

**Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»**

Кафедра интеллектуальных информационных технологий

**Отчет по лабораторной работе №3
по курсу «Проектирование программ в интеллектуальных системах»**

Выполнил студент группы 921703

Кравцов М.С.

Проверил:

Садовский М.Е.

МИНСК 2020

Тема:

Объектная модель в разработке программной системы. Исключения.

Цель работы:

Получить навыки проведения объектно-ориентированного анализа предметной области.

Задание:

Выбрать вариант задания, провести объектно-ориентированный анализ предметной области, выделить классы и связи между ними. На основании объектной модели реализовать программу на C++ в соответствии с требованиями варианта задания. Общие требования к выполнению лабораторной работы: · использование наследования; · использование полиморфизма; · использование обработки исключений при возникновении ошибочных ситуаций.

Вариант: модель океана

При выполнении этого варианта студент должен будет разработать игровую модель океанического мира. Мир, в котором живут обитатели океана, представляется прямоугольным полем произвольного размера. Прямоугольное поле разбито на равные квадраты (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.

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 обеспечивает разгрузку модели из текстового файла
- Реализованная на основе разработанной модели программа должна:
- получать имя входного файла с описанием мира и описание режима работы (автоматическое моделирование или пошаговое моделирование с ожиданием подтверждения от пользователя) через командный аргумент;
 - выводить в консоль состояние игрового поля на каждом ходу.

Выполнение задания:

В функцию передается 2 параметра – файл, в котором храниться модель океана и кол-во циклов жизни, которое программа должна отработать.

```
int main()
{
    setlocale(LC_ALL, "Russian");
    srand(time(NULL));

    Starter::start("ocean2.txt", 10);

    return 0;
}
```

Далее происходит разгрузка поля из файла и создание поля.

```
void ocean::Starter::start(const string& nameOfFile, const int& cycles)
{
    Field* field = nullptr;

    try
    {
        field = LoadFieldFromFile::loadField(nameOfFile);

        Life::life(cycles, field);

        delete field;
    }
    catch (const runtime_error& ex)
    {
        delete field;
        cerr << endl << ex.what();
    }
}
```

Этапы жизни:

```
Field:
! o< << * ! ! ! * o< !
o< ! * * o< o< * * * *
! o< ! ! << ! o< o< * *
o< * * * o< * ! o< * *
<< ! o< * o< ! * * o< *
o< * * * ! ! * * * *
! o< ! ! << ! o< o< o< *
o< * * * o< * ! o< * *
! o< ! ! * * << ! o< o<
<< * * * * o< * ! o<
Press any key to continue . . .
```

1-ый этап:

```
Move:
! << * o< ! ! ! o< * !
o< ! * * * * o< o< * *
! * ! ! * ! * * o< o<
* o< o< o< * o< ! * * *
* ! o< * << ! o< o< * *
* o< * * ! ! * * * *
! o< ! ! o< ! o< << o< o<
<< o< o< o< * << ! * o< *
! * ! ! o< o< * ! * *
* << * * * * * o< ! *
Press any key to continue . . .
```

2-ой этап:

```
Eat:
! *      * o<  ! !      o< *      * !
<< !      * *      * *      o< o<      * *
o< *      o< o<      * o<      * *      o< o<
* *      * *      * *      ! *      * *
! * o<      o< *      << !      o< o<      * *
* *      * *      ! !      * *      * *
<< *      o< o<      * <<      * *      o< o<
* o<      * *      * *      << *      o< *
! *      ! !      o< o<      * o<      * *
* <<      * *      * *      * *      ! *
Press any key to continue . . .
```

3-ий этап:

```
Reproduction:
! *      * o<  ! !      o< o<      ! !
<< !      * *      ! *      o< o<      * *
o< *      o< o<      * o<      ! *      o< o<
* *      o< *      * *      ! *      o< *
* o<      o< *      << !      o< o<      * *
* *      * *      ! !      * *      * *
<< *      o< o<      * <<      * *      o< o<
* o<      o< *      * *      << *      o< o<
! !      ! !      o< o<      * o<      ! *
* <<      ! *      o< *      * *      ! *
Press any key to continue . . .
```

Так же в ходе каждого этапа организм стареет (уменьшается кол-во жизней организма). Когда кол-во жизней становится ≤ 0 , то организм умирает.

Вывод:

В ходе лабораторной работы мы построили модель океана с применением полиморфизма, наследования, исключений и интерфейса. С помощью полиморфизма наше поле способно вызывать общие для всех обитателей методы, через базовый класс. Это очень удобно, ведь можно создавать классы других обитателей, при этом не переписывая логику самого поля и других классов, которые моделируют этапы жизни.