**День 1.**

**Описание компетенции. Обзор основного программного обеспечения. Изучение предметной области. Проектирование программной системы**

**Основные темы цикла занятий**

[1. Описание компетенции 1](#_Toc92551363)

[2. Знакомство с используемыми в курсе программами 2](#_Toc92551364)

[3. Изучение материалов задания демонстрационного экзамена 3](#_Toc92551365)

[4. Проектирование программной системы 4](#_Toc92551366)

# 1. Описание компетенции

**WorldSkills International (WSI)** — международная некоммерческая ассоциация, целью которой является повышение статуса и стандартов профессиональной подготовки и квалификации по всему миру, популяризация рабочих профессий через проведение международных соревнований по всему миру. Основана в 1953 году. На сегодняшний день в деятельности организации принимают участие 77 стран. Своей миссией WSI называет привлечение внимания к рабочим профессиям и создание условий для развития высоких профессиональных стандартов. Её основная деятельность — организация и проведение профессиональных соревнований различного уровня для молодых людей в возрасте до 22 лет. Раз в два года проходит мировой чемпионат рабочих профессий WorldSkills, который также называют «Олимпиадой для рабочих рук». В настоящее время это крупнейшее соревнование подобного рода.

Участники компетенции «Программные решения для бизнеса» разрабатывают, модифицируют и документируют информационные системы. Эксперты оценивают результаты работы конкурсантов, профессионализм выполнения работы и возможность развития и обслуживания разработанных систем.

Для выполнения разработки информационных систем участникам необходимо знать и уметь применять методики анализа деятельности пользователей, приемы проектирования архитектуры, иметь уверенные навыки программирования на языке высокого уровня, а также владеть подходами к описанию и демонстрации результатов своей работы. Эти ключевые навыки нужно развивать и укреплять участникам.

Профессионалы в области программных решений могут быть приняты на работу в крупные, средние и малые предприятия в качестве разработчиков ПО, в компании, выпускающие ПО, в качестве подрядчиков, в консалтинговые фирмы. Они могут работать в разных ролях, в том числе в роли разработчика, позволяющего адаптировать или настраивать программные решения, в роли службы поддержки при работе с системами, в роли бизнес-аналитика для предоставления решений, упрощающих и автоматизирующих рутинные офисные и бизнес-процессы, а также в роли тренера для обучения пользователей применению прикладных программ.

**Для успешного освоения материала рекомендуем вам изучить следующие понятия:**

**База данных** (БД; англ. *database*) — организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов

**Кросс-платформенность** (англ. *cross-platform*) — способность программного обеспечения работать с двумя и более аппаратными платформами и (или) операционными системами

**Эмулятор** (англ. *emulator*) — аппаратное или программное обеспечение, которое позволяет одной компьютерной системе (называемой хостом) вести себя как другая компьютерная система (называемая гостем)

**Integrated Development Environment** (IDE; рус. интегрированная среда разработки) — система программных средств, используемая программистами для разработки программного обеспечения

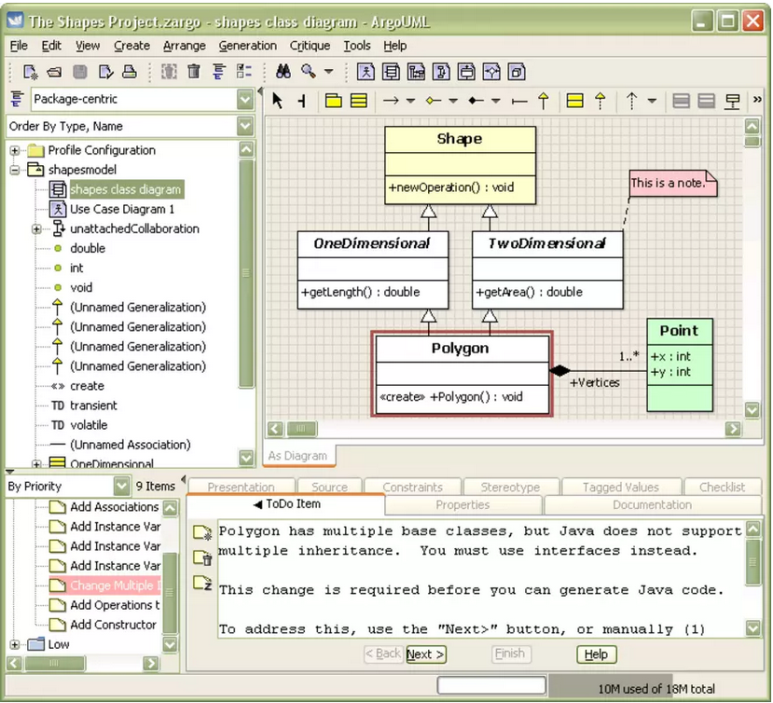
**WorldSkills Standard Specification** (WSSS; рус. спецификация стандартов WorldSkills) — спецификация стандартов, которая определяет знание, понимание и конкретные компетенции, лежащие в основе лучших международных практик технического и профессионального уровня выполнения работы

