МИНИСТЕРСТВО ОБРАЗОВАНИЕ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ПО ОБРАЗОВАНИЮ

МОГИЛЁВСКОГО ОБЛАСТНОГО ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«МОГИЛЕВСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

Специальность 2-40 01 01

Учебная группа ПО-455

Учебная дисциплина Технология разработки

программного обеспечения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №12**

**«Построение диаграмм классов»**

Выполнил Савич А.О

Проверил Базанова Л.Д

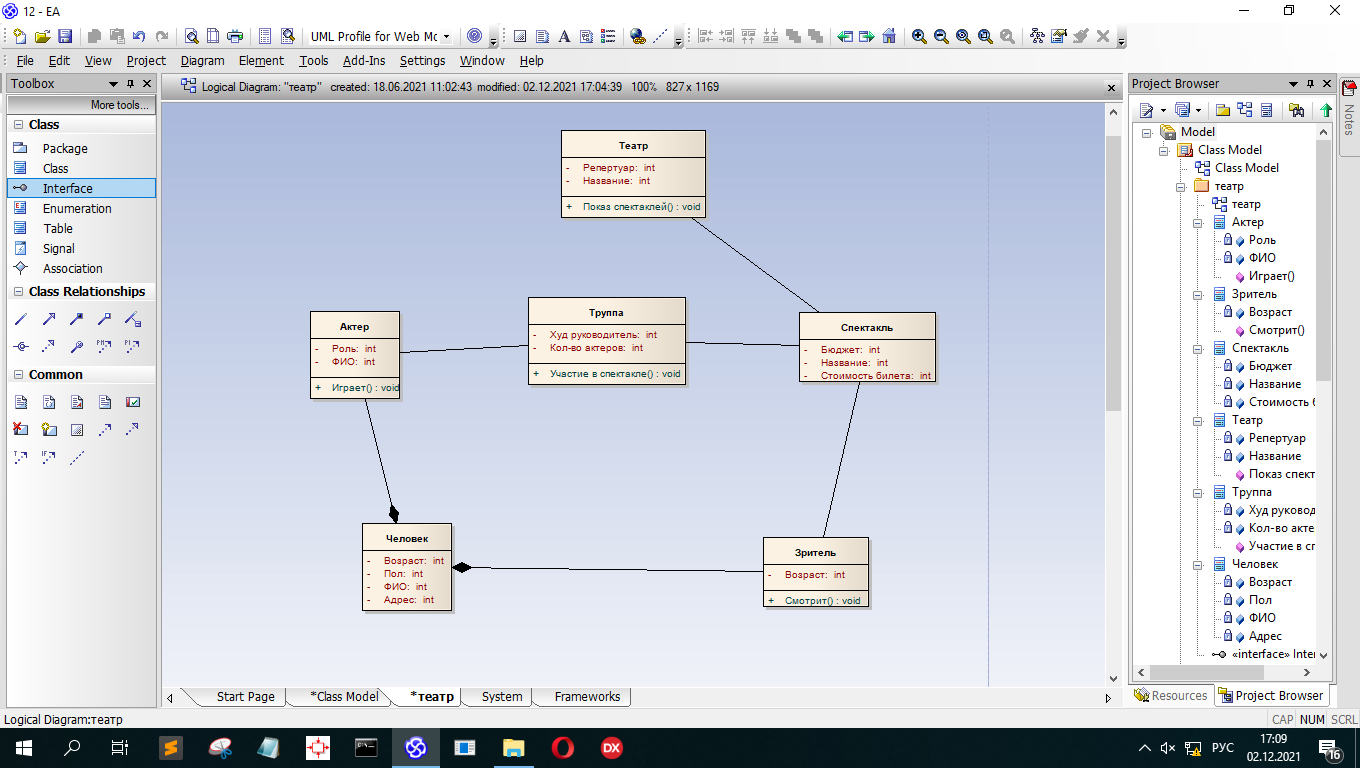
2021

1.Цель работы

Формирование умения проектировать и строить диаграмму классов по заданной предметной области

2. Индивидуальное задание

2.1 По описанию, представленному ниже, постройте диаграмму классов для предметной области ТЕАТР.



