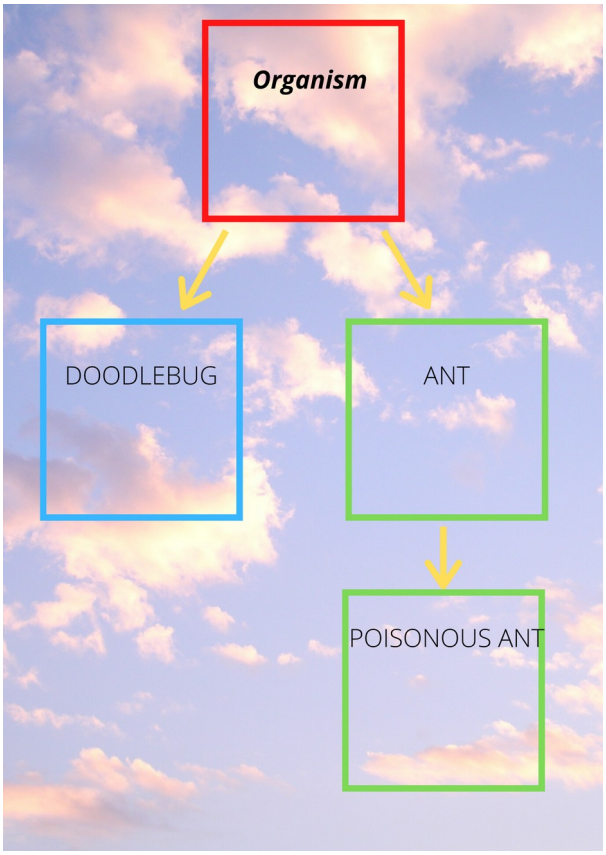
1) GENERAL STRUCTURE OF THE PROJECT

Organism is abstract class which has pure virtual functions. This pure virtual functions are implemented in derived classes. Each derived class implements this pure virtual functions differently.

a) Global Constant Variables

```
const int WORLD_SIZE = 20;

const int INITAL_ANT_NUMBER = 100;

const int INITIAL_DOODLEBUG_COUNT = 5;

how many count after doodlebug going to die from poison
const int STEP_COUNT_WITH_POISON = 2;

how many count after doodlebug going to die
const int STEP_COUNT_WITH_HUNGER = 3;

percantage of the becoming poisionus
const int CHANCE_OF_BECOMING_POISONOUS = 2;

how many count does ant need to wait
const int STEP_COUNT_TO_WAIT_BREED_ANT = 3;

how many count does dodlebug need to wait
const int STEP_COUNT_TO_WAIT_BREED_DOODLE_BUG = 3;

how many count does poisonous ant need to wait
const int STEP_COUNT_TO_WAIT_BREED_POISONOUS_ANT = 4;
```



b) Global Enums

// type of the organism
```
enum OrganismType{DOODLEBUG = 0,ANT = 1,POISONOUSANT = 2};
```

//shows direction
```
enum Direction{RIGHT = 0,LEFT,UP,DOWN};
```

2) BASE CLASS - ORGANİSM

a) Variables

i)Coordinates

```
int x;
int y;
```

This variables shows the position of the cell

ii) type

```
OrganismType type
```

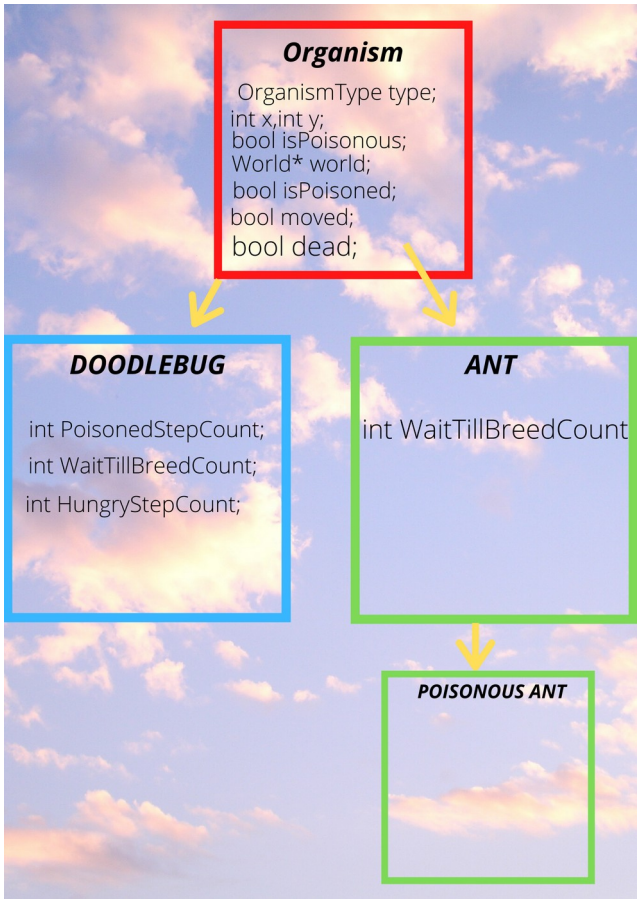
Shows type of the organism
organismType topic are held in enums.

