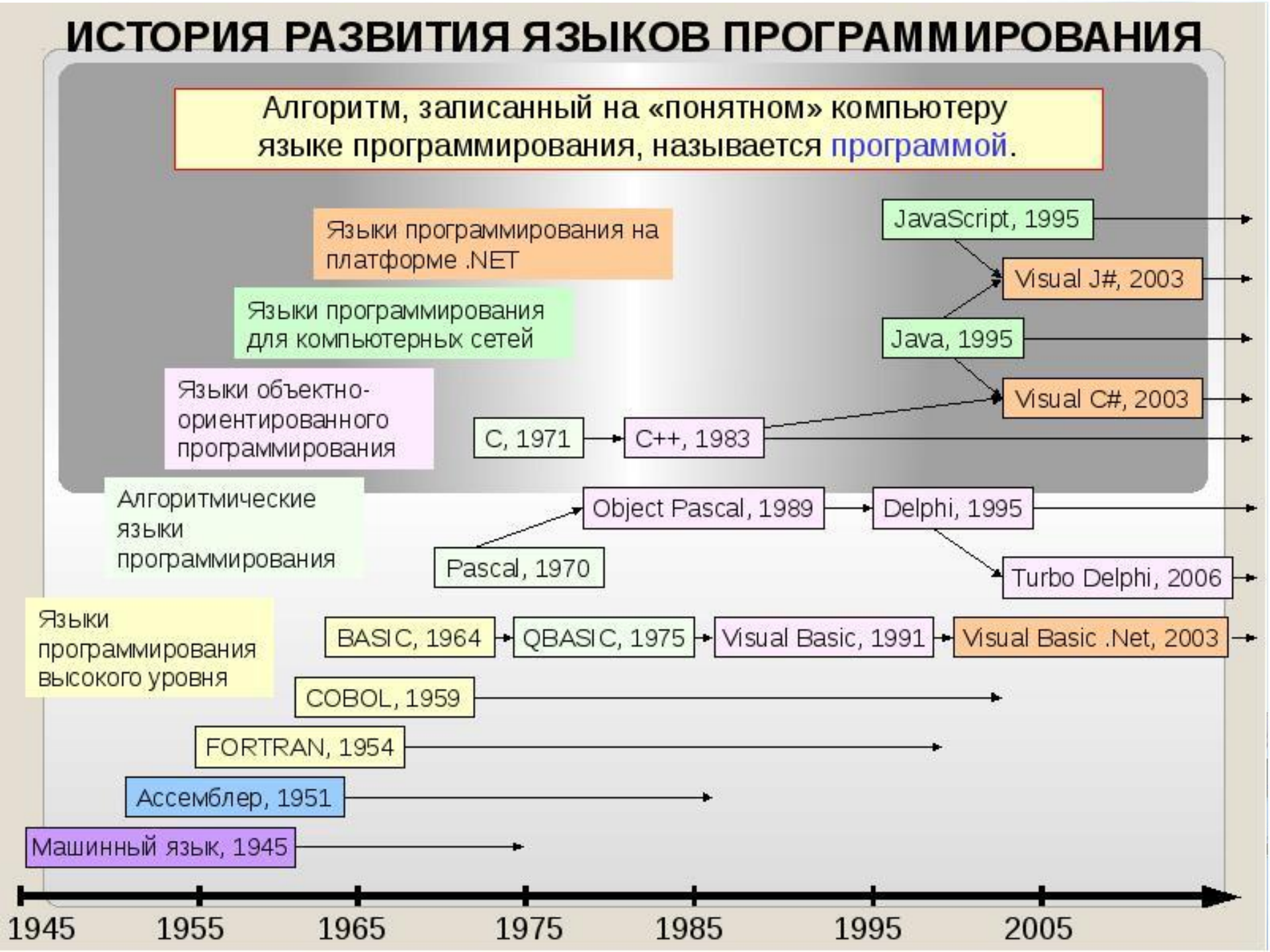
Принципы ООП. Основные понятия.

