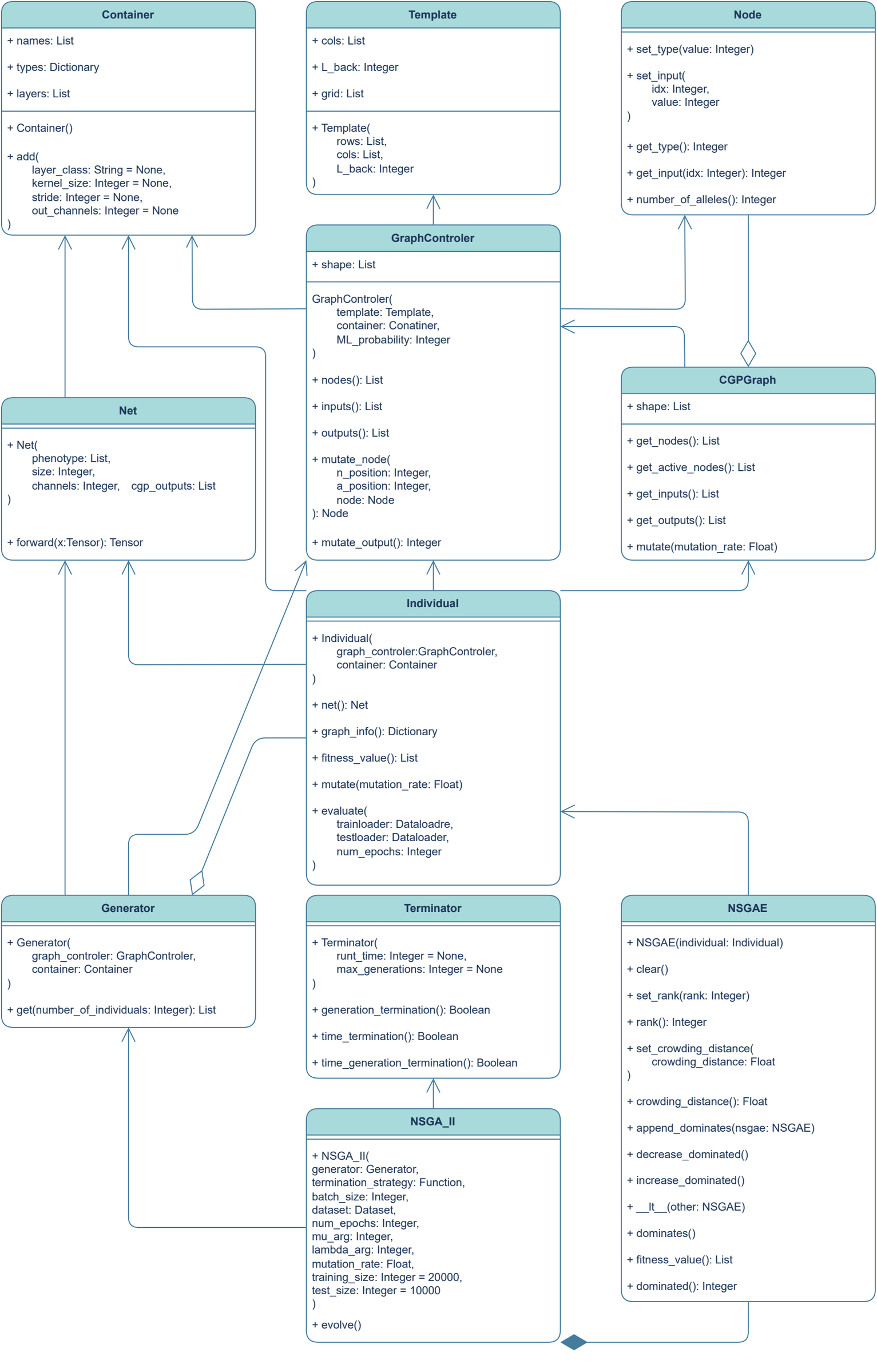
Container
+ names: List
+ types: Dictionary
+ layers: List
+ Container()
+ add(
layer_class: String = None,
kernel_size: Integer = None,
stride: Integer = None,
out_channels: Integer = None
)
Template
+ cols: List
+ L_back: Integer
+ grid: List
+ Template(
rows: List,
cols: List,
L_back: Integer
)
Node
+ set_type(value: Integer)
+ set_input(
idx: Integer,
value: Integer
)
+ get_type(): Integer
+ get_input(idx: Integer): Integer
+ number_of_alleles(): Integer
GraphControler
+ shape: List
GraphControler(
template: Template,
container: Conatiner,
ML_probability: Integer
)
+ nodes(): List
+ inputs(): List
+ outputs(): List
+ mutate_node(
n_position: Integer,
a_position: Integer,
node: Node
): Node
+ mutate_output(): Integer
CGPGraph
+ shape: List
+ get_nodes(): List
+ get_active_nodes(): List
+ get_inputs(): List
+ get_outputs(): List
+ mutate(mutation_rate: Float)
Net
+ Net(
phenotype: List,
size: Integer,
channels: Integer, cgp_outputs: List
)
+ forward(x:Tensor): Tensor
Individual
+ Individual(
graph_controler:GraphControler,
container: Container
)
+ net(): Net
+ graph_info(): Dictionary
+ fitness_value(): List
+ mutate(mutation_rate: Float)
+ evaluate(
trainloader: Dataloadre,
testloader: Dataloader,
num_epochs: Integer
)
Generator
+ Generator(
graph_controler: GraphControler,
container: Container
)
+ get(number_of_individuals: Integer): List
Terminator
+ Terminator(
runt_time: Integer = None,
max_generations: Integer = None
)
+ generation_termination(): Boolean
+ time_termination(): Boolean
+ time_generation_termination(): Boolean
NSGAE
+ NSGAE(individual: Individual)
+ clear()
+ set_rank(rank: Integer)
+ rank(): Integer
+ set_crowding_distance(
crowding_distance: Float
)
+ crowding_distance(): Float
+ append_dominates(nsgae: NSGAE)
+ decrease_dominated()
+ increase_dominated()
+ __lt__(other: NSGAE)
+ dominates()
+ fitness_value(): List
+ dominated(): Integer
NSGA_II
+ NSGA_II(
generator: Generator,
termination_strategy: Function,
batch_size: Integer,
dataset: Dataset,
num_epochs: Integer,
mu_arg: Integer,
lambda_arg: Integer,
mutation_rate: Float,
training_size: Integer = 20000,
test_size: Integer = 10000
)
+ evolve()