: A

+create_c
+get_card
+set_deck

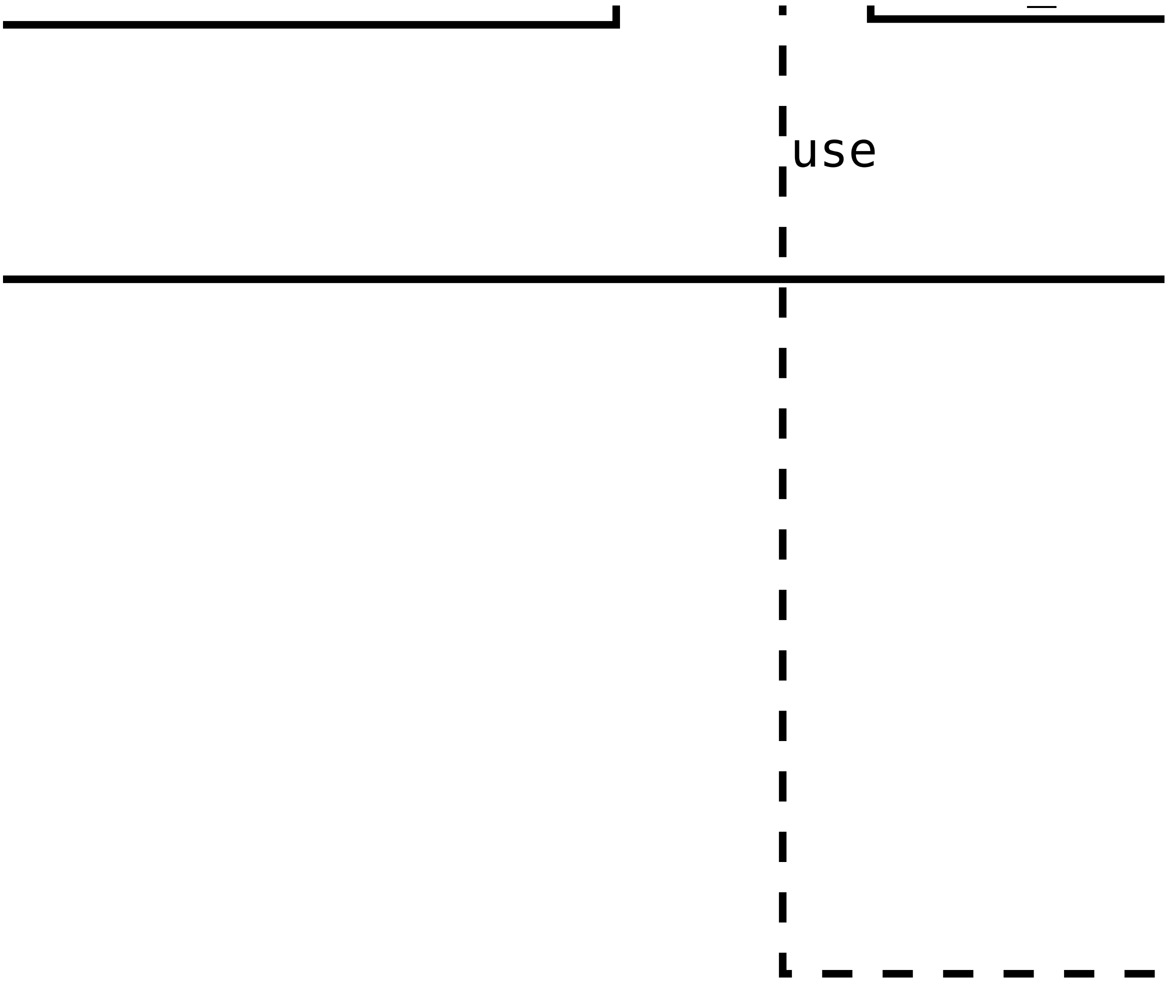
Doesnt wor

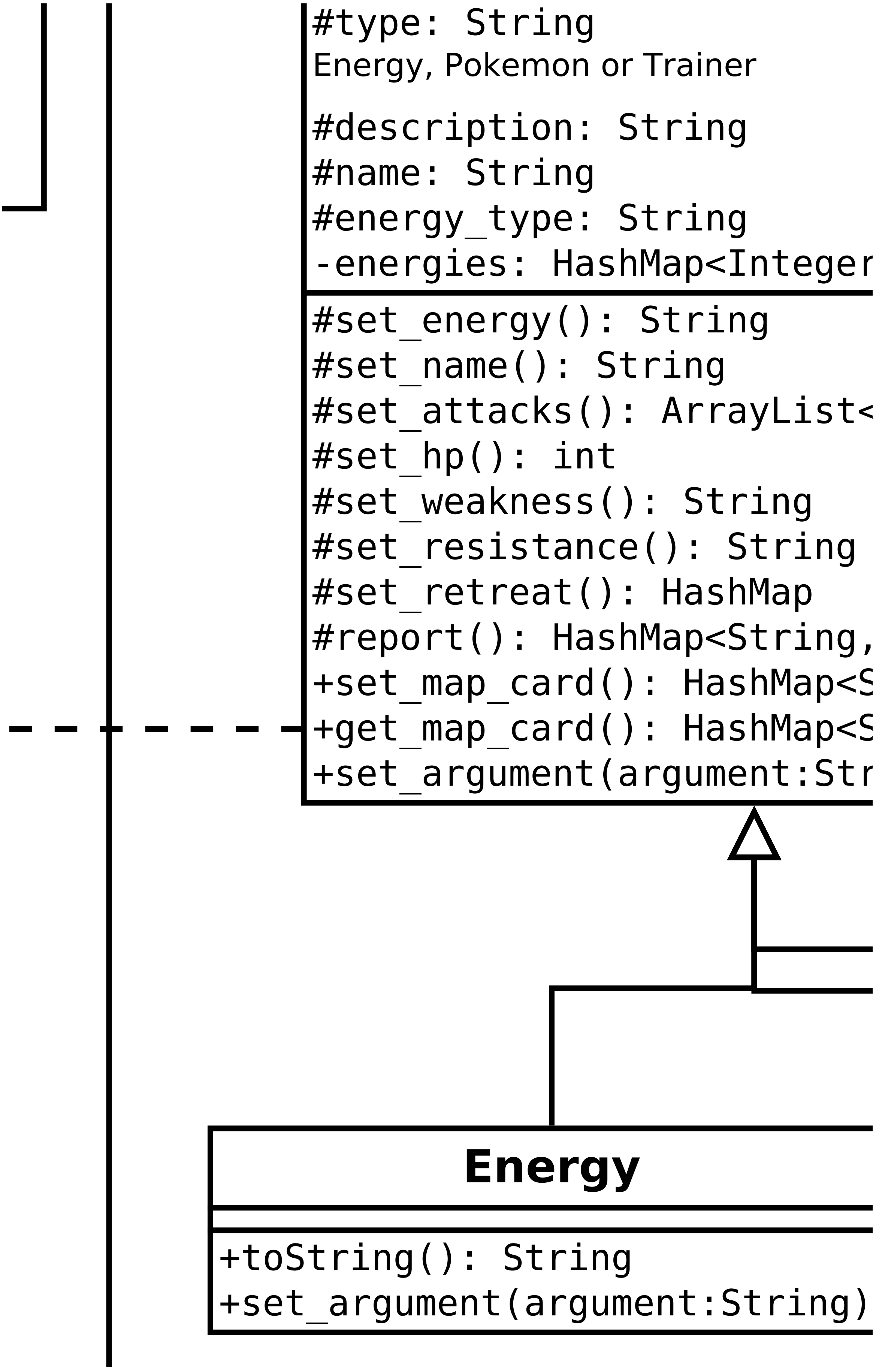


Deck

```
ArrayList<Card>
card(card_type:String)
ds(): ArrayList<Card>
k(deckIn:ArrayList<HashMap<String,
  String>>)
```

k yet (but sool)