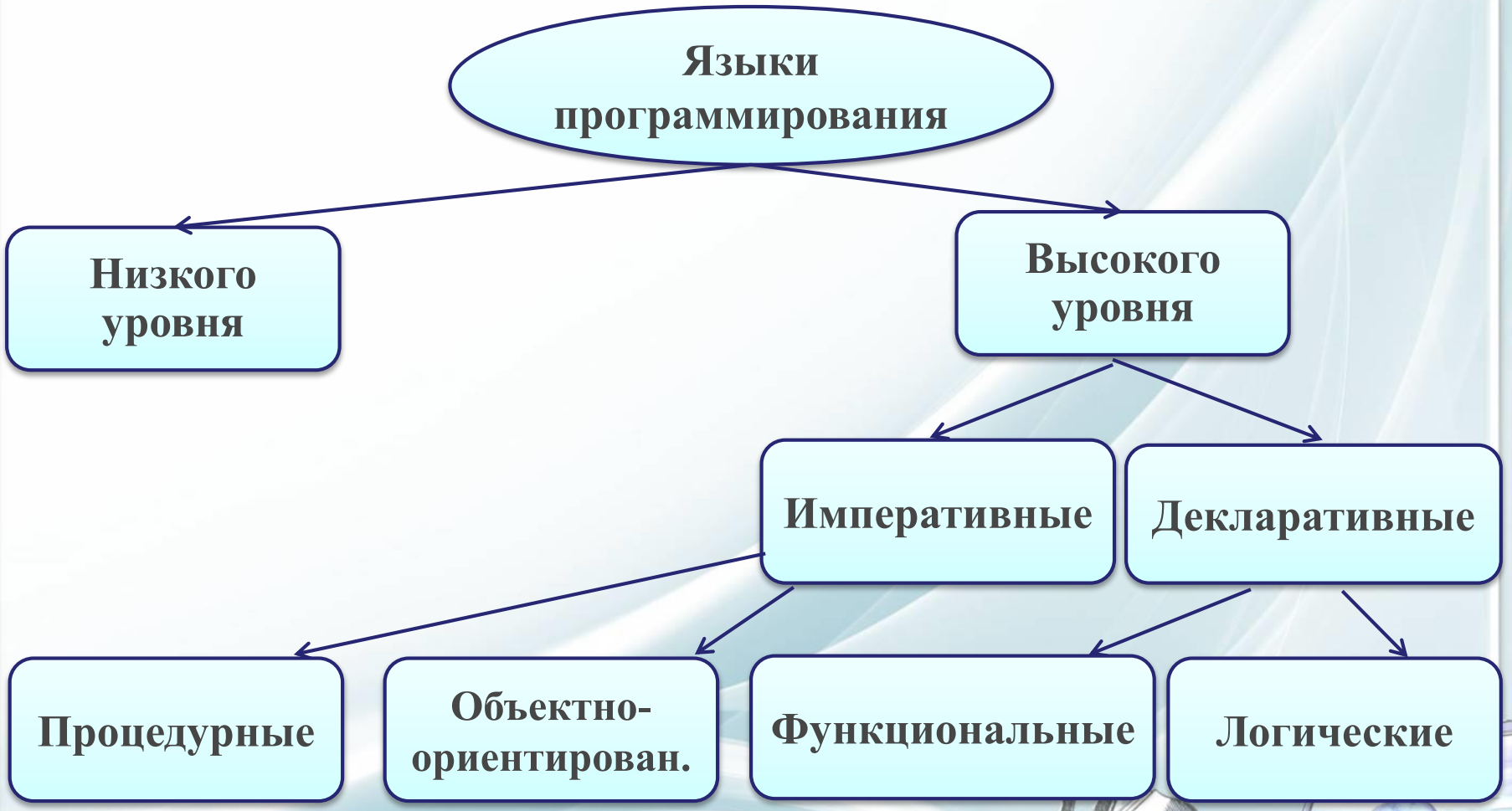
Объектно-ориентированное программирование

ИСТОРИЯ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется программой.

