Лабораторная работа №4
Диаграмма классов

Выполнила: Крабу Кира, 15.11Д-БИЦТ09/21б

Вариант 10.

Формирование чека для оплаты покупок в супермаркете

Диаграмма вариантов использования

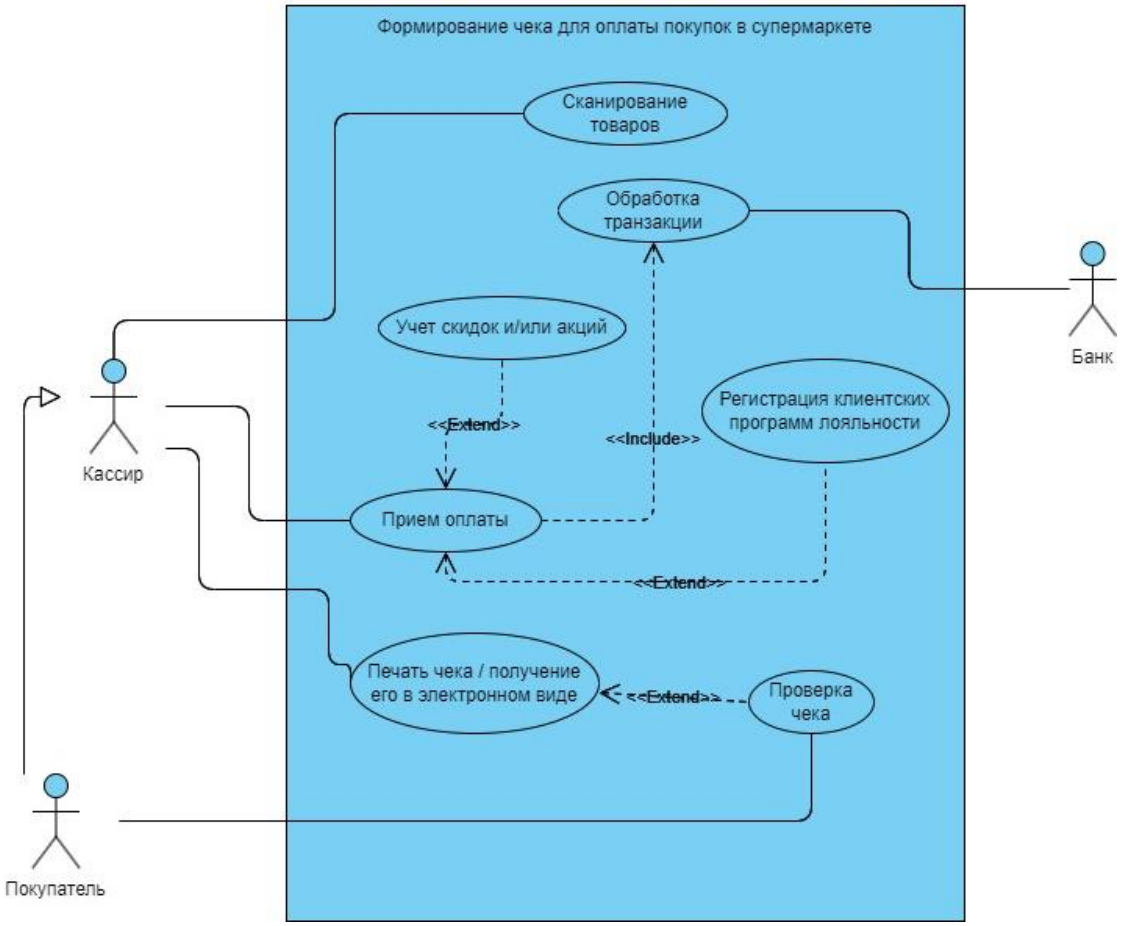

Рисунок 1. Диаграмма вариантов использования «Формирования чека для оплаты покупок в супермаркете»