**Основные разделы компетенции по WSSS**

* Организация и управление работой
* Компетенции общения и межличностных отношений
* Решение проблем инноваций и креативность
* Анализ и проектирование программных решений
* Разработка программных решений для различных платформ
* Тестирование программных решений

# 2. Знакомство с используемыми в курсе программами

**ArgoUML -** инструмент, разработанный на Java, с помощью которого можно создавать UML-модели, совместимые со стандартами, установленными версией 1.4 данного языка программирования.



С помощью ArgoUML можно создать 9 типов диаграмм: диаграммы классов, состояний, деятельности, прецедентов, взаимодействия, развертывания и последовательности.

ArgoUML использует следующие форматы для сохранения диаграмм: GIF, PNG, PostScript, encapsulated PS, PGML и SVG.

Функционал AgroUML также включает:

- Совместимость со стандартом XMI: позволяет использовать модели, созданные с помощью UML в других программах.

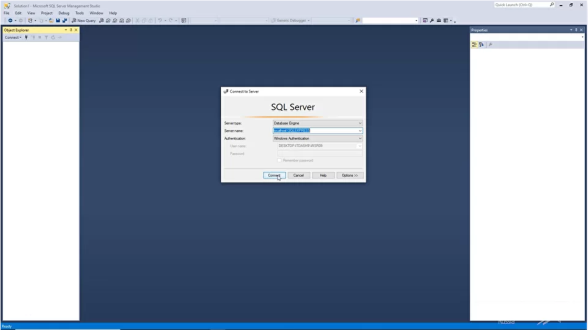
- Поддержка OCL для классов.

- Генерация исходного кода для Java, C , C, PHP4 и PHP5.

- Обратный инжиниринг из исходного кода.

- Автоматическая верификация модели UML (design critics), созданная ArgoUML для ее улучшения.

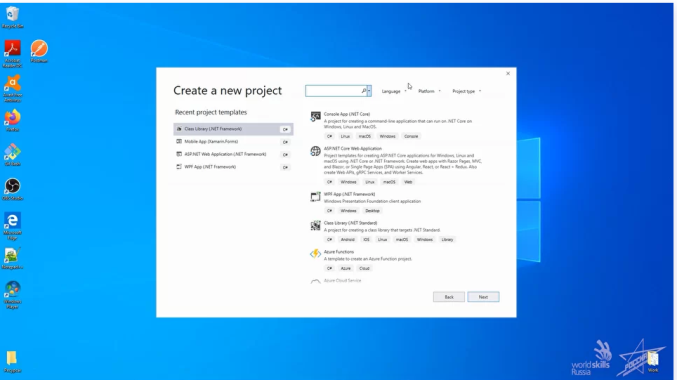
**SQL Server Management Studio (SSMS)** – Интегрированная среда, предоставляющая средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных. В курсе SSMS будет использоваться для работы с базами данных



Элементы рабочего окна:

* Рабочая область
* Обозреватель объектов
* Настройка компонентов базы данных

**Microsoft Visual Studio 2019** – IDE для создания, редактирования и отладки кода, а также публикации приложений. В курсе Visual Studio будет использоваться для разработки настольных программных решений



Элементы рабочего окна:

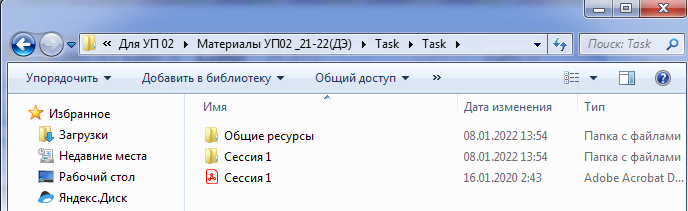
* Окно редактора
* Панель инструментов
* Обозреватель решений
* Панель свойств

*Во время проведения демоэкзамена выделяется время на изучения необходимого ПО.*

Вы узнали о  компетенции «Программные решения для бизнеса» в WSSS. Вы также познакомились с программными продуктами, которые будут использоваться для разработки решений в этом цикле занятий.

