Kaan Karataş
150200081

GPU

- cuda: CUDA

+ GPU(int)
+ execute(string): string
+ printLog(): void

Computer

<<pointer>>
<<get/set>> - mp_attachedCPU: CPU
<<get/set>> - mp_attachedGPU: GPU

+ Computer()
+ operator+(CPU): void
+ operator+(GPU): void
+ execute(string): void
+ printLog(): void

CPU

- alu: ALU

+ CPU(int)
+ execute(string): string
+ getInput(int, int): void
+ printLog(): void

0..1   *
attachedGPU

*   0..1
attachedCPU

cuda

1

alu

1

CUDA

- numCores: int

+ CUDA(int)
+ render(): string
+ trainModel(): string
+ printLog(): void

ALU

- numPerCores: int

+ ALU(int)
+ add(int, int): int
+ subtract(int, int): int
+ multiply(int, int): int
+ printLog(): void

Computer class holds CPU and GPU object pointers. CPU and GPU do not get initialized with computer, they rather get get attached with the + operator. Because of the specified situations, their lifetimes are not connected thus there is an aggregationrelationship between Computer and CPU, GPU. However, as CUDA and ALU objects of GPU and CPU get initialized with corresponding constructors of CPU/GPU and get destructed whith the corresponding destructors of CPU/GPU, their life cycles are related. ALU and CUDA are member variables of CPU/GPU. Hence, there is a Composition relationship between CPU-ALU and GPU-CUDA. Inside operator+ methods of Computer, there is a parameter visibility between Computer and CPU/GPU. All other visibilities are attribute visibility.