Диаграмма классов

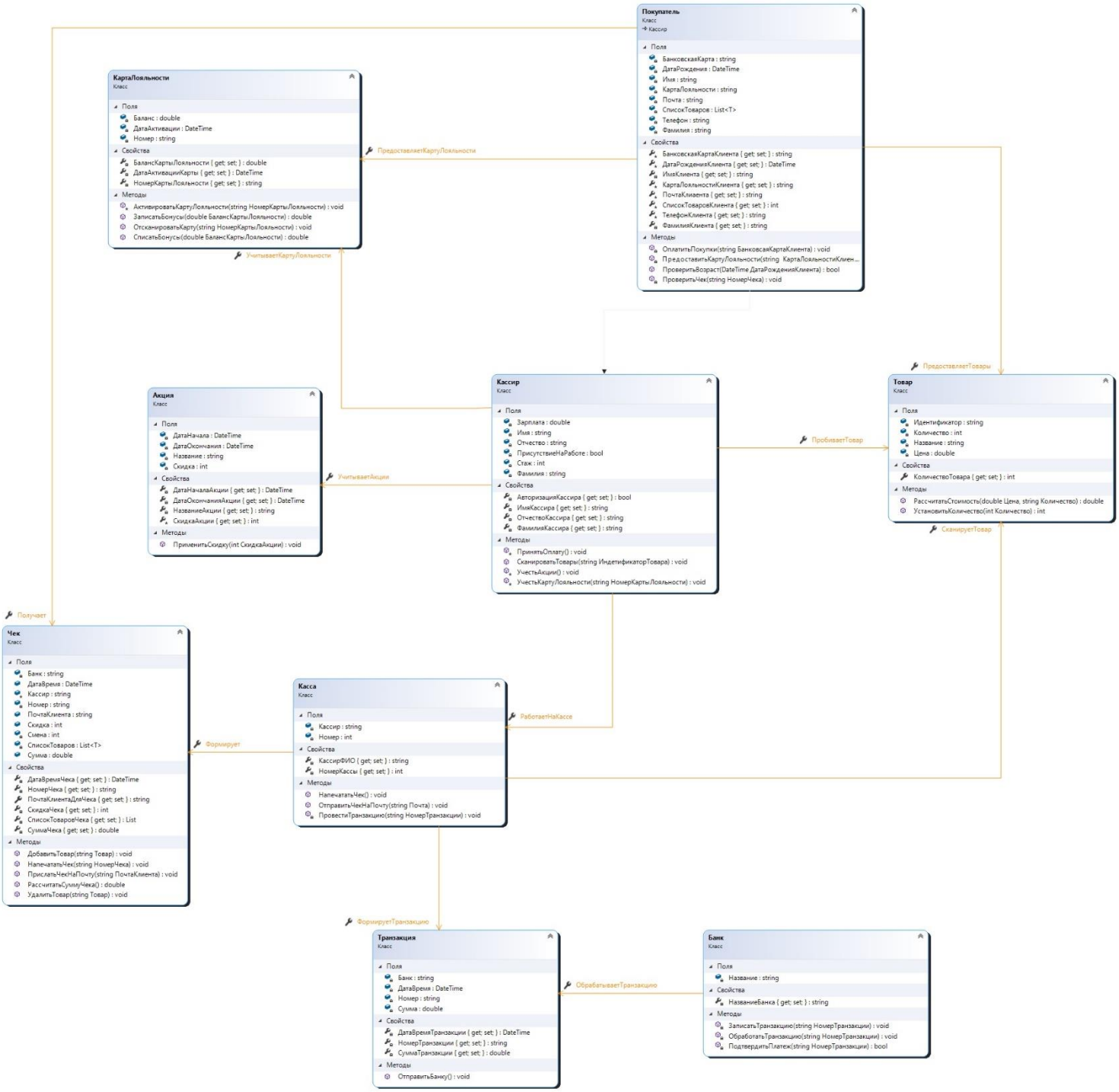

Рисунок 2. Диаграмма классов «Формирования чека для оплаты покупок в супермаркете»

