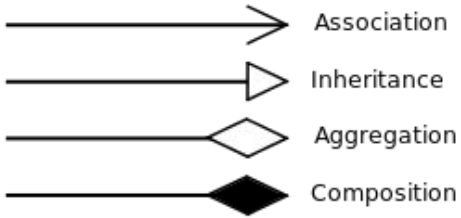
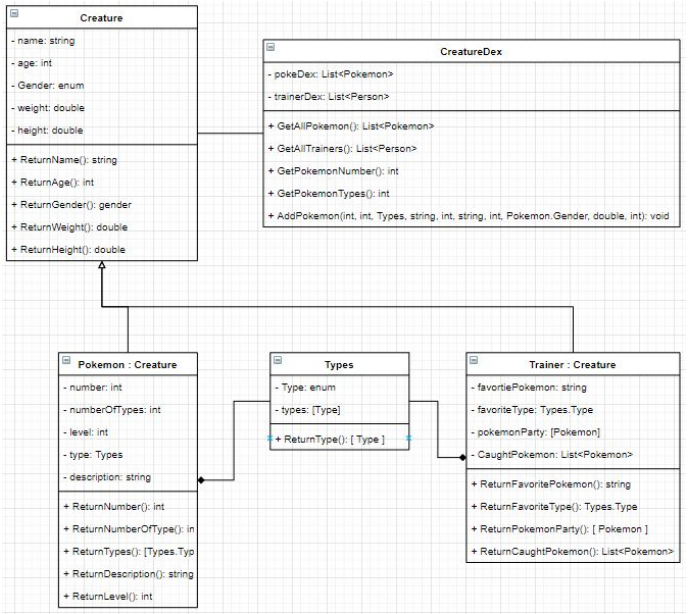
UML Class Diagrams, Delegates & Lambda

