Практическая работа

По Проектному практикуму

Выполнил студент

Группы К17.1

Шишелов Владимир

Введение

Идиома программирования –  устойчивый способ выражения некоторой составной конструкции в одном или нескольких [языках программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Идиома является шаблоном решения задач, которые часто повторяется в программировании.

Если шаблоны проектирования решают глобальные проблемы во время проектирования продукта, то шаблоны программирования решают проблемы непосредственно при написании кода.

Также как и в шаблонах проектирования в идиомах уже давно есть устоявшиеся примеры решения частых задач.

В данной работе рассмотрим пять наиболее часто используемых идиом и построим к ним диаграммы логики функционирования. А также рассмотрим пример проекта с использованием шаблонов программирования.

Примеры и листинг представлен на языке С#.

Инкремент (Increment) – это операция во многих языках программирования, которая выполняет увеличение переменной. Чаще всего под инкрементом подразумевается увеличение переменной на 1 единицу.

Такой шаблон часто используется в итерационных задачах, а именно в циклах для счета итераций, а также в многих похожих по смыслу задачах.

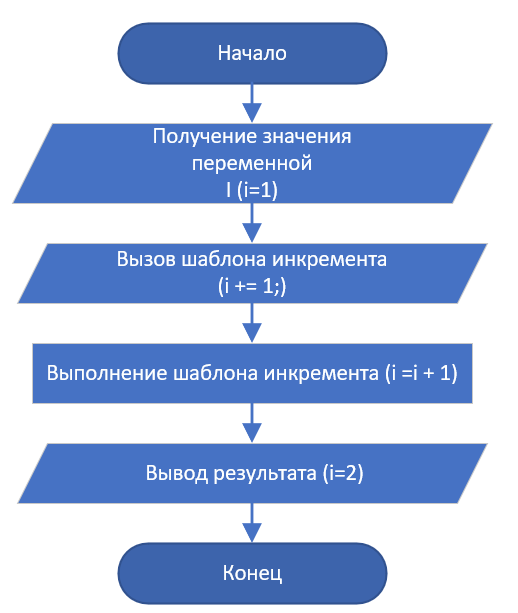


Рисунок 1 – Блок-схема логики Increment.

Пример на языке С#:

i += 1; /\* i = i + 1; \*/

++i; /\* тот же результат \*/

i++; /\* тот же результат \*/

Цикл (Cycle) – определенный алгоритм действий повторяющийся множество раз. Циклы являются управляющими конструкциями, позволяя в зависимости от определенных условий выполнять некоторое действие множество раз

Потребность в этом возникает довольно часто. Например, пользователь последовательно вводит числа, и каждое из них требуется добавлять к общей сумме. Или нужно вывести на экран квадраты ряда натуральных чисел и тому подобные задачи.



Рисунок 2 -Блок-схема логики Cycle.

Пример на языке С#:

for (int i = 0; i < n; i++){

do\_something();

}

Обмен значениями(Swap) – это конструкция применяется при потребности обмена значениями переменных. В большинстве случаев обмен происходит за счет третьей, временной переменной.

Обмен часто применяют при разработке игр, при использовании сложных логических конструкций и работе с большим количеством числовых значений.

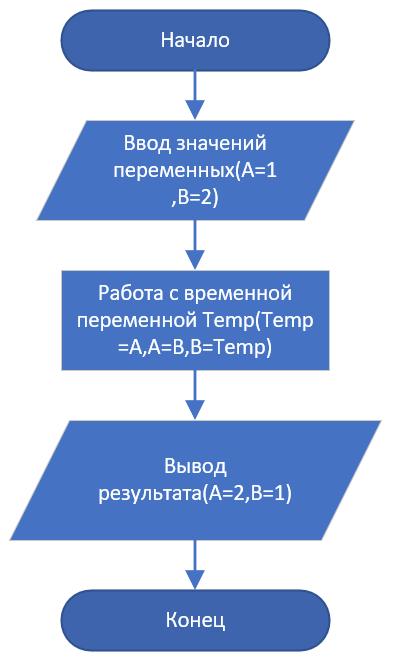


Рисунок 3 – Блок-Схема логики Swap.

Реализация Swap в C#:

public void Swap(a,b){

temp = a;

a = b;

b = temp;

}

Нулевой указатель(NULL) - Одно из отличий ссылочных типов от типов значений состоит в том, что переменные ссылочных типов могут принимать значение NULL. Необходимо отличать 0 от нулевого указателя, так как 0 может восприниматься как значение, NULL указывает на его отсутствие.

Реализация NULL в C#:

Object a = null;

char \* str = NULL;

Рекурсия (Recursion)- представляет такую конструкцию, при которой функция вызывает саму себя. С помощью нее можно реализовать рекурсивные вычисления для увеличения эффективности алгоритма.

Реализация Recursion в C#:

public void function(…){

function(…);

}

Вывод к проделанной работе.

Идиомы программирования созданы для упрощения и ускорения работы разработчика. Идиомы предназначены для выполнения типичных и базовых задач и большая часть проектов не могут обойтись без идиом, а точнее это не целесообразно.