Диаграмма объектов

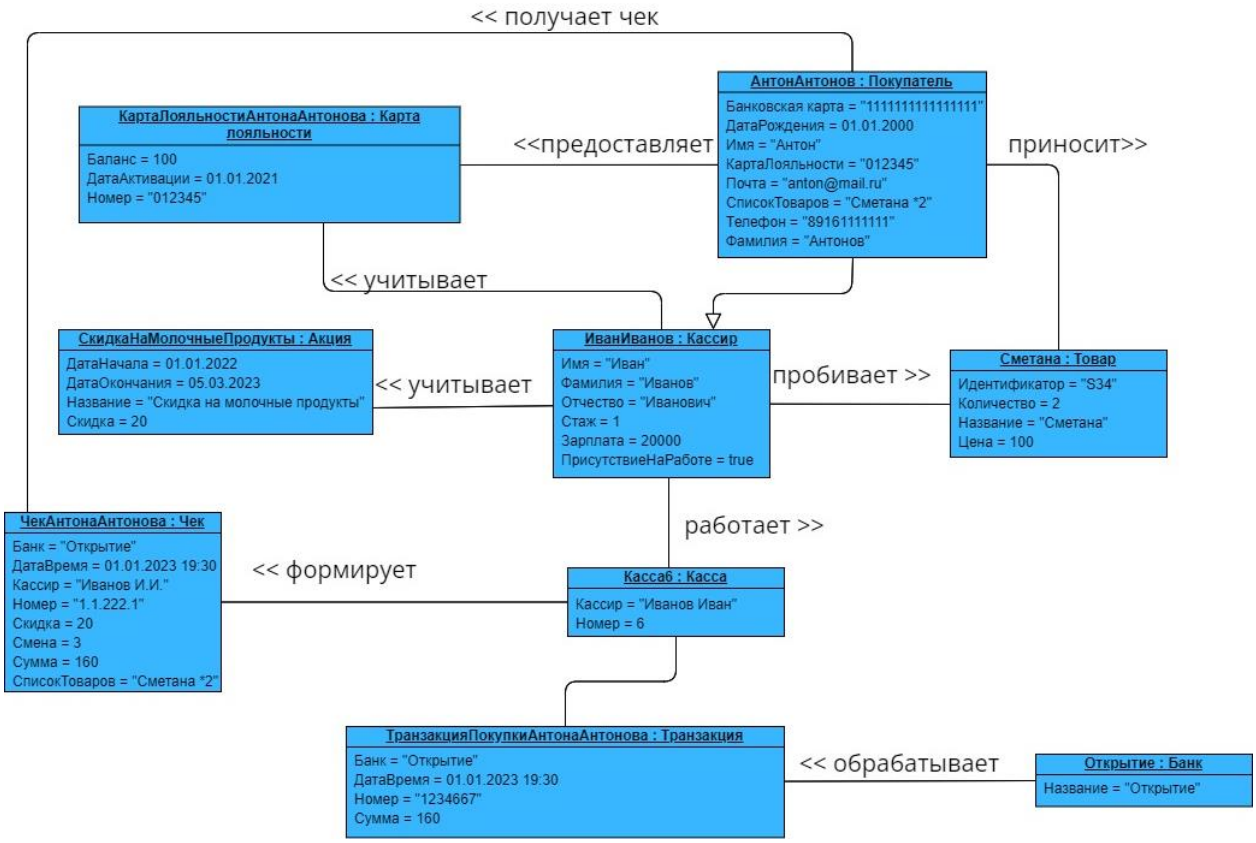

Рисунок 3. Диаграмма объектов «Формирования чека для оплаты покупок в супермаркете»

