

Лекция №3

Программа. Жизненный цикл программы

C#

Программа

Программа – это последовательность инструкций, предназначенная для исполнения вычислительной машиной.

Все программы по характеру использования и категориям пользователей можно разделить на 2 класса:

- утилитарные программы;
- программные продукты (изделия).

Программа

Утилитарные программы – это программы, предназначенные для удовлетворения нужд их разработчиков. Обычно такие программы решают задачи, не предназначенные для широкого распространения.

Программный продукт– это комплекс взаимосвязанных программ, предназначенный для поставки, передачи, продажи пользователю.

Характеристики

программного продукта

- **эффективность** - соответствие требованиям пользователя при минимальных требованиях к ресурсам;
- **надежность** - отсутствие отказов, обеспечение работоспособности продукта;
- **удобство использования** - наличие дружественного интерфейса, системы помощи, обучающей системы, хорошей документации;
- **мобильность** - независимость от ОС и аппаратный средств;
- **коммуникативность** - возможность интеграции с другими программными продуктами;
- **стоимость.**



Определение

Жизненный цикл программы — это период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации.

Этапы жизненного цикла

Жизненный цикл традиционно разделяют на следующие основные *этапы*:

- анализ требований;
- определение спецификаций;
- проектирование;
- кодирование (программирование);
- тестирование и отладка;
- эксплуатация и сопровождение.

Анализ

Для создания конкурентоспособных продуктов в ходе выполнения этого этапа должны быть получены четкие ответы на следующие вопросы:

- Что должна делать программа?
- Какие проблемы она должна решать?
- Что представляют собой входные данные?
- Какими должны быть выходные данные?
- Какими ресурсами располагает проектировщик?



Определение спецификаций

Требования к программе должны быть представлены в виде ряда спецификаций, явно определяющих рабочие характеристики будущей программы. В число таких характеристик могут входить скорость выполнения, объем потребляемой памяти, гибкость применения и др.

Проектирование

На этом этапе создается общая структура программы, которая должна удовлетворять спецификациям; определяются общие принципы управления и взаимодействия между различными компонентами программы. Сюда же можно отнести графическое представление алгоритма работы программы.





Кодирование (программирование)

Закljučается в переводе на язык программирования конструкций, записанных на языке проектирования.



Тестирование и отладка

На этом этапе производится всесторонняя проверка программ.

Тестирование программ - этап разработки компьютерной программы, в процессе которого проверяется работоспособность программы, не содержащей явных ошибок.

Отладка программы - этап разработки компьютерной программы, в процессе которого происходят обнаружение, локализация и устранение явных ошибок в программе.



Тестирование программы

- ПО обычно проверяется по 3 показателям:
- **правильность** - проверка соответствия функциональности ПО решаемым задачам;
 - **эффективность реализации** – это поиск способов повышения эффективности работы программы (быстрая работа, уменьшение затрат ресурсов ПК);
 - **вычислительную сложность** – это экспериментальный анализ сложности алгоритма или экспериментальное сравнение нескольких алгоритмов, решающих одну и ту же задачу.





Сопровождение

Это этап эксплуатации программы.

В процессе эксплуатации устраняются обнаруженные ошибки и обнаруживаются неудачные проектные решения.

Кроме того сопровождение может включать в себя проведение консультаций, обучение пользователей программы, информирование пользователей о новых версиях системы и т.п. Этапа сопровождения в большой степени определяет коммерческий успех программного продукта.



Модели жизненного цикла

Каскадная модель – это модель основной характеристикой которой является возможность перехода с одного этапа на следующий только после полного завершения работы на текущем этапе.



Модели жизненного цикла

Инкрементная модель – это поэтапная модель с промежуточным контролем.



Модели жизненного цикла

Спиральная модель – это модель в которой на каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка.





Основные этапы решения задач на компьютере

1. Постановка задачи;
2. Анализ и исследование задачи, модели;
3. Разработка алгоритма;
4. Программирование;
5. Тестирование и отладка.

Постановка задачи

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т. п.).



Анализ и исследование задачи, модели

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

Разработка алгоритма

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

Программирование

- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке Программирования.



Тестирование

- синтаксическая отладка;
- отладка семантики и логической структуры;
- тестовые расчеты и анализ результатов тестирования;
- совершенствование программы.

Домашнее задание

Проведите сравнительный анализ моделей жизненного цикла ПО. Результат оформить в виде таблицы:

Название модели	Достоинства	Недостатки

Для заполнения таблицы использовать информацию ресурсов сети Интернет.

Домашнее задание выполнить до 26.09
и отправить на почту starlight87@yandex.ru
Указать ФИО, группу