iii) word pointer

```
World* world;

This data type allows the accessing the other cell
which is in the world class
```

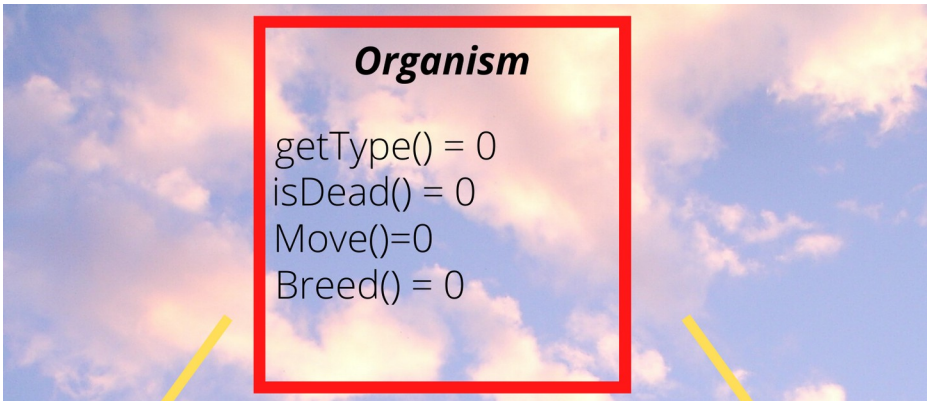
iiii) boolean variables

```
bool isPoisonous;
shows organism is poisonous or not
```



```
bool isPoisoned;
shows organism is poisoned or not

bool moved;
shows organism is moved or not

bool dead;
shows organism is dead or not
```

b) Pure virtual functions




```
getter of type
virtual OrganismType getType() const = 0;

moves the organism
virtual void Move()   =0;

breeds organism
virtual void Breed()  =0;

getter of dead
bool virtual isDead() =0;
```

c) Other functions

```
this function is used for choosing correct direction
returns  Direction(enum)

Direction DirectionDecision()


function that shows current coordinates is in area or not
takes coordinates as parameter
bool in_Border(int x, int y)
```

d) Constructor

```
Organism(World* theWorld, int x, int y)

takes world pointer as argument to determine which world is going to have this cell

takes the coordinates to determine the adress of the cell in area
```

e) virtual destructor

```
virtual ~Organism() { }
```

to free allocated space in memory

3) DERİVED CLASS -DOODLEBUGS

a) Variables

This variable shows steps that could’nt eat ant

```
int HungryStepCount;
```

This variable shows steps that could’nt breed
```
int WaitTillBreedCount;
```

This variable shows that how many step pasted after became poisioned
```
int PoisonedStepCount;
```

b) implemeting pure virtual functions

i) Move

This function is also used for eating ants . Firstly doodlebug search an ant . If there is an ant around the doodllebug it eats, if not exist doodlebug makes a random move .

ii) Breed

If doodlebug watis enough to breed it searchs for an empty cell. If it find an empty cell it breeds if it could’nt find it waits for next step.

ii) IsDead
shows the doodlebug is dead or not. Poison death or starve may cause death.
Couldnt eating an ant `STEP_COUNT_WITH_HUNGER` steps and
`STEP_COUNT_WITH_POISON` steps after eating poisonous ant is the reasons of death

iii) getType

returns the type of the organism

3) DERİVED CLASS -ANTS

a) Variables

This variable shows steps that could’nt breed
```
int WaitTillBreedCount;
```

b) implemeting pure virtual functions

i) Move

Firstly an ant search an empty cell . If exist ant makes a random move . If nor exist wait for next step

ii) Breed

If ant watis enough to breed it searchs for an empty cell. If it find an empty cell it breeds if it could’nt find it waits for next step.

iii) getType

returns the type of the organism

4) DERİVED CLASS – POISONOUS ANTS

This class is a derived class of ant.
It has poison.
It poisons doodlebugs

5) CLASS – WORLD

this class is about gameplay. This class has member functions like display,clear,create,createworld,

a) Variables

```
Organism *area[20][20];
```

it is a 2D-array that has Organism *