Коды

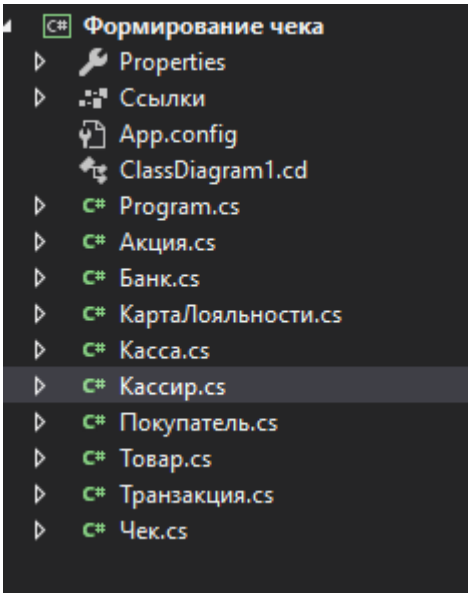

Рисунок 4. Окно решений проекта

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Формирование_чека
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Акция.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Акция
    {
        private DateTime ДатаНачала;
        private DateTime ДатаОкончания;
        private string Название;
        private int Скидка;

        private string НазваниеАкции
        {
            get => default;
            set
            {
            }
        }

        protected int СкидкаАкции
        {
            get => default;
            set
            {
            }
        }

        private DateTime ДатаНачалаАкции
        {
            get => default;
            set
            {
            }
        }

        private DateTime ДатаОкончанияАкции
        {
            get => default;
            set
            {
            }
        }