use



#type: String

Energy, Pokemon or Trainer

#description: String

#name: String

#energy_type: String

-energies: HashMap<Integer

#set_energy(): String

#set_name(): String

#set_attacks(): ArrayList<

#set_hp(): int

#set_weakness(): String

#set_resistance(): String

#set_retreat(): HashMap

#report(): HashMap<String,

+set_map_card(): HashMap<S

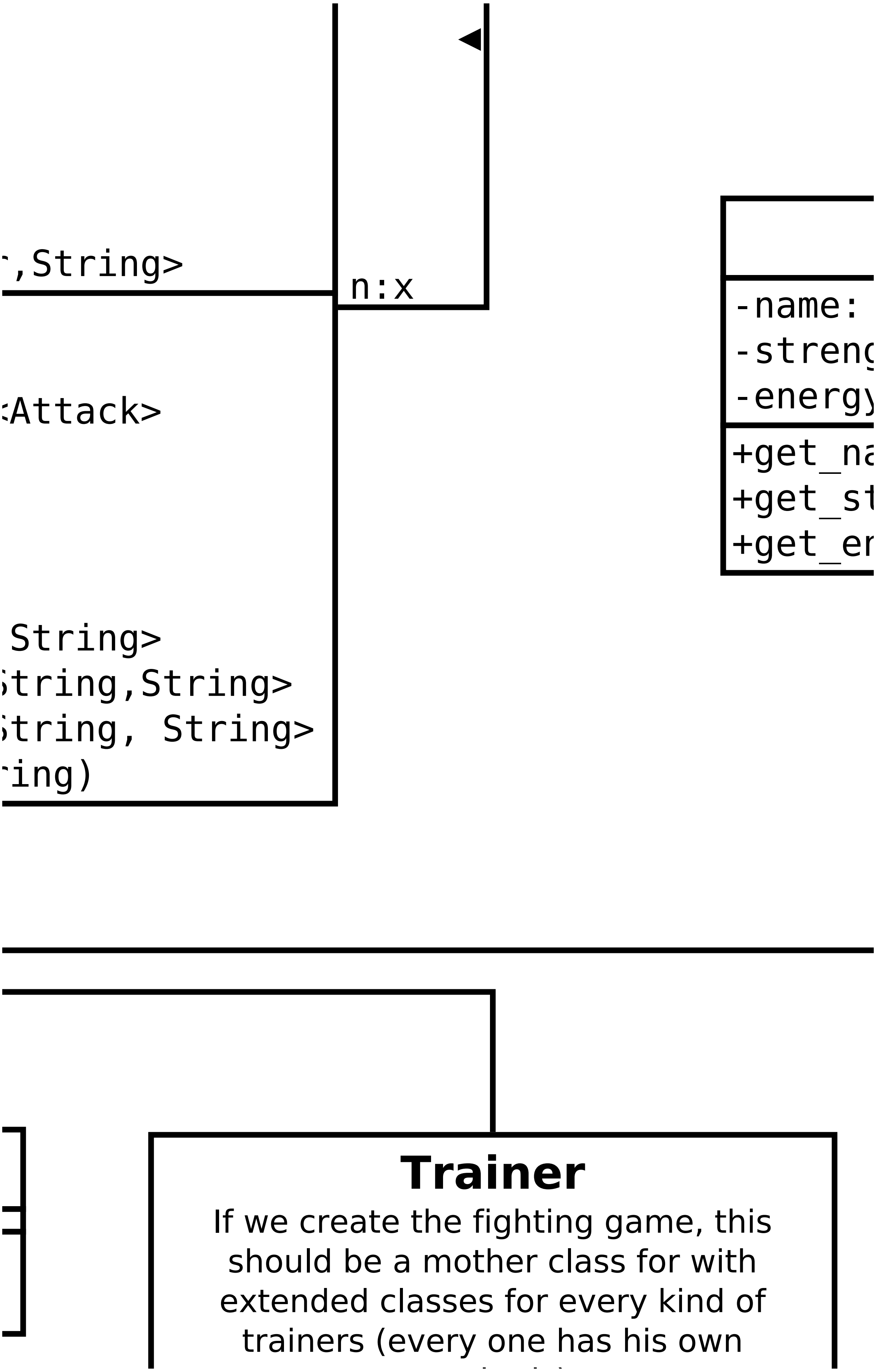
+get_map_card(): HashMap<S

+set_argument(argument:Str

Energy

+toString(): String

+set_argument(argument:String)



