

# НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Задание №1

по дисциплине «Технологии программирования»

Студенты:

Кац М.

Шляхтин Д.

Преподаватель:

Орлов И.С.

Группа: АО-51

Факультет: АВТ

Новосибирск 2017

# UML

- **UML** – это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования ОО систем. **UML** призван поддерживать процесс моделирования ПС на основе ОО подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем.

# Применение

- Определение, визуализация, проектирование и документирование, в основном, программных систем. Также позволяет разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение, агрегация и поведение) и больше сконцентрироваться на проектировании и архитектуре.

# Преимущества

- UML объектно-ориентирован, в результате чего методы описания результатов анализа и проектирования семантически близки к методам программирования на современных объектно-ориентированных языках;
- UML позволяет описать систему практически со всех возможных точек зрения и разные аспекты поведения системы;
- Диаграммы UML сравнительно просты для чтения после достаточно быстрого ознакомления с его синтаксисом;
- UML расширяет и позволяет вводить собственные текстовые и графические стереотипы, что способствует его применению не только в сфере программной инженерии;
- UML получил широкое распространение и динамично развивается.

# Недостатки

- Избыточность языка. UML часто критикуется как неоправданно большой и сложный.
- Неточная семантика. Так как UML определён комбинацией себя (абстрактный синтаксис), то он лишен скованности, присущей языкам, точно определённым техниками формального описания.
- Проблемы при изучении и внедрении. Вышеописанные проблемы делают проблематичным изучение и внедрение UML.
- Только код отражает код. Ещё одно мнение — что важны рабочие системы, а не красивые модели
- Кумулятивная нагрузка или же рассогласование нагрузки. Т.е. неспособность входа одной системы воспринять выход другой.
- Пытается быть всем для всех. UML — это язык моделирования общего назначения, который пытается достигнуть совместимости со всеми возможными языками разработки.

# Применение

- UML - идеальный инструмент для индустриального программирования. В экстремальном же он может играть лишь вспомогательную роль на второстепенных этапах и как поддержка (причем отнюдь не всегда адекватная) некоторых аспектов документации.
- Довольно хорошо справляется с программированием автоматов.

# .NET

**Хотя, однозначного ответа на этот вопрос не существует, можно сказать, что «.NET» это:**

- Платформа, на которой выполняется программный код;
- Библиотека программного кода, который можно использовать с помощью средств любого языка программирования; „
- Новые версии инструментов разработки приложений, например, Visual Studio; „
- Набор серверных продуктов, способствующих продвижению платформы .NET; „
- Новый способ проектирования и создания приложений, разделяющих выполнение задачи между различными компонентами, которые могут быть расположены даже в Internet.

# Преимущества

- Многоязыковая среда разработки. Разработка программного обеспечения на наиболее подходящем языке программирования;
- Строгая иерархичность организации пространств для типов, классов и имен сущностей;
- Компонентно-ориентированный подход к проектированию и реализации программного обеспечения.



# Недостатки

- Сложности работы с некоммерческими релизами программного обеспечения (некоторая неустойчивость, отсутствие полномасштабной документации);
- Поддержка ряда теоретически интересных и практически полезных языков программирования не в полном объеме (SML для Visual Studio .NET – в процессе реализации);
- Инструментарий .NET (и компиляторы для языков программирования) не ратифицированы по международным стандартам.

# CLR

- CLR - исполняющая среда для байт-кода CIL (MSIL), в который компилируются программы, написанные на .NET-совместимых языках программирования (C#, Managed C++, Visual Basic .NET, F# и прочие).
- В отличие от переносимых виртуальных машин Java, абстрагирующихся от нижележащих операционных систем, CLR позиционируется как не «виртуализированная» платформа, тесно связанная с операционной системой Microsoft Windows.

# MFC

- Это библиотека, которая используется в для программирования под Windows на Visual C++.
- MFC инкапсулирует в себя многие рутинные операции, с которыми сталкивается программист при создании программ по Windows.

# Принцип действия

- Библиотека MFC облегчает работу с GUI путём создания каркаса приложения — «скелетной» программы, автоматически создаваемой по заданному макету интерфейса и полностью берущей на себя рутинные действия по его обслуживанию.
- Так же MFC предоставляет объектно-ориентированный слой обёрток над множеством функций Windows API, делающий несколько более удобной работу с ними. Этот слой представляет множество встроенных в систему объектов в виде классов и опять же берёт на себя рутинные действия вроде закрытия дескрипторов и выделения/освобождения памяти.