        public void ПрименитьСкидку(int СкидкаАкции)
        {
```

```
            throw new System.NotImplementedException();
        }
    }
}
```

Банк.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Банк
    {
        private string Название;

        private string НазваниеБанка
        {
            get => default;
            set
            {
            }
        }

        public Транзакция ОбрабатываетТранзакцию
        {
            get => default;
            set
            {
            }
        }

        private void ОбработатьТранзакцию(string НомерТранзакции)
        {
            throw new System.NotImplementedException();
        }

        private void ЗаписатьТранзакцию(string НомерТранзакции)
        {
            throw new System.NotImplementedException();
        }

        private bool ПодтвердитьПлатеж(string НомерТранзакции)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

КартаЛояльности.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class КартаЛояльности
    {
        private double Баланс;
        private DateTime ДатаАктивации;
        private string Номер;

        private string НомерКартыЛояльности
        {
```

```
            get => default;
            set
            {
            }
        }

        private double БалансКартыЛояльности
        {
            get => default;
            set
            {
            }
        }

        private DateTime ДатаАктивацииКарты
        {
            get => default;
            set
            {
            }
        }

        public void ОтсканироватьКарту(string НомерКартыЛояльности)
        {
            throw new System.NotImplementedException();
        }

        public double СписатьБонусы(double БалансКартыЛояльности)
        {
            throw new System.NotImplementedException();
        }

        public double ЗаписатьБонусы(double БалансКартыЛояльности)
        {
            throw new System.NotImplementedException();
        }

        protected void АктивироватьКартуЛояльности(string НомерКартыЛояльности)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Касса.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Касса
    {
        private string Кассир;
        private int Номер;

        private int НомерКассы
        {
            get => default;
            set
            {
            }
        }

        private string КассирФИО
        {
```

```
            get => default;
            set
            {
            }
        }

        public Товар СканируетТовар
        {
            get => default;
            set
            {
            }
        }

        public Транзакция ФормируетТранзакцию
        {
            get => default;
            set
            {
            }
        }

        public Чек Формирует
        {
            get => default;
            set
            {
            }
        }

        public void НапечататьЧек()
        {
            throw new System.NotImplementedException();
        }

        public void ОтправитьЧекНаПочту(string Почта)
        {
            throw new System.NotImplementedException();
        }

        private void ПровестиТранзакцию(string НомерТранзакции)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Кассир.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Кассир
    {
        private double Зарплата;
        private int Стаж;
        private string Имя;
        private string Фамилия;
        private string Отчество;
        private bool ПрисутствиеНаРаботе;

        private string ИмяКассира
        {
            get => default;
```

```
        set
        {
        }
    }

    private string ФамилияКассира
    {
        get => default;
        set
        {
        }
    }

    private string ОтчествоКассира
    {
        get => default;
        set
        {
        }
    }

    private bool АвторизацияКассира
    {
        get => default;
        set
        {
        }
    }

    public Товар ПробиваетТовар
    {
        get => default;
        set
        {
        }
    }

    public Касса РаботаетНаКассе
    {
        get => default;
        set
        {
        }
    }

    public Акция УчитываетАкции
    {
        get => default;
        set
        {
        }
    }

    public КартаЛояльности УчитываетКартуЛояльности
    {
        get => default;
        set
        {
        }
    }

    protected void ПринятьОплату()
    {
        throw new System.NotImplementedException();
    }

    protected void УчестьАкции()
    {
        throw new System.NotImplementedException();
```

```
        }

        protected void УчестьКартуЛояльности(string НомерКартыЛояльности)
        {
            throw new System.NotImplementedException();
        }

        public void СканироватьТовары(string ИндетификаторТовара)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Покупатель.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Покупатель : Кассир
    {
        private string БанковскаяКарта;
        private DateTime ДатаРождения;
        private string Имя;
        private string Фамилия;
        private string КартаЛояльности;
        private string Почта;
        private string Телефон;
        private List<T> СписокТоваров;

        private string ИмяКлиента
        {
            get => default;
            set
            {
            }
        }

        private string ФамилияКлиента
        {
            get => default;
            set
            {
            }
        }

        protected DateTime ДатаРожденияКлиента
        {
            get => default;
            set
            {
            }
        }

        protected string ПочтаКлиаента
        {
            get => default;
            set
            {
            }
        }

        protected string ТелефонКлиента
        {
```

```
        get => default;
        set
        {
        }
    }

    protected string КартаЛояльностиКлиента
    {
        get => default;
        set
        {
        }
    }

    protected string БанковскаяКартаКлиента
    {
        get => default;
        set
        {
        }
    }

    protected int СписокТоваровКлиента
    {
        get => default;
        set
        {
        }
    }

    public КартаЛояльности ПредоставляетКартуЛояльности
    {
        get => default;
        set
        {
        }
    }

    public Товар ПредоставляетТовары
    {
        get => default;
        set
        {
        }
    }

    public Чек Получает
    {
        get => default;
        set
        {
        }
    }

    public bool ПроверитьВозраст(DateTime ДатаРожденияКлиента)
    {
        throw new System.NotImplementedException();
    }

    private void ОплатитьПокупки(string БанковсаяКартаКлиента)
    {
        throw new System.NotImplementedException();
    }

    private void ПредоставитьКартуЛояльности(string КартаЛояльностиКлиента)
    {
        throw new System.NotImplementedException();
    }
```

```
        private void ПроверитьЧек(string НомерЧека)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Товар.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Товар
    {
        private string Идентификатор;
        private int Количество;
        private string Название;
        private double Цена;

        public int КоличествоТовара
        {
            get => default;
            set
            {
            }
        }

        public int УстановитьКоличество(int Количество)
        {
            throw new System.NotImplementedException();
        }

        public double РассчитатьСтоимость(double Цена, string Количество)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Транзакция.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Транзакция
    {
        private string Банк;
        private DateTime ДатаВремя;
        private string Номер;
        private double Сумма;

        private string НомерТранзакции
        {
            get => default;
            set
            {
            }
        }

        private double СуммаТранзакции
```

```
        {
            get => default;
            set
            {
            }
        }

        private DateTime ДатаВремяТранзакции
        {
            get => default;
            set
            {
            }
        }

        public void ОтправитьБанку()
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Чек.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Формирование_чека
{
    public class Чек
    {
        private string Банк;
        protected string Кассир;
        private int Смена;
        private string Номер;
        public DateTime ДатаВремя;
        public string ПочтаКлиента;
        public int Скидка;
        private List<T> СписокТоваров;
        public double Сумма;

        private string НомерЧека
        {
            get => default;
            set
            {
            }
        }

        private DateTime ДатаВремяЧека
        {
            get => default;
            set
            {
            }
        }

        private List СписокТоваровЧека
        {
            get => default;
            set
            {
            }
        }

        private double СуммаЧека
```

```
        {
            get => default;
            set
            {
            }
        }

        public string ПочтаКлиентаДляЧека
        {
            get => default;
            set
            {
            }
        }

        private int СкидкаЧека
        {
            get => default;
            set
            {
            }
        }

        public void ДобавитьТовар(string Товар)
        {
            throw new System.NotImplementedException();
        }

        public void УдалитьТовар(string Товар)
        {
            throw new System.NotImplementedException();
        }

        public double РассчитатьСуммуЧека()
        {
            throw new System.NotImplementedException();
        }

        public void НапечататьЧек(string НомерЧека)
        {
            throw new System.NotImplementedException();
        }

        public void ПрислатьЧекНаПочту(string ПочтаКлиента)
        {
            throw new System.NotImplementedException();
        }
    }
}
```