2.2Постройте диаграмму классов для выбранной предметной области

Вариант №8 Интернет-магазин

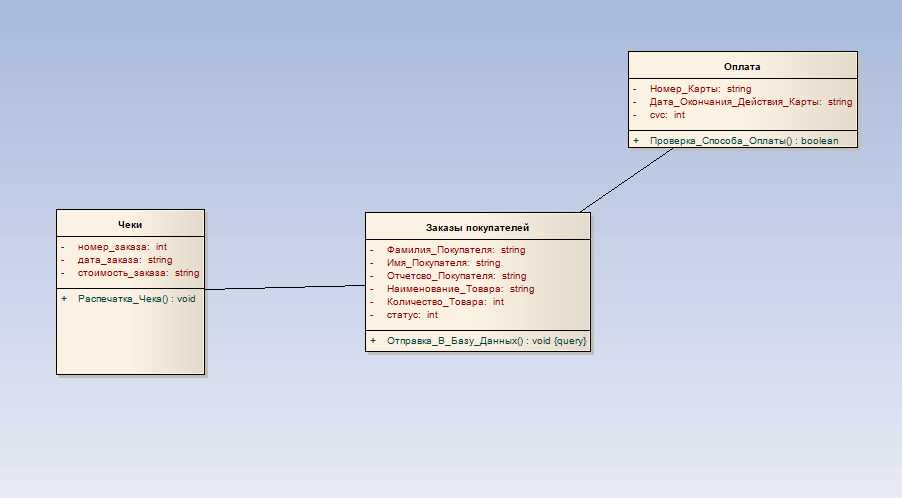


Рисунок 1 – Диаграмма классов

3. Контрольные вопросы

3.1 Что является основным средством для представления статических моделей?

Основным средством для представления статических моделей являются диаграммы классов.

Вершины диаграмм классов нагружены классами а дуги(ребра) – отношения между ними.

Используются в ходе анализа и в ходе проектирования для указания ролей и обязанностей сущностей и для фиксации структуры классов

3.2 Какие секции входят в графическое обозначение класса?

В первой секции обозначения класса могут находиться ссылки на стандартные шаблоны или абстрактные классы, от которых образован данный класс и, соответственно, от которых он наследует свойства и методы. В этой секции может приводиться информация о разработчике данного класса и статус состояния разработки, а также могут записываться и другие общие свойства этого класса, имеющие отношение к другим классам диаграммы или стандартным элементам языка UML.

Примерами имен классов могут быть такие существительные, как «Сотрудник», «Компания», «Руководитель», «Клиент», «Продавец», «Менеджер», «Офис» и многие другие, имеющие непосредственное отношение к моделируемой предметной области и функциональному назначению проектируемой системы.

Класс может не иметь экземпляров или объектов. В этом случае он называется абстрактным классом, а для обозначения его имени используется наклонный шрифт (курсив). В языке UML принято общее соглашение о том, что любой текст, относящийся к абстрактному элементу, записывается курсивом. Данное обстоятельство является семантическим аспектом описания соответствующих элементов языка UML.

Атрибуты класса

Во второй сверху секции прямоугольника класса записываются его атрибуты (attributes) или свойства. В языке UML принята определенная стандартизация записи атрибутов класса, которая подчиняется некоторым синтаксическим правилам. Каждому атрибуту класса соответствует отдельная строка текста, которая состоит из квантора видимости атрибута, имени атрибута, его кратности, типа значений атрибута и, возможно, его исходного значения:

[кратность]:

= {строка-свойство}

Квантор видимости может принимать одно из трех возможных значений и, соответственно, отображается при помощи специальных символов:

• Символ + обозначает атрибут с областью видимости типа общедоступный (public). Атрибут с этой областью видимости доступен или виден из любого другого класса пакета, в котором определена диаграмма.

• Символ # обозначает атрибут с областью видимости типа защищенный (protected). Атрибут с этой областью видимости недоступен или невиден для всех классов, за исключением подклассов данного класса.

• И, наконец, знак — обозначает атрибут с областью видимости типа закрытый (private). Атрибут с этой областью видимости недоступен или невиден для всех классов без исключения.

Квантор видимости может быть опущен. В этом случае его отсутствие просто означает, что видимость атрибута не указывается. Эта ситуация отличается от принятых по умолчанию соглашений в традиционных языках программирования, когда отсутствие квантора видимости трактуется как public или private. Однако вместо условных графических обозначений можно записывать соответствующее ключевое слово: public, protected, private.

3.3 Какие секции класса можно не показывать?

Изображая класс, не обязательно показывать сразу все его атрибуты и операции. Для конкретного представления, как правило, существенна только часть атрибутов и операций класса. В силу этих причин допускается упрощенное представление класса, то есть для графического представления выбираются только некоторые из его атрибутов. Если помимо указанных существуют другие атрибуты и операции, вы даете это понять, завершая каждый список многоточием.