# 3. Изучение материалов задания демонстрационного экзамена

Скачайте в свою директорию папку Task (ссылка Задание). Изучите содержимое папки.



В папке **Общие ресурсы** необходимо внимательно прочесть Требования и рекомендации, Руководство по стилю и Введение ПО\_Услуги школы. Эти документы определяют основные направления работы на демоэкзамене. Остальные файлы вам пригодятся во время выполнения задания.

Папка Общие ресурсы становится доступна студенту до выполнения основной части заданий. *На ее изучение студенту выделяется отдельное время*.

Файл **Сессия 1.pdf** содержит непосредственно задание самого экзамена. В зависимости от вариантов, экзамен может содержать более одной сессии. *Содержимое этого файла доступно студенту с момента начала отсчета времени сессии.*

Папка **Сессия 1** содержит материалы, необходимые для выполнения заданий сессии. В нем могут содержаться скрипты базы данных, файлы данных и др.

# 4. Проектирование программной системы

**4.1 Глоссарий**

Для успешного освоения материала рекомендуем вам изучить следующие понятия:

**Use Case Diagram.** Диаграмма вариантов использования. Диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне

**Предметная область**. Часть реального мира, рассматриваемая в пределах данного контекста

**UML**. *Unified Modeling Language* (унифицированный язык моделирования). Язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

**ТЗ.** Техническое задание. Документ, содержащий требования заказчика к объекту закупки, определяющие условия и порядок ее проведения для обеспечения государственных или муниципальных нужд, в соответствии с которым осуществляются поставка товара, выполнение работ, оказание услуг и их приемка

**Actor.** Актер (*Use Case*). Роль объекта вне системы, который прямо взаимодействует с ее частью — конкретным элементом

**Use Case**. Вариант использования (прецедент). Описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды

**Диаграмма классов.** *Class Diagram*. Структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними

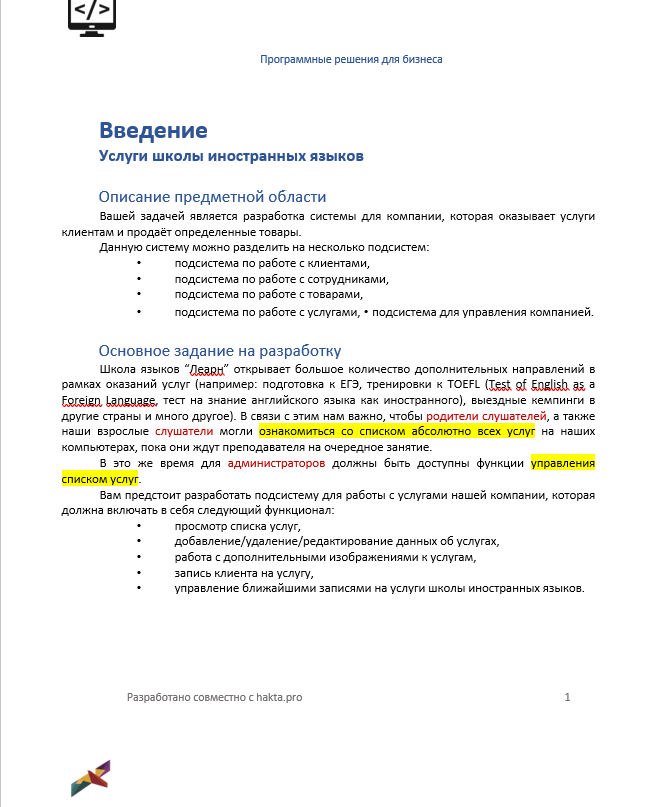
**4.2 Анализ предметной области и проектирование**

Анализ предметной области и проектирование являются первыми этапами в жизненном цикле создания программного решения. Одним из результатов этого этапа является диаграмма вариантов использования (*Use Case)*, описывающая основные группы пользователей системы и варианты ее использования.

Предметная область — это часть реального мира, данные и особенности которой будут отражены в разрабатываемом программном решении. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т. д. Предметная область бесконечна и содержит как важные понятия и данные, так и малозначащие или вообще ничего не значащие данные. Так, если в качестве предметной области выбрать учет товаров на складе, то понятия «накладная» и «счет-фактура» являются важными, а то, что сотрудница, принимающая накладные, имеет двоих детей — это для учета товаров неважно. Однако с точки зрения отдела кадров данные о наличии детей являются важными. Таким образом, значимость данных зависит от выбора предметной области.

