

**УВАЖАЕМЫЕ СТУДЕНТЫ!** Изучите приведенную лекцию, законспектируйте основные сведения программных средствах, используемых на разных этапах разработки программ.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: [r.bigangel@gmail.com](mailto:r.bigangel@gmail.com) **до 20.02.2023.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

**ВНИМАНИЕ!!!** При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

### Лекция 7 (продолжение)

**Тема: «Основные средства, используемые на разных этапах разработки программ»**

**Цель: Изучить основные средства, используемые на разных этапах разработки программ**

### СРЕДСТВА РАЗРАБОТКИ ПРОГРАММ

*Средства разработки программного обеспечения – совокупность приемов, методов, методик, а также набор инструментальных программ (компиляторы, прикладные/системные библиотеки и т.д.), используемых разработчиком для создания программного кода Программы, отвечающего заданным требованиям.*

Разработка программ – сложный процесс, основной целью которого является создание, сопровождение программного кода, обеспечивающего необходимый уровень надежности и качества. Для достижения основной цели разработки программ используются средства разработки программного обеспечения.

### ОСНОВНЫЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ НА РАЗНЫХ ЭТАПАХ РАЗРАБОТКИ ПРОГРАММ

В зависимости от предметной области и задач, поставленных перед разработчиками, разработка программ может представлять собой достаточно сложный, поэтапный процесс, в котором задействовано большое количество участников и разнообразных средств. Для того, чтобы определить, когда и в каких случаях какие средства применяются, выделяют следующие основные этапы разработки программного обеспечения:

1. Проектирование приложения.

2. Реализация программного кода приложения.
3. Тестирование приложения.

### **Средства проектирования приложений**

На этапе проектирования приложения в зависимости от сложности разрабатываемого программного продукта, напрямую зависящего от предъявляемых требований, выполняются следующие задачи проектирования:

1. Анализ требований.
2. Разработка архитектуры будущего программного обеспечения.
3. Разработка устройств основных компонент программного обеспечения.
4. Разработка макетов Пользовательских интерфейсов.

Результатом проектирования обычно является «Эскизный проект» (Software Design Document) или «Технический проект» (Software Architecture Document).

Задача «Анализ требований» обычно выполняется с использованием методов системологии (анализа и синтеза) с учетом экспертного опыта проектировщика. Результатом анализа обычно является содержательная или формализованная модель процесса функционирования программы. В зависимости от сложности процесса для построения данных моделей могут быть применены различные методы и вспомогательные средства. В общем случае для описания моделей обычно применяются следующие нотации (в скобках приведены программные средства, которые могут быть использованы для получения моделей):

- BPMN (Vision 2003 + BPMN, AcuaLogic BPMN, Eclipse, Sybase Power Designer).
- Блок-схемы (Vision 2003 и многие другие).
- ER-диаграммы (Visio 2003, ERWin, Sybase Power Designer и многие другие).
- UML-диаграммы (Sybase Power Designer, Rational Rose и многие другие).
- макеты, мат-модели и т.д.

Результаты анализа позволяют сформировать обоснованные требования к той или иной функциональности разрабатываемой программы и просчитать реальную выгоду от внедрения разрабатываемого продукта. Более того, иногда получается так, что по результатам анализа первоначальные цели и задачи автоматизации кардинально меняются или по результатам оценки эффективности разработки и внедрения принимается решение продукт не разрабатывать.

Целью второй и третьей задачи из приведенного списка задач является разработка модели (описания) будущей системы, понятной для кодировщика – человека, который пишет код программы. Здесь огромное значение имеет то, какую парадигму программирования необходимо использовать при написании программы. В качестве примера основных парадигм необходимо привести следующее:

- Функциональное программирование;
- Структурное программирование;

- Императивное программирование;
- Логическое программирование;
- Объектно-ориентированное программирование (прототипирование; использование классов; субъективно-ориентированное программирование).

Выбор её во многом зависит от сложившихся привычек, опыта, традиций, инструментальных средств, которыми располагает коллектив разработчиков. Иногда разрабатываемый программный продукт настолько сложен, что для решения ряда задач в разных компонентах системы используются разные парадигмы. Выбор того или иного подхода накладывает ограничения на средства, которые будут применены на этапе реализации программного кода. Результатом решения данной задачи в зависимости от подхода могут быть (в скобках приведены программные средства, которые могут быть использованы для их получения):

- диаграмма классов и т.д (Ration Rose, Sybase PowerDisigner и многие другие).
- описание модулей структур и их программного интерфейса (например, Sybase PowerDisigner и многие другие).

Разработка макетов пользовательских интерфейсов подразумевает создание наглядного представления того, как будут выглядеть те или иные видеоформы, окна в разрабатываемом приложении. Решение данной задачи основывается на применении средств дизайнера.

### **Средства реализации программного кода**

На этапе реализации программного кода выполняется кодирование отдельных компонент программы в соответствии с разработанным техническим проектом. Средства, которые могут быть применены, в значительной степени зависят от того, какие подходы были использованы во время проектирования и, кроме этого, от степени проработанности технического проекта. Тем не менее, среди средств разработки программного кода необходимо выделить следующие основные виды средств:

- методы и методики алгоритмирования.
- языки программирования (C++, Си, Java, C#, php и многие другие);
- средства создания пользовательского интерфейса (MFC, WPF, QT, GTK+ и т.д.)
- средства управления версиями программного кода (cvs, svn, VSS).
- средства получения исполняемого кода (MS Visual Studio, gcc и многие другие).
- средства управления базами данных (Oracle, MS SQL, FireBird, MySQL и многие другие).
- отладчики (MS Visual Studio, gdb и т.д.).

### **Средства тестирования программ**

Основными задачами тестирования является проверка соответствия функциональности разработанной программы первоначальным требованиям, а также выявление ошибок, которые в явном или неявном виде проявляются во время работы программы. Среди основных работ по тестированию можно выделить следующее:

- Тестирование на отказ и восстановление.
- Функциональное тестирование.
- Тестирование безопасности.
- Тестирование взаимодействия.
- Тестирование процесса установки.
- Тестирование удобства пользования.
- Конфигурационное тестирование.
- Нагрузочное тестирование.

Среди основных видов средств, которые могут быть применены для выполнения поставленных работ можно привести следующие:

- средства анализа кода, профилирования (Code Wizard – ParaSoft, Purify – Rational Software, Test Coverage – Semantic и т.д.);
- средства для тестирования функциональности (TEST – Parasoft, QACenter – Compuware, Borland SilkTest и т.д.);
- средства для тестирования производительности (QACenter Performance – Compuware и т.д).