

**Министерство образования Республики Беларусь**

**Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»**

---

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
И УПРАВЛЕНИЯ**

**Кафедра интеллектуальных информационных технологий**

**Отчет по лабораторной работе №2**

**Проектирование программного обеспечения интеллектуальных  
систем**

Выполнил: Стронгин А.В., 121701

Проверил: Никифоров С. А.

**Минск 2022**

**Цель:**

Подробно изучить объектно-ориентированное программирование.

**Задание:**

Прямоугольное поле произвольного размера представляет мир, в котором живут обитатели океана. Прямоугольное поле разбито на равные клетки, и в каждой клетке может находиться максимум четверо обитателей. Моделирование жизни в таком мире происходит в пошаговом режиме, при котором каждый шаг содержит следующие четко выделенные стадии: передвижение, питание, размножение, умирание от старости или от голода.

Способы поведения обитателей в каждой из стадий будут зависеть от типа обитателя. Студенту предлагается два основных типа обитателей со следующими свойствами:

**Планктон:**

- в клетке может находиться только одно растение;
- передвигается;
- имеет пункты жизни (как в ролевых играх):
  - \* теряет некоторое количество пунктов жизни каждый ход (эффект умирания);
  - \* если не остается пунктов жизни, то планктон умирает;
- не питается;
- размножается в текущую клетку или на соседние клетки:
  - \* не нужна парная особь для размножения;
  - \* планктон может размножаться на соседнюю клетку даже, если там уже есть другой планктон с неполным процентом жизни (в этом случае размножение обновит жизнь до 100

**Хищник:**

- передвигается (количество клеток, на которое может передвинуться, зависит от вида обитателя);
- имеет размер (зависит от вида обитателя):
  - \* более крупному хищнику надо больше есть;
  - \* более крупный хищник может съесть более крупное травоядное;
- имеет срок жизни;

- может умереть, если голодно определенное количество ходов;
- питается планктоном или другим хищником (зависит от гастрономических пристрастий обитателя океана):
  - \* жертва должна находиться на той же клетке;
  - \* всегда съедает целого обитателя, если он не убежал;
- размножение:
  - \* имеет пол;
  - \* нужна парная особь для размножения на той же клетке;
  - \* детеныш рождается сразу на ту же клетку.

Изучив и уточнив приведенные правила, студенту необходимо разработать объектную модель предметной области и программу, которая будет осуществлять моделирование процесса жизни. Причем обитателями мира будут конкретные виды обитателей. Например, акула – хищник среднего размера, который может передвигаться на 2 клетки за ход, не питается планктоном и китами. Как видно из примера, конкретный вид обитателя может иметь какие-то специальные качества и какое-то специальное поведение. К планктону это не относится, потому что в модели должен быть только один вид планктона. При разработке иерархии классов обязательно должен быть применен такой атрибут любого объектно-ориентированного языка, как наследование.

Разработанная объектная модель должна соответствовать следующим требованиям:

- включать классы видов обитателей океана (количество на усмотрение разработчика)
- классы обитателей должны составлять иерархию, чтобы общие свойства и логика работы описывалась надклассом
- включать классы описания океанического мира (игрового поля)
- предусматривать механизм моделирования океанического мира
- включать классы, которые обеспечивают разгрузку модели из текстового файла

Реализованная на основе разработанной модели программа должна:

- выводить в консоль состояние игрового поля на каждом ходу.

### **Ход выполнения:**

Реализован абстрактный класс Organism, в котором определены виртуальные функции (type, superclass, iteration, nutrition, get\_size, get\_sex, is\_alive).

Реализованы классы Plankton & Animal, где последний является родительским для Predator & Herbivore. В свою очередь, Predator & Herbivore являются для Shark & Predator\_fish, Whale & Herbivore\_fish соответственно.

В классе Plankton определено поле age.

В классе Animal определены поля age, hungry\_points, sex.

Реализован класс Cell (выполняет функцию клетки), который содержит указатели на объекты классов его обитателей. В этом классе определены и реализованы функции умирания, размножения, питания, поедания.

Реализован класс Field, который позволяет симулировать океан произвольного размера. В этом классе определены и реализованы методы вывода итерации, размножения, передвижения, окончания симуляции и произведение итерации.

Реализован класс ReadFromFile, который читает входные данные через указанный файл.

### **Вывод:**

В ходе работы были теоретически и практически изучена объектно-ориентированная архитектура.