**4.3 Изучение предметной области**

В рамках курса для демонстрации основных модулей была выбрана деятельность школы иностранных языков. Давайте проанализируем вводное описание и определим данные, которые действительно необходимы для нашей системы. Перед вами описание предметной области (Введение) (важные данные мы будем отмечать маркерами: красным  — роль пользователя, желтым  — важные действия, которые могут совершать пользователи).



Итак, мы выделили:

**Родители слушателей, слушатели** — просмотр списка услуг

**Администратор** - добавление/удаление/редактирование данных об услугах

**Администратор** - работа с дополнительными изображениями к услугам,

**Администратор** - запись клиента на услугу,

**Администратор** - управление ближайшими записями на услуги школы иностранных языков

**4.4 Использование ArgoUML для создания диаграмм**

После определения требований переходим к этапу проектирования. В ходе проектирования архитектором создается проектная документация, включающая:

* текстовые описания
* диаграммы
* модели будущей программы

Для этого используется графический язык для визуализации, описания параметров, конструирования и документирования различных систем UML. Для визуализации модели существуют различные типы диаграмм. Мы используем некоторые из них.

* Диаграмма вариантов использования (*use case diagram*)
* Диаграмма классов (*class diagram*) и др.

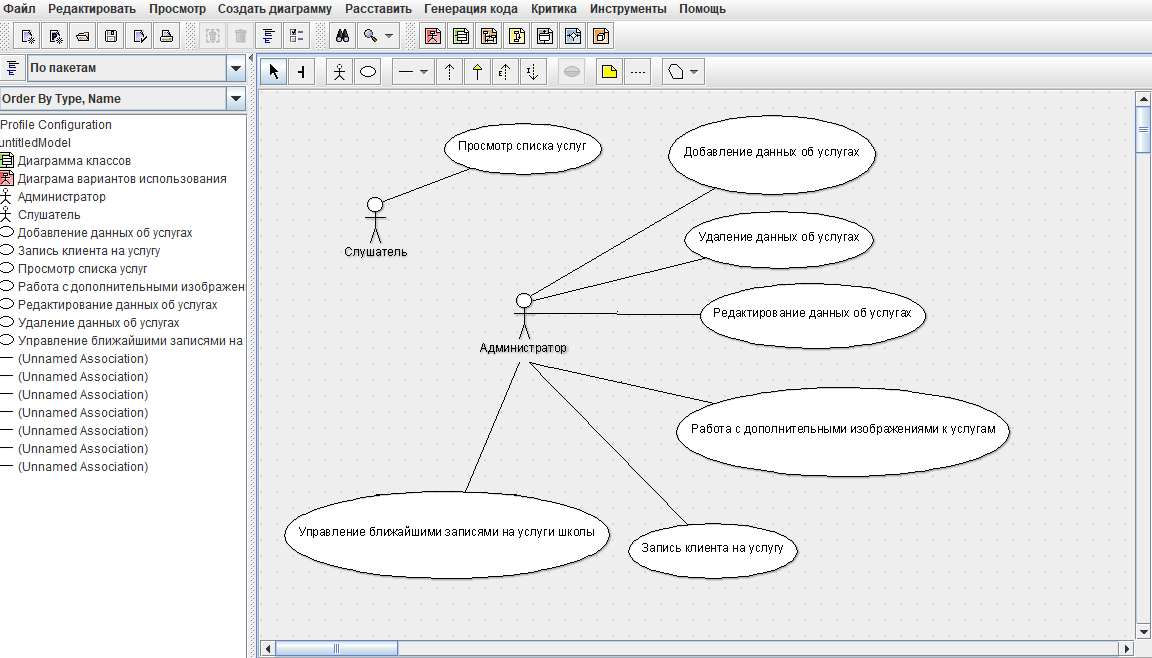
**4.5 Создание диаграммы Use -case для школы**

Определение рамок системы согласно заданию. Для этого используем Диаграмму Use-case, там будут располагаться актеры и прецеденты (функционал, реализуемый системой).

Так как Родители слушателей и Слушатели - это роли с одинаковыми возможностями, объединим их в одну роль Слушатели и зададим актера Слушатель с прецедентом «Просмотр списка услуг».