,String>

n:x

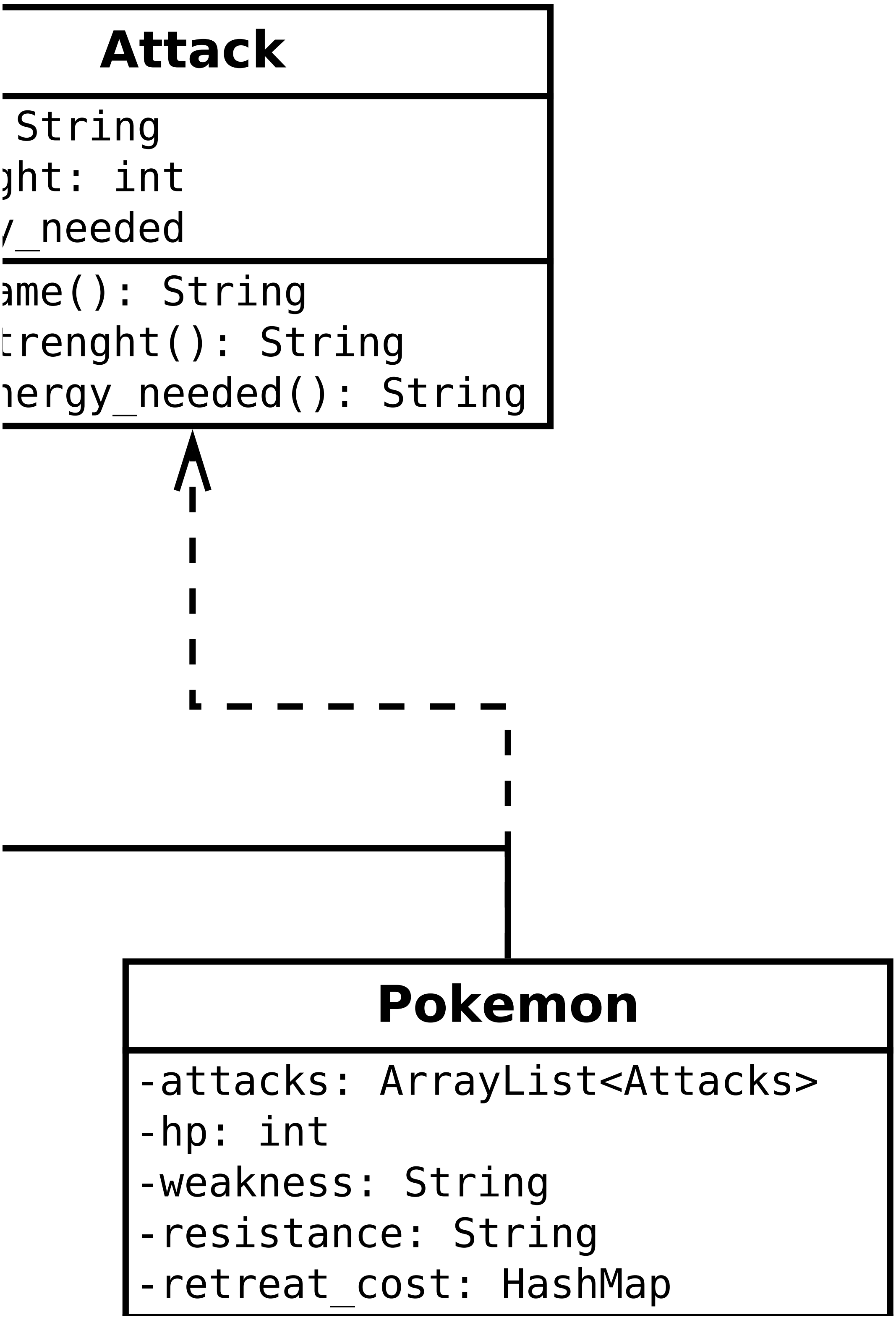
<Attack>

String>
String,String>
String, String>
ring)

-name:
-streng
-energy
+get_na
+get_st
+get_en

Trainer

If we create the fighting game, this should be a mother class for with extended classes for every kind of trainers (every one has his own


Attack
String
ght: int
y_needed
ame(): String
trenght(): String
nergy_needed(): String
Pokemon
-attacks: ArrayList<Attacks>
-hp: int
-weakness: String
-resistance: String
-retreat_cost: HashMap