*Диаграммы классов* (**class diagram**) – это UML-диаграммы, представляющие общую структуру иерархии классов системы, их взаимосвязи, атрибуты и поля, методы и интерфейсы.

*Назначение*:

1. Документирование
2. Визуализация
3. Конструирование путем прямого или обратного инжиниринга

*Основная задача диаграммы классов*: графическое представление структуры декларативных элементов системы (классов, типов и т.д.), описание типов и их поведение, то есть операций.

*Пример case-средств*:

* Visual Paradigm
* Visual Studio (создали=>добавить=>создать элемент=>диаграмма классов)

*Средства рисования схем*:

* MS Visio (тип диаграмм=>статическая структура UML)
* draw.io (диаграмма класса)

**Описание типа**

*Тип описывается в виде прямоугольника и разделен на 3 секции*:

1. В верхней части название типа (писать имя по центру, полужирным). Если название курсивом, то класс абстрактный. Над названием типа может быть уточнение\стереотип (Пример: <<Интерфейс>>, <<Перечисление>>);
2. В центральной секции атрибуты или поля (по левому краю). В них может указываться тип или значение по умолчанию;
3. В нижней секции операции типа (всё что содержит какие-либо действия).

Т.к. свойства есть не во всех языках программирования то в диаграмме они могут быть представлены как два метода GetИмяСвойства() : типСвойства и SetИмяСвойства(value : типСвойства) : void.

Указание модификаторов перед атрибутами и модификациями:

**-** private

**#** protected

**+** public

Указание модификаторов у параметров метода:

**in** – входной параметр

**out** – выходной параметр

**inout** (ref) - передается по ссылке

+GetSum(a : int, b : int, out result : int) : void

Или

+GetSum(a : int, b : int = 1) : int

Область действия для элементов класса:

1. Экземпляр (на диаграмме не помечаются);
2. Класс – это элемент с модификатором static (в диаграмме подчёркиваются);
3. Связи\Взаимосвязи – это особый тип логических отношений между сущностями показанных на диаграммах класса и объекта.

Виды отношений

…выписать из тетради п.1…

*Зависимость*:

Объект первого класса используется как параметр метода не является атрибутом класса. Стрелка указывает на тип, который будет использоваться как тип параметра метода.

…выписать из тетради п.2…

*Агрегация*:

Объект одного класс является полем\частью другого класса. Передается как параметр метода (т.е. создается вне класса, в который передан). Ромб рядом с агрегатором.

…выписать из тетради п.3…

*Композиция*:

Объект является контейнером для объектов другого типа (объекты создаются внутри объекта-контейнера). Ромб рядом с контейнером.

…выписать из тетради п.4…

Композиция:

public void AddToy(string toyName)

{

Toy toy = new() {Name = toyName};

toys.Add(toy);

}

Агрегация:

public void AddToy(Toy toy)

{

toys.Add(toy);

}

При удалении объекта агрегатора, объекты, которые передавались в параметрах, продолжают существовать

Ассоциация – это общий случай для композиции и агрегации

…выписать из тетради п.5…

Мощность, или кратность отношений, или взаимосвязь. Связь с 0 или одним объектом. 1 с одним и только одним объектом

….выписать из тетради п.6…