3.4 Какие имеются разновидности области действия свойства (операции)?

***Операция (operation)*** - это сервис, предоставляемый каждым экземпляром или объектом *класса* по требованию своих клиентов, в качестве которых могут выступать другие объекты, в том числе и экземпляры данного *класса* .

*Операции класса* записываются в третьей сверху секции прямоугольника *класса*, которую часто называют секцией *операций*. Совокупность *операций* характеризует функциональный аспект поведения всех объектов данного *класса*. *Запись* *операций класса* в языке *UML* также стандартизована и подчиняется определенным синтаксическим правилам. При этом каждой *операции класса* соответствует отдельная строка, которая состоит из *квантора видимости* *операции*, имени *операции*, выражения типа возвращаемого *операцией* значения и, возможно, строка-свойство данной *операции*. Общий формат записи отдельной *операции класса* следующий:

<квантор видимости> <имя операции>(

список параметров):

<выражение типа возвращаемого значения>

{строка-свойство}

*Квантор видимости*, как и в случае *атрибутов класса*, может принимать одно из четырех возможных значений и, соответственно, отображается при помощи специального символа либо ключевого слова. Символ " + " обозначает *операцию* с областью видимости типа общедоступный ( public ). Символ " # " обозначает *операцию* с областью видимости типа защищенный ( protected ). Символ " - " используется для обозначения *операции* с областью видимости типа закрытый ( private ). И, наконец, символ " ~ " используется для обозначения *операции* с областью видимости типа пакетный ( package ).

*Квантор видимости* для *операции* может быть опущен. В этом случае его отсутствие просто означает, что видимость *операции* не указывается. Вместо условных графических обозначений также можно записывать соответствующее *ключевое слово*: public, protected, private, package.

Имя *операции* представляет собой строку текста, которая используется в качестве идентификатора соответствующей *операции* и поэтому должна быть уникальной в пределах данного *класса*. Имя *операции* - единственный обязательный элемент синтаксического обозначения *операции*, должно начинаться со строчной (малой) буквы, и, как правило, записываться без пробелов.

*Список* *параметров* является перечнем разделенных запятой формальных *параметров*, каждый из которых, в свою *очередь*, может быть представлен в следующем виде:

<направление параметра> <имя параметра>:

<выражение типа> =

<значение параметра по умолчанию>.

***Параметр (parameter)*** — спецификация переменной *операции*, которая может быть изменена, передана или возвращена.

*Параметр* может включать имя, тип, направление и *значение* *по* умолчанию. Направление *параметра* — есть одно из ключевых слов in, out или inout со значением in *по* умолчанию, в случае если вид *параметра* не указывается. Имя *параметра* есть *идентификатор* соответствующего формального *параметра*, при записи которого следуют правилам задания имен *атрибутов*. *Выражение* типа является спецификацией типа данных для допустимых значений соответствующего формального *параметра*. Наконец, *значение* *по* умолчанию в общем случае представляет собой некоторое конкретное *значение* для этого формального *параметра*.

*Выражение* типа возвращаемого значения также указывает на *тип данных* значения, которое возвращается объектом после выполнения соответствующей *операции*. Две точки и *выражение* типа возвращаемого значения могут быть опущены, если *операция* не возвращает никакого значения. Для указания нескольких возвращаемых значений данный элемент спецификации *операции* может быть записан в виде списка отдельных выражений.

*Операция* с областью действия на весь *класс* показывается подчеркиванием имени и строки выражения типа. В этом случае под областью действия *операции* понимаются все объекты этого *класса*. В этом случае вся строка записи *операции* подчеркивается.

Строка-свойство служит для указания значений свойств, которые могут быть применены к данной *операции*. Строка-свойство может отсутствовать, если свойства не специфицированы.

*Список* формальных *параметров* и тип возвращаемого значения не обязателен. *Квантор видимости* *атрибутов* и *операций* может быть указан в виде специального значка или символа, которые используются для графического представления моделей в инструментальном средстве. Еще раз следует напомнить, что имена *операций*, так же как *атрибутов* и *параметров*, записываются со строчной (малой) буквы, а их типы *параметров* — с заглавной (большой) буквы. При этом обязательной частью строки записи *операции* является наличие имени *операции* и круглых скобок.

3.5 Какой вид имеет форма представления параметра операции?

Общий синтаксис представления операции имеет вид

Видимость Имя (Список Параметров): ВозвращаемыйТип {Характеристики}