Сложность и воображаемые действия

№ урока:	24	**Курс:**	Процедурное программирование на языке C#

Средства обучения: Visual Studio 2019 Community Edition

Обзор, цель и назначение урока

Задача данного урока – рассмотреть в теории и на практических примерах такие важные понятия как точка принятия решений и сложность программного обеспечения.

Изучив материал данного занятия, учащийся сможет:

- Понимать смысл понятия сложность, применительно к коду готовой программы.
- Понимать смысл понятия точка принятия решения.
- Понимать и уметь применять на практике способы уменьшения сложности программного кода.

Содержание урока

1. Рассмотрение практической задачи
2. Сложность и воображаемые действия
3. Вложенные условные конструкции
4. Решение практической задачи

Резюме

- **Гибкость** программы или гибкость отдельных программных модулей – это способность быстро и легко вносить изменения и дополнения в существующий код.

- С точки зрения рефакторинга, дублирующийся код – это плохо и неправильно. Дублирующегося кода, нужно избегать.

- **Сложность кода** измеряется количеством «**воображаемых действий**», которые нам приходится одновременно держать в голове, чтобы написать тот или иной фрагмент программы. Сравнение, сложение, использование условных конструкций и даже присвоение значения переменной — это всё воображаемые действия.

- Чем больше воображаемых действий у нас в программе, тем больше умственных усилий нужно потратить на понимание и написание программы.

- Самым сложным воображаемым действием в мире программирования считается точка принятия решения.

- **Точка принятия решения** – это воображаемое действие, которое представлено условной конструкцией **if-else.**

CyberBionic systematics

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Точка принятия решения считается сложным воображаемым действием, потому что условная конструкция сама по себе является «составным воображаемым действием», и, в своем условии, конструкция if, может содержать выражение из множества воображаемых действий, и в телах своих блоков, может содержать огромное количество воображаемых действий, в том числе вложенных условных конструкций (то есть, точек принятия решений). Нужно стараться сделать так, чтобы точка принятия решений содержала в себе минимальное количество воображаемых действий.

- Воображаемые действия в коде зависят одно от другого. Именно эти зависимости определяют и образуют программную логику. Говоря точнее - **программной логикой**, можно назвать множество переменных и значений, плюс множество воображаемых действий, плюс множество отношений и зависимостей между воображаемыми действиями.

- Самое сложное в построении логики программы — это попытка разобраться в отношениях между воображаемыми действиями.

- Чтобы минимизировать концентрацию воображаемых действий в блоке, который занимается расчётом, можно частично вынести определенную функциональность за пределы блока, и тем самым «разгрузить» блок.

- Правильное распределение воображаемых действий в программе уменьшает сложность, с которой нам приходится иметь дело в каждый отдельный момент времени.

- Одно из часто используемых понятий для определения сложности кода программы - цикломатическая сложность.

- **Цикломатическая сложность** части программного кода — мера сложности кода, определяемая количеством отдельных маршрутов последовательностей действий, проходящих через программный код. Например, если исходный код не содержит никаких точек ветвления или циклов, то сложность равна единице, поскольку есть только единственный маршрут через код - шаг за шагом. Если код имеет один оператор **if**, содержащий простое условие, то существует два пути через код: один если условие оператора **if** имеет значение **true** и один — если **false**.

Закрепление материала

- Что такое сложность кода?
- Чем определяется сложность кода?
- Как можно уменьшить сложность блока кода?
- В чем состоит сложность точки принятия решений?

Самостоятельная деятельность учащегося

- Задание 1

Ознакомьтесь с дополнительными материалами к уроку.

- Задание 2

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Перепишите код, указанный ниже, в свой проект в Visual Studio, и выполните для него уменьшение сложности, согласно всем правилам, изученным ранее.

```
static void Main(string[] args)
{
    double value;
    int g = 0;
    int x = 1;
    int y = 2;
    int z = 3;

    if (x > y && Convert.ToInt32(Console.ReadLine()) < 15)
    {
        g = y;
        value = g + 3.1415926535897931;
        Console.WriteLine("число g = {0}, значение value = {1}", g, value);
    }
    else
    {
        if (z == y && z < 4)
        g = z;
        value = g + 3.1415926535897931 * 2;
        Console.WriteLine($"число g = {g}, значение value = {value}"); ;
    }

    Console.ReadKey();
}
```

Рекомендуемые ресурсы

https://ru.wikipedia.org/wiki/Цикломатическая_сложность

CyberBionic systematics

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com