

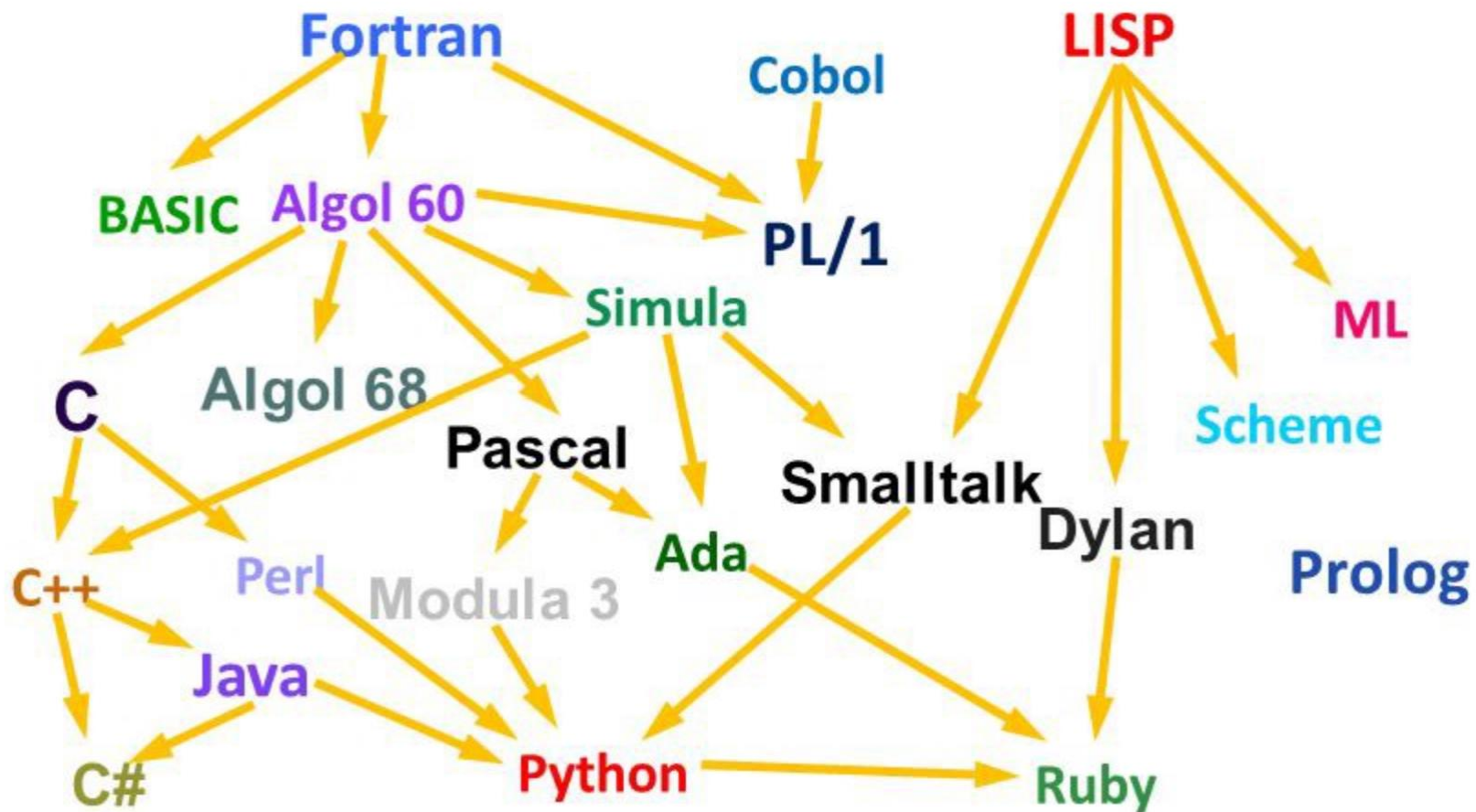
**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

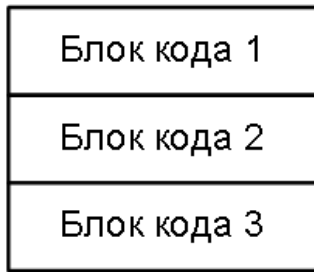
**КАФЕДРА
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ**

ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ

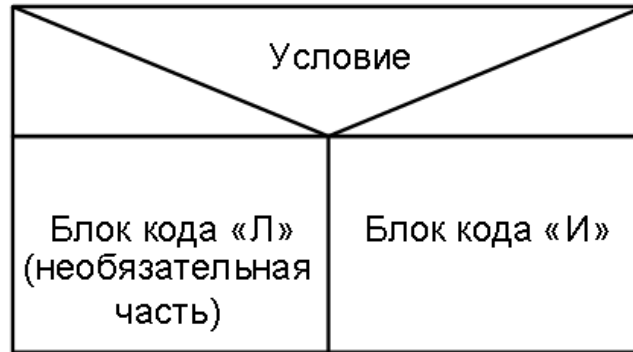
2022



Диаграммы Насси-Шнайдермана



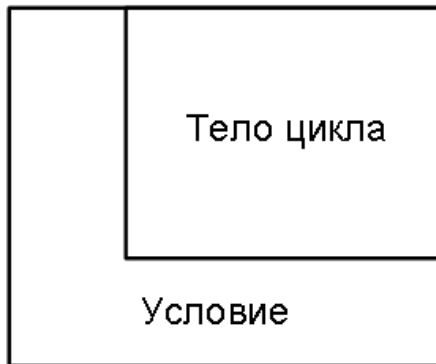
Последовательное
выполнение алгоритма



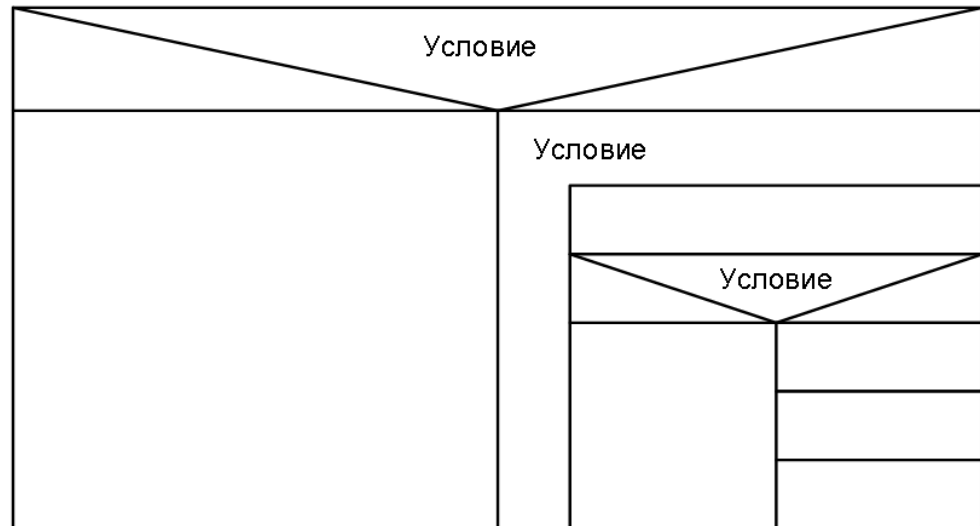
Ветвящаяся структура



Цикл с предусловием



Цикл с постусловием



Пример вложенных структур диаграмм Насси-Шнайдермана

Общефилософское определение класса и объекта

Объект – любая физическая или абстрактная четко идентифицируемая сущность.

Класс – конечная или бесконечная совокупность объектов, выделенная по определенному признаку (признакам) и мыслимая как единое целое.

Программные класс и объект

Программный объект является по отношению к объекту предметной области моделью.