# WPF

- Система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая подсистема в составе .NET Framework использующая язык XAML.

# Преимущества

- Привязка данных
- Возможность использования стилей
- Использование шаблонов элементов управления
- Использование шаблонных данных
- Использование ресурсов
- Масштабируемый и гибкий набор графических возможностей
- Двухмерная и трёхмерная графика

# Недостатки

- Идеология WPF в корне отличается от всего, что было раньше. Из-за своей гибкости тяжёл в понимании.
- Чувствительность к драйверам видеокарты.
- Плохая документированность.
- Неудачные названия многих классов. Путаница в названиях.
- Слабая типизированность.

# XAML

- XML – расширяемый язык разметки для приложений-основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft.
- Включает в себя основные четыре категории элементов: панели, элементы управления, элементы, связанные с документом и графические фигуры.



# Применение

- Широко используется в WPF, как язык разметки. Так же используется в WWF и Silverlight для определения последовательности выполняемых действий.

# Преимущества

- Так как XAML основан на XML, существует возможность работать над содержимым без необходимости компиляции.
- Широкий выбор инструментов для создания и редактирования, среди которых инструменты визуального конструирования и стандартные текстовые редакторы.
- Простота в понимании и использовании.
- Поддержка большого количества языков программирования.

# ASP.NET

- ASP.NET - технология создания мощных клиент-серверных интернет приложений.
- Является составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP.

# Преимущества

- Широкий список поддерживаемых языков программирования.
- Достаточно лёгкий переход с Microsoft ASP.
- Компонентно-ориентированная архитектура.
- Использование строго типизируемых компилируемых языков.
- Более высокая скорость работы при больших нагрузках, по сравнению с конкурентами.
- Поддержка всех браузеров.
- Возможность создания динамических страниц.

# Недостатки

- Для сохранения введённых пользователем данных, требуется их сохранение в невидимом компоненте.
- Более низкая скорость работы при низких нагрузках, по сравнению с конкурентами.

# SharePoint

- SharePoint - это коллекция программных продуктов и компонентов, включающая в себя:
- Набор веб-приложений для организации совместной работы;
- Функциональность для создания веб-порталов;
- Модуль поиска информации в документах и информационных системах;
- Модуль создания форм для ввода информации;
- Функциональность для бизнес-анализа.
- Функциональность управления рабочими процессами и систему управления содержимым масштаба предприятия;

# Преимущества

- Возможность использования сайта, созданного на платформе SharePoint, как хранилища данных.
- Возможность использования сайта, созданного на платформе SharePoint, для исполнения облегчающих взаимодействие веб-приложений, таких как вики и блоги.
- Допустимость управления и взаимодействия пользователя с информацией посредством элементов управления (веб-частей).
- Большая библиотека готовых модулей, в том числе от сторонних разработчиков.
- Возможность довольно удобно использовать SharePoint как ECM.
- Широкие возможности настройки кэширования.
- Легко кластеризуется и масштабируется.

# Недостатки

- При хранении информации на сайте, выполненном при помощи SharePoint, затрудняется создания backup.
- Отдельная настройка правил доступа к документам.
- Плохое качество официальной документации.
- Нестабильность при разработке.
- Не очень удачное архитектурное решения, практически исключаящее возможность использования TDD.



# Применение

- Создание публичных сайтов и интранетов.
- Построение систем с открытыми бизнес-процессами.
- Сбор статистики.
- Создание онлайн-чата, в том числе с видео.
- Документооборот.
- Интеграция с Active Directory, Outlook.
- Создание онлайн калькуляторов, посредством возможности использования Excel Services.

# Scrum

- Scrum - методология гибкой разработки ПО. Методология делает акцент на качественном контроле процесса разработки.
- Является каркасом разработки, с использованием которого люди могут решать появляющиеся проблемы, при этом продуктивно и производя продукты высочайшей значимости.