Актер Администратор имеет возможность выполнять в системе сценарии, описанные в предметной области: «Добавление/удаление/редактирование данных об услугах», «Работа с дополнительными изображениями к услугам», «Запись клиента на услугу», «Управление ближайшими записями на услуги школы иностранных языков».

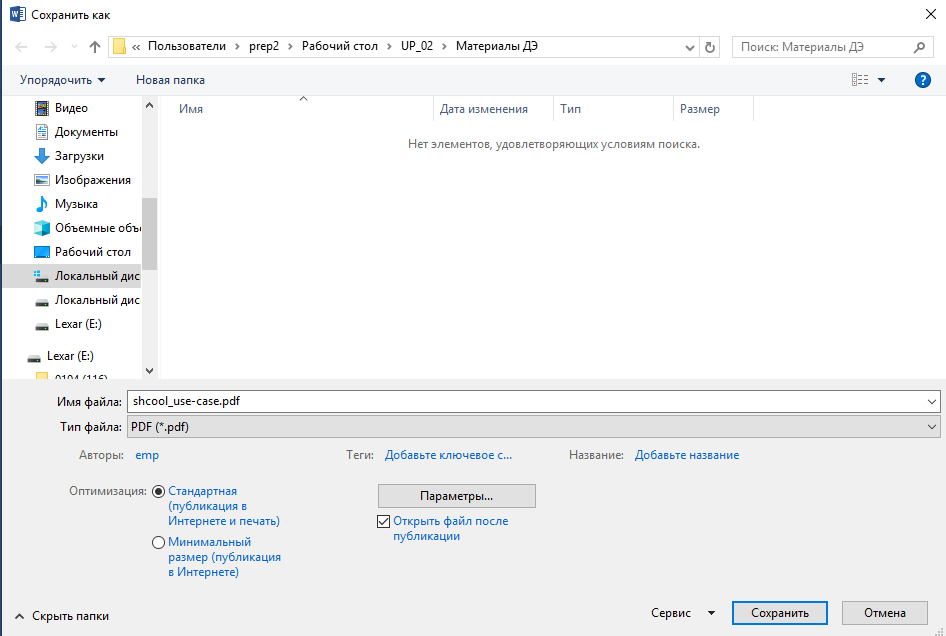
Откроем проектировщик ArgoUML. Сразу сохраним его в своей папке как ***school.zargo.*** Построим диаграмму вариантов использования для заданной предметной области.



**4.6. Формат сохранения диаграммы**

На этом проектирование диаграммы завершено, и мы можем перейти к ее сохранению. Созданная диаграмма по умолчанию хранится в формате .zargo, но для гарантированного запуска файла на других устройствах рекомендуется сохранять диаграмму еще и в .pdf формате.

Для этого сделайте скриншот диаграммы, вставьте в новый Документ Micrisoft Word. Сохраните его через «Сохранить как…» в формате .pdf с именем ***school\_use-case***.



**4.7 Диаграмма классов**

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются также свойства классов, операции классов и ограничения, которые накладываются на связи между объектами

Построение диаграммы классов для программной системы содержит основные шаги построения диаграммы классов:

* Добавление классов
* Добавление связей и их настройка
* Добавление атрибутов и операций

На диаграмме класс изображается в виде прямоугольника, разделенного на три части: имя класса (на английском языке), его атрибуты и его операции.

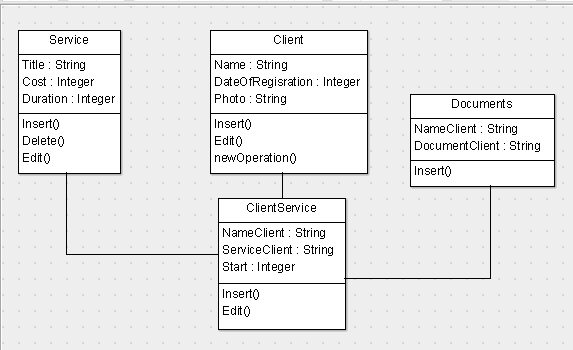
Откроем файл ***school.zargo***. Построим диаграмму классов для заданной предметной области. Для этого выделим те классы, которые содержат основные данные, необходимые для работы системы:

- Service (содержит данные об услугах школы);

- Client (содержит данные о клиентах школы);

- ClientService (содержит данные о клиентах, записанных на услуги);

- Documents (содержит данные о документах клиентов).



Сохраним диаграмму в формате **.pdf** в своей папке.