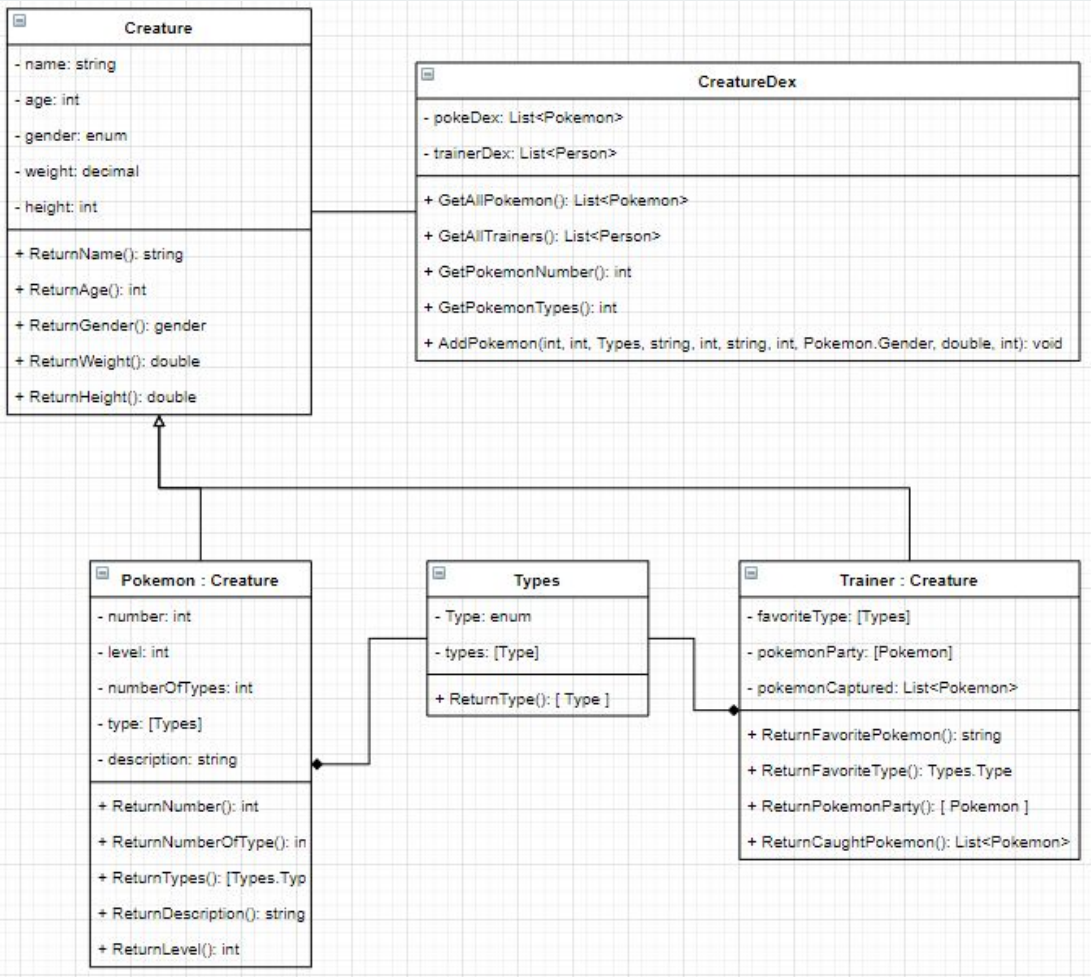
By Kasper og René

UML Class Diagrams(): Diagram

- Use cases for UML
- Content of Class Diagrams
- Access Modifiers
- Relations






Creature
- name: string
- age: int
- gender: enum
- weight: decimal
- height: int
+ ReturnName(): string
+ ReturnAge(): int
+ ReturnGender(): gender
+ ReturnWeight(): double
+ ReturnHeight(): double
CreatureDex
- pokeDex: List<Pokemon>
- trainerDex: List<Person>
+ GetAllPokemon(): List<Pokemon>
+ GetAllTrainers(): List<Person>
+ GetPokemonNumber(): int
+ GetPokemonTypes(): int
+ AddPokemon(int, int, Types, string, int, string, int, Pokemon.Gender, double, int): void
Pokemon : Creature
- number: int
- level: int
- numberOfTypes: int
- type: [Types]
- description: string
+ ReturnNumber(): int
+ ReturnNumberOfType(): in
+ ReturnTypes(): [Types.Typ
+ ReturnDescription(): string
+ ReturnLevel(): int
Types
- Type: enum
- types: [Type]
+ ReturnType(): [Type]
Trainer : Creature
- favoriteType: [Types]
- pokemonParty: [Pokemon]
- pokemonCaptured: List<Pokemon>
+ ReturnFavoritePokemon(): string
+ ReturnFavoriteType(): Types.Type
+ ReturnPokemonParty(): [Pokemon]
+ ReturnCaughtPokemon(): List<Pokemon>

Delegates();

Simpel forklaring af Delegates

```
1 reference
static string isFavouritePokemon(string name, string name2)
{
    if (name == name2)
    {
        return "WOW!";
    }
    else
    {
        return "Nope";
    }
}
```

Eksempel

```
Trainer Ash = new Trainer("Pikachu", Types.Type.Dark, new Pokemon[6], new List<Pokemon>(), "Ash", 10, Creatures.Gend
stringDel example21;
stringDel example22;

example21 = Ash.returnFavouritePokemon;
example22 = creaturedex.getAllPokemon()[0].returnName;

Console.WriteLine(isFavouritePokemon(example21(), example22()));
```

Delegates.Delegate2();

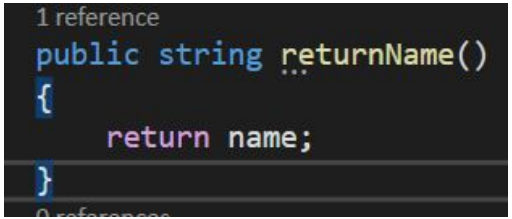
```
doubleDel example31;
doubleDel example32;

example31 = creaturedex.getAllPokemon()[0].returnHeight;
example32 = creaturedex.getAllPokemon()[0].returnWeight;

Console.WriteLine(weightPerKilo(example31(), example32()));
```

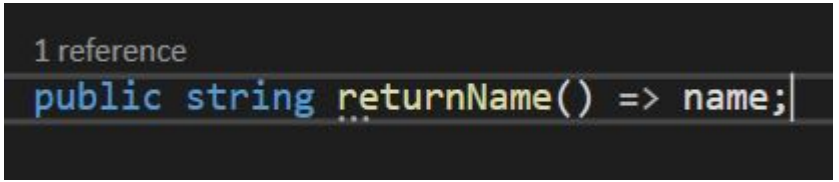
```
1 reference
static double weightPerKilo(double height, double weight)
{
    double result = height / weight;
    return result;
}
```

Lambda () => Lambda

Annonyme methods

Syntactic sugar





Projekt : Nedarvning

Syntax

Variabler

Nedarvning og access modifiers

```
namespace Fremvisning
{
    public delegate string stringDel();
    public delegate double doubleDel();
    8 references
    public class Creatures
    {
        protected string name;
        protected int age;
        protected Gender gender;
        protected double weight;
        protected double height;

        0 references
        public Creatures(string name, int age, Gender gender, double weight, double height)
        {
            this.name = name;
            this.age = age;
            this.gender = gender;
            this.weight = weight;
            this.height = height;
        }
```

```
namespace Fremvisning
{
    17 references
    public class Pokemon : Creatures
    {
        protected int number;
        protected int numberOfTypes;
        protected Types types;
        protected string description;
        protected int level;

        1 reference
        public Pokemon(int number, int numberOfTypes, Types types, string description, int level, string name, int age, Gender gender, double
        {
            this.number = number;
            this.numberOfTypes = numberOfTypes;
            this.types = types;
            this.description = description;
            this.level = level;
            this.name = name;
            this.age = age;
            this.gender = gender;
            this.weight = weight;
            this.height = height;
        }
```