**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Школа: Инженерная школа энергетики

Направление: Прикладная информатика

Отделение: Электроэнергетики и электротехники

Лабораторная работа №5

Проектная документация

по дисциплине «*Основы объектно-ориентированного программирования*»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент гр. О-5КМ81 | \_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_ | С.С. Жидов |
|  | Подпись | Дата |  |
| Проверил преподаватель | \_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_ | А.А. Калентьев |
|  | Подпись | Дата |  |

Томск 2020 г.

Содержание

[Литературный обзор 3](#_Toc44062423)

[Диаграмма вариантов использования 5](#_Toc44062424)

[Диаграммы классов 6](#_Toc44062425)

[Описание классов 7](#_Toc44062426)

[Заключение 9](#_Toc44062427)

Литературный обзор

Диаграмма классов — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы (классов, типов и т. п.) Она содержит в себе также некоторые элементы поведения (например — операции), однако их динамика должна быть отражена на диаграммах других видов (диаграммах коммуникации, диаграммах состояний). Для удобства восприятия диаграмму классов можно также дополнить представлением пакетов, включая вложенные.

При представлении сущностей реального мира разработчику требуется отразить их текущее состояние, их поведение и их взаимные отношения. На каждом этапе осуществляется абстрагирование от маловажных деталей и концепций, которые не относятся к реальности (производительность, инкапсуляция, видимость и т. п.). Классы можно рассматривать с позиции различных уровней. Как правило, их выделяют три основных: аналитический уровень, уровень проектирования и уровень реализации:

* на уровне анализа класс содержит в себе только набросок общих контуров системы и работает как логическая концепция предметной области или программного продукта.
* на уровне проектирования класс отражает основные проектные решения касательно распределения информации и планируемой функциональности, объединяя в себе сведения о состоянии и операциях.
* на уровне реализации класс дорабатывается до такого вида, в каком он максимально удобен для воплощения в выбранной среде разработки; при этом не воспрещается опустить в нём те общие свойства, которые не применяются на выбранном языке программирования.

Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

При моделировании системы с помощью диаграммы прецедентов системный аналитик стремится:

* чётко отделить систему от её окружения;
* определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;
* определить в глоссарии предметной области понятия, относящиеся к детальному описанию функциональности системы (то есть прецедентов).

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ).

В соответствии с заданием ниже представлены диаграмма вариантов использования и диаграмма классов для программы, разработанной в ходе выполнения лабораторной работы №4.