Контрольные вопросы

1. Назначение диаграммы классов?

Диаграмма классов (class diagram) служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

2. Как обозначается класс и что в себя включает?

Класс (class) в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линиями на разделы или секции. В этих разделах могут указываться имя класса, атрибуты (переменные) и операции (методы).

Рисунок 5. Класс в диаграмме классов

3. Что такое атрибут в диаграмме классов?

Атрибут в диаграмме классов — это содержательная характеристика класса, описывающая множество значений, которые могут принимать отдельные объекты этого класса. Атрибут класса служит для представления отдельного свойства или признака, который является общим для всех объектов данного класса. Каждый атрибут имеет имя и тип данных. Тип данных может быть примитивным (например, целочисленным, строковым) или ссылочным, то есть типом другого класса.

4. Как обозначаются области видимости у атрибута?

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.
- Знак "-" обозначает атрибут с областью видимости типа закрытый (private). Атрибут с этой областью видимости недоступен или невиден для всех классов без исключения.

5. Что такое операция в диаграмме классов?

Операция (operation) представляет собой некоторый сервис, предоставляющий каждый экземпляр класса по определенному требованию. Совокупность операций характеризует функциональный аспект поведения класса.

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.

6. Перечислить виды отношений между классами, охарактеризовать каждое из них?

Базовыми отношениями или связями в языке UML являются:

- Отношение зависимости (dependency relationship). Отношение зависимости используется в такой ситуации, когда некоторое изменение одного элемента модели может потребовать изменения другого зависимого от него элемента модели.
- Отношение ассоциации (association relationship). Отношение ассоциации соответствует наличию некоторого отношения между классами. Данное отношение обозначается сплошной линией с дополнительными специальными символами, которые характеризуют отдельные свойства конкретной ассоциации.
- Отношение обобщения (generalization relationship). Отношение обобщения является обычным таксономическим отношением между более общим элементом (родителем или предком) и более частным или специальным элементом (дочерним или потомком).
- Отношение реализации (realization relationship). Это отношение между двумя элементами модели, где один элемент (клиент) реализует (выполняет) поведение, которое определяет другой элемент (поставщик).
- Отношение агрегации. Отношение агрегации имеет место между несколькими классами в том случае, если один из классов представляет собой некоторую сущность, включающую в себя в качестве составных частей другие сущности («часть-целое»). Части системы никак не обязаны наследовать ее свойства и поведение, поскольку являются вполне самостоятельными сущностями.
- Отношение композиции. Отношение композиции является частным случаем отношения агрегации. Это отношение служит для выделения специальной формы отношения «часть-целое», при которой составляющие части в некотором смысле находятся внутри целого. Специфика взаимосвязи между ними заключается в том, что части не могут выступать в отрыве от целого, т. е. с уничтожением целого уничтожаются и все его составные части.

7. Как обозначается объект (экземпляр) класса?

Объект (object) является отдельным экземпляром класса, который создается на этапе выполнения программы. Он имеет свое собственное имя и конкретные значения атрибутов. Для графического изображения объектов используется такой же символ прямоугольника, что и для классов. Отличия проявляются при указании имен объектов, которые в случае объектов обязательно подчеркиваются

Рисунок 6. Объекты класса