# Преимущества

- Возможность быстрого запуска проекта с наиболее приоритетными функциями и минимально возможным бюджетом;
- Ежедневный контроль над ходом работ, и более гибкий контроль над бюджетом проекта;
- Частые демонстрации проекта. Применение данной методологии предполагает регулярную демонстрацию разработок заказчику, что позволяет в будущем избежать полного провала работы команды и разочарований клиента;
- Возможность вносить коррективы в техническое задание по ходу реализации проекта, что является несомненным преимуществом для заказчика.

# Недостатки

- Сложности при заключении договоров. Scrum в принципе не подразумевает наличие фиксированного бюджета и фиксированного технического задания, что затрудняет юридическое оформление такого рода договоренностей;
- Большое количество исключений. Специалисты в этой области считают данную методологию неприменимой для работы с государственными заказами, а также совершенно нерабочей при низкой квалификации команды, заниженных сроках работ или бюджете, некомпетентном менеджере проекта. В то время как другие методологии позволяют завершить проект при подобных условиях, хотя и на низком уровне;
- Узкая специализация методов. Так, например, если использовать Scrum при разработке сайтов, этапы дизайна и контента уже будут выходить за рамки методологии и требовать совершенно иного подхода.

# TDD

- TDD (разработка через тестирование) –это набор юнит-тестов, которые можно запустить на любом этапе, чтобы убедиться, что программное обеспечение по-прежнему работает.
- Основывается на 3 коротких циклах разработки: создание теста, создание кода, рефакторинг кода.

# Преимущества

- Основным преимуществом является то, что программист, использующий TDD, становится более продуктивным и чувствует себя уверенней.
- Благодаря тестам проще представить, какая функциональность необходим пользователю.
- Чаще всего, несмотря на то, что при тестировании пишется больше кода, общее время на разработку оказывается меньше.
- Возможность рефакторинга кода, без риска его испортить.
- Способствует более модульному, гибкому и расширяемому коду.
- Автоматизированные тесты показывают все пути исполнения.
- Тесты могут использоваться как документация, что способствует пониманию кода, другими разработчиками

# Недостатки

- На данный момент, существуют задачи, которые невозможно решить с помощью использования TDD.
- При необходимости прохождения многофункциональных тестов, довольно трудно применять TDD.
- В некоторых случаях, время на разработку увеличивается.
- Если разработчик неправильно истолковал требования к приложению, и тест, и тестируемый модуль будут содержать ошибку.
- Возможно появление ложного ощущения надёжности.
- Тесты являются источником накладных расходов.
- Уровень покрытия тестами, получаемый в результате разработки через тестирование, не может быть легко получен впоследствии.

# WCF

- WCF – это фреймворк для построения распределенных приложений и межпроцессного взаимодействия, который является логическим развитием предыдущих подобных технологий компании Майкрософт, в частности веб-сервисов, .Net Remoting и DCOM.



# Возможности

- Сервис-ориентированность.
- Совместимость с веб-службой.
- Несколько шаблонов сообщений.
- Поддержка публикации метаданных службы с использованием форматов, указанных в отраслевых стандартах.
- Безопасность, например возможность шифрования сообщений.
- Наличие нескольких транспортов и кодировок.
- Наличие надёжных сообщений и сообщений в очереди.
- Наличие устойчивых сообщений.
- Поддержка транзакций.
- Поддержка таких технологий как AJAX и REST.
- Имеется возможность расширения среды.
- Использование контрактов данных.
- Интеграция с другими технологиями Microsoft.

# Преимущества

- Исключительная гибкость, что позволяет применять WCF в ряде других продуктов Microsoft.
- В сочетании с WF, рабочие процессы упрощают разработку приложений, путём инкапсуляции.
- Возможность абстрагироваться от вида распределённого взаимодействия и легко позволяет изменить своё решение в будущем.
- Надёжный и защищённый обмен данными.
- Поддержка большинства протоколов, таких как HTTP/HTTPS, TCP, MSMQ и т.д.
- Хостинг на IIS и сервисах Windows.
- Высокая производительность.

# Недостатки

- Основным недостатком является, достаточно сложное обучение.
- Для развёртывания приложений WCF требуются дополнительные аппаратные ресурсы.
- API функции контролируются исключительно MS.
- Из-за сложного обучения, возможно неправильное понимание устройства работы данной технологии, что в итоге приведёт к большому количеству ошибок.



# НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Студенты:

Кац М.

Шляхтин Д.

Преподаватель:

Орлов И.С.

Группа: АО-51

Факультет: АВТ

Новосибирск 2017