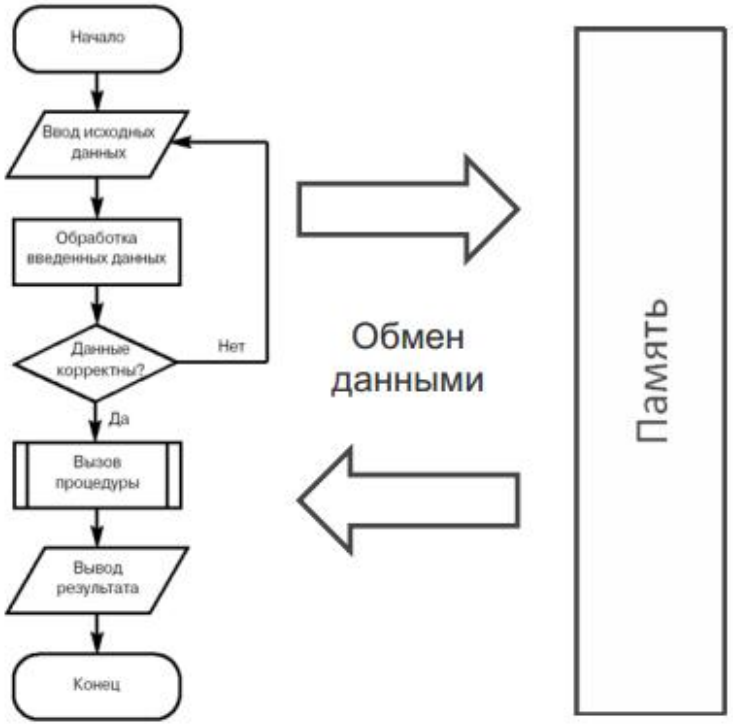


Классификация языков программирования



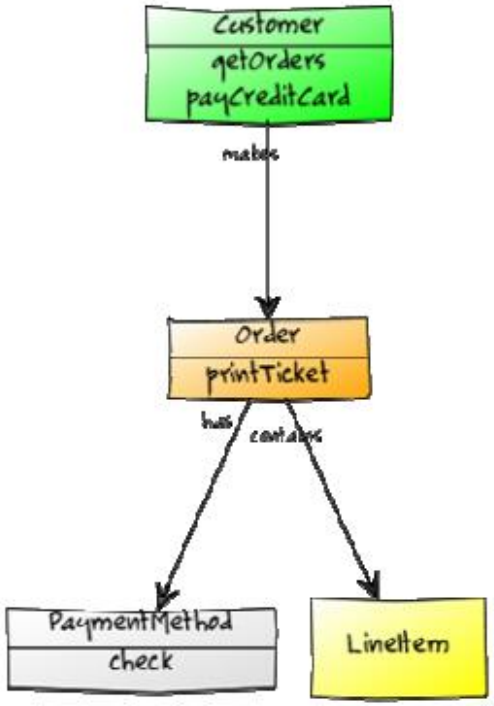
Объектно-ориентированное программирование

Процедурный подход



Программа – алгоритм последовательного вызова проедур изменения данных в памяти.

Объектно-ориентированный



Программа – взаимодействие объектов, компонентов, отсылка и обработка событий.

Достоинства и недостатки ООП

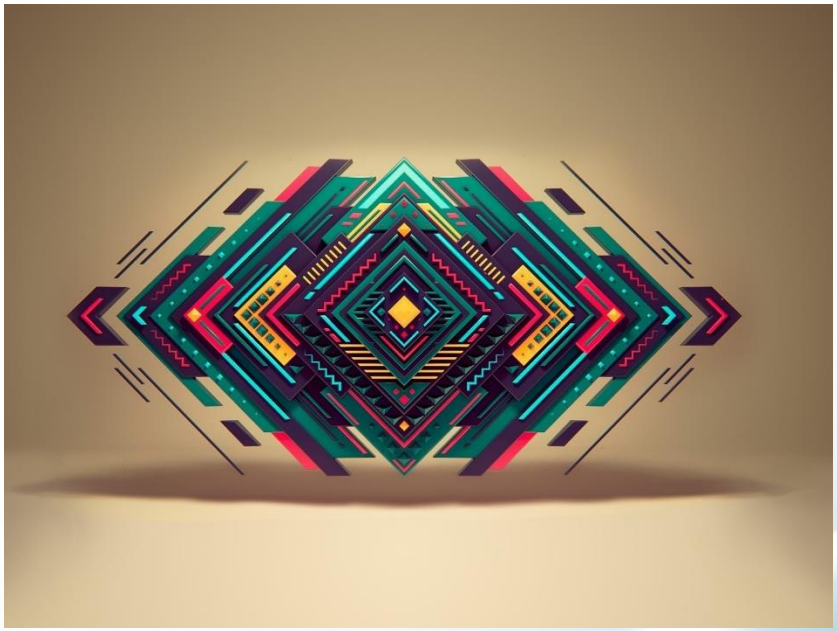
Достоинства

- Конструирование из простых компонент (абстракция).
- Данные связаны с операциями обработки.
- Инкапсуляция делает код более безопасным.
- Повторное использование компонент.
- Обобщенные алгоритмы.
- Изменение поведения во время выполнения (полиморфизм).
- Создание полуфабрикатов или фреймворков.

Недостатки

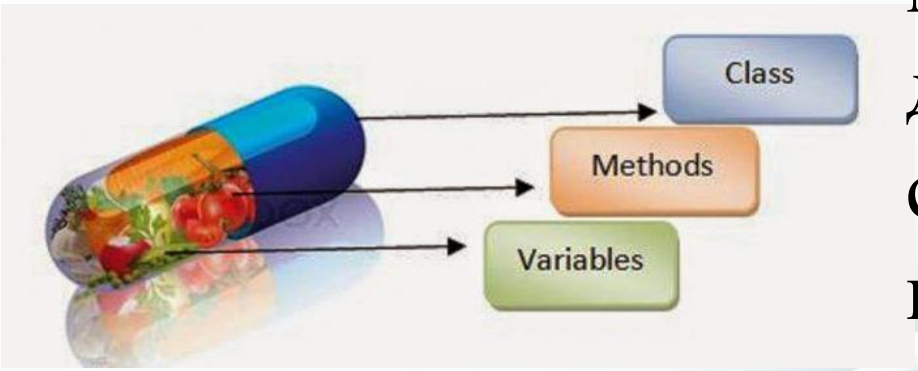
- Необходимость изучения концепций ООП
- Обилие библиотек компонентов повторного использования.
- Проектирование классов сложный процесс.
- Меньшее быстродействие
- Больший расход памяти
- Излишняя универсальность

Механизмы ООП: ***Абстракция***