# Диаграмма вариантов использования

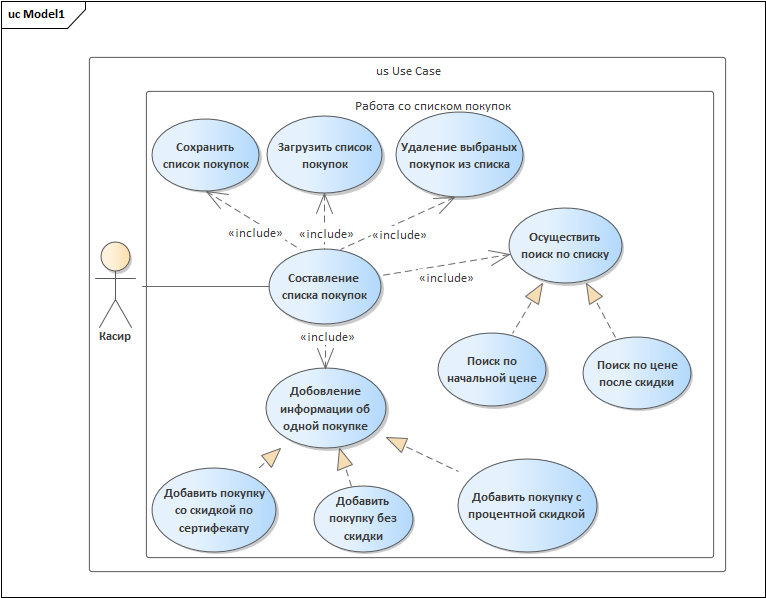


Рисунок 1 – Диаграмма вариантов использования

Диаграммы классов

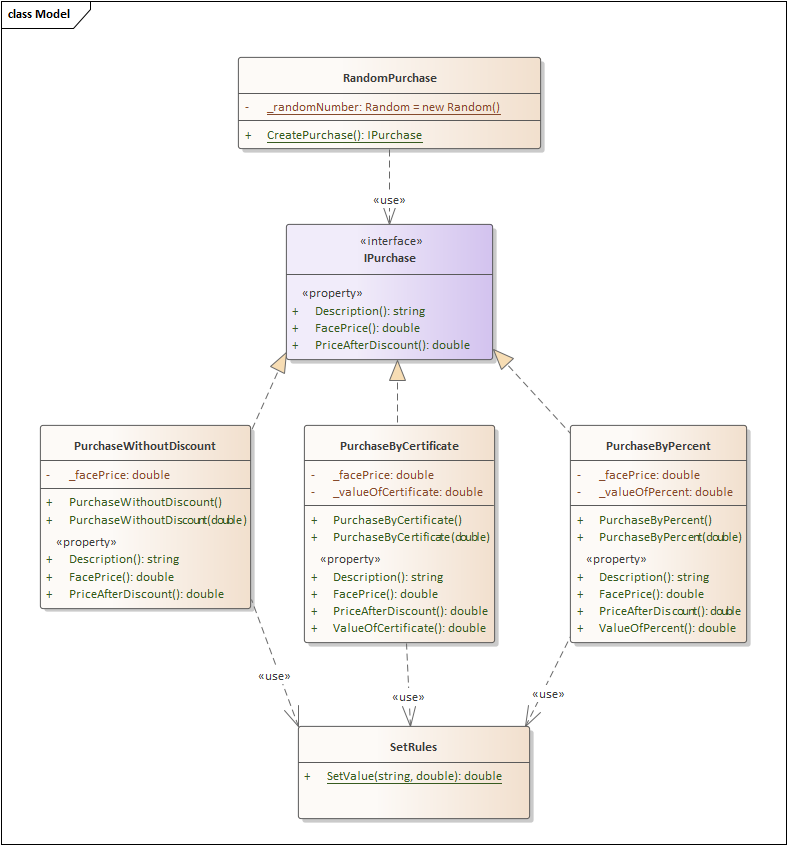


Рисунок 2 – Диаграмма классов библиотеки Model

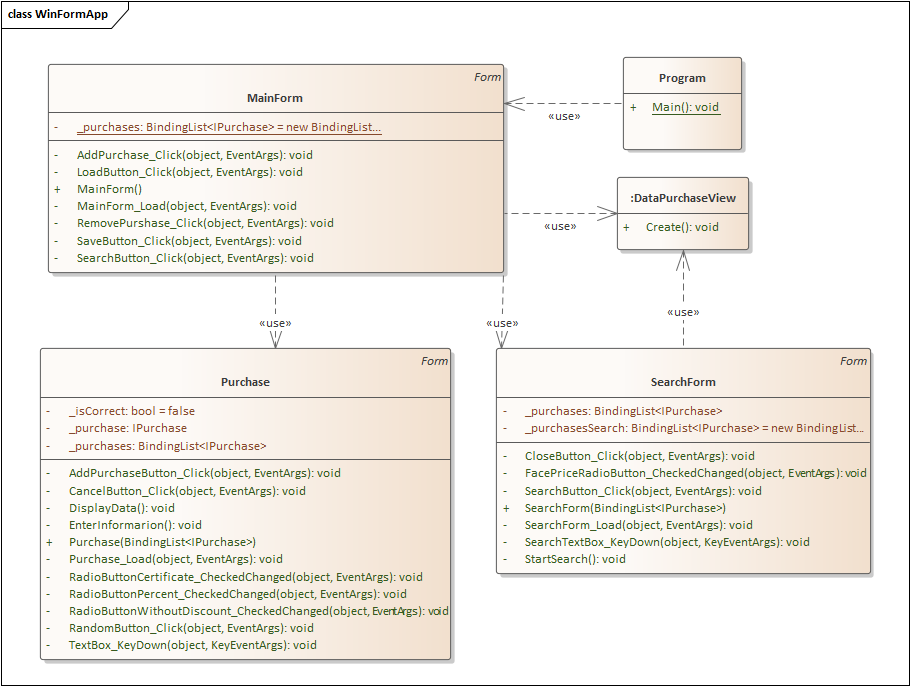


Рисунок 3 – Диаграмма классов для графического интерфейса

Описание классов

Таблица 1 – Описание интерфейса IPurchase

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание интерфейса | | |
| Интерфейс IPurchase – сущность для описания абстрактной покупки в программе. | | |
| Свойства | | |
| +FacePrice | double | Стартовая цена товара (до скидки) |
| +PriceAfterDiscount | double | Цена товара после применения скидки |
| +Description | string | Качественное и количественное описание скидки |

Таблица 2 – Описание класса PurchaseWithoutDiscount

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание Класс | | |
| Класс PurchaseWithoutDiscount – сущность для описания абстрактной покупки без скидки в программе. | | |
| Поля | | |
| -\_facePrice | double | Стартовая цена товара (до скидки) |
| Свойства | | |
| +FacePrice | double | Стартовая цена товара (до скидки) |
| +PriceAfterDiscount | double | Цена товара после применения скидки |
| +Description | string | Качественное и количественное описание скидки |
| Методы | | |
| #PurchaseWithoutDiscount (double) |  | Конструктор для создания новой покупки с помощью задания стартовой цены товара.  *Double* – Стартовая цена товара. |
| #PurchaseWithoutDiscount ( ) |  | Конструктор для создания новой покупки по умолчанию. |

Таблица 3 – Описание класса PurchaseByCertificate

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание Класс | | |
| Класс PurchaseByCertificate – сущность для описания абстрактной покупки со скидкой по сертификату в программе. | | |
| Поля | | |
| -\_facePrice | double | Стартовая цена товара (до скидки) |
| -\_valueOfCertificate | double | Размер скидки по сертификату |
| Свойства | | |
| +FacePrice | double | Стартовая цена товара (до скидки) |
| + ValueOfCertificate | double | Размер скидки по сертификату |
| +PriceAfterDiscount | double | Цена товара после применения скидки |
| +Description | string | Качественное и количественное описание скидки |
| Методы | | |
| # PurchaseByCertificate (double) |  | Конструктор для создания новой покупки с помощью задания стартовой цены товара.  *Double* – Стартовая цена товара. |
| # PurchaseByCertificate ( ) |  | Конструктор для создания новой покупки по умолчанию. |

Таблица 4 – Описание класса PurchaseByPercent

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| Описание Класс | | |
| Класс PurchaseByPercent – сущность для описания абстрактной покупки с процентной скидкой в программе. | | |
| Поля | | |
| -\_facePrice | double | Стартовая цена товара (до скидки) |
| -\_valueOfPercent | double | Размер процентной скидки |
| Свойства | | |
| +FacePrice | double | Стартовая цена товара (до скидки) |
| + ValueOfPercent | double | Размер процентной скидки |
| +PriceAfterDiscount | double | Цена товара после применения скидки |
| +Description | string | Качественное и количественное описание скидки |
| Методы | | |
| # PurchaseByPercent (double) |  | Конструктор для создания новой покупки с помощью задания стартовой цены товара.  *Double* – Стартовая цена товара. |
| # PurchaseByPercent ( ) |  | Конструктор для создания новой покупки по умолчанию. |

Заключение

1. В ходе работы изучены основные диаграммы UML нотации.
2. Составлена диаграмма вариантов использования.
3. Составлены диаграммы классов.
4. Составлено описание классов, образующих связь типа «общее-частное».
5. Получены базовые навыки работы с программой среде Enterprise Architect.