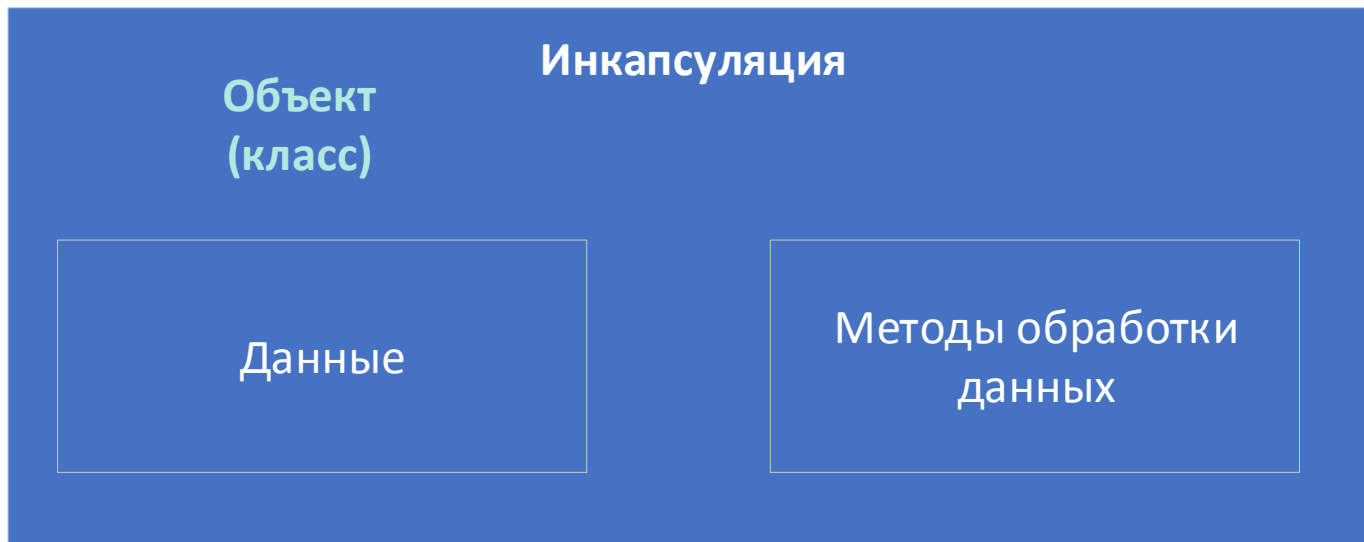
Программный класс является определением объекта.

Структурное программирование

Данные

Методы обработки
данных

Объектно-ориентированное программирование



Инкапсуляция – это упаковка данных и функций в один компонент (например, класс) и последующий контроль доступа к этому компоненту, создавая тем самым «чёрный ящик» из объекта. По этой причине, пользователю необходимо знать только интерфейс этого класса (то есть данные и функции, предоставляемые для взаимодействия с классом извне), а не то, как он реализован внутри.

```
public class Транспортное_средство
{
    public int Текущая_скорость { get; set; }

    public virtual string Получить_параметры_движения()
    {
        return "Текущая скорость: " + Текущая_скорость;
    }
}
```

```
public class Автомобиль : Транспортное_средство
{
    public int Текущая_передача { get; set; }

    public override string Получить_параметры_движения()
    {
        return base.Получить_параметры_движения()
            + ", текущая передача: " + Текущая_передача;
    }
}
```



```
public class Самолет : Транспортное_средство
{
    public int Текущая_высота { get; set; }

    public override string Получить_параметры_движения()
    {
        return "Полет на высоте "
            + Текущая_высота
            + " со скоростью " + Текущая_скорость;
    }
}
```

```
List<Транспортное_средство> транспорт = new List<Транспортное_средство>();

транспорт.Add(new Автомобиль { Текущая_скорость = 80, Текущая_передача = 4 });
транспорт.Add(new Автомобиль { Текущая_скорость = 120, Текущая_передача = 4 });
транспорт.Add(new Самолет { Текущая_скорость = 860, Текущая_высота = 10500 });
транспорт.Add(new Автомобиль { Текущая_скорость = 50, Текущая_передача = 3 });
транспорт.Add(new Самолет { Текущая_скорость = 540, Текущая_высота = 4500 });

List<string> параметры_движения = new List<string>();

foreach (Транспортное_средство транспортное_средство in транспорт)
{
    параметры_движения.Add(транспортное_средство.Получить_параметры_движения());
}

GridView1.DataSource = параметры_движения;
Page.DataBind();

Label1.Text = транспорт[0].Текущая_скорость.ToString();
Label2.Text = ((Автомобиль)транспорт[0]).Текущая_передача.ToString();
Label3.Text = ((Самолет)транспорт[2]).Текущая_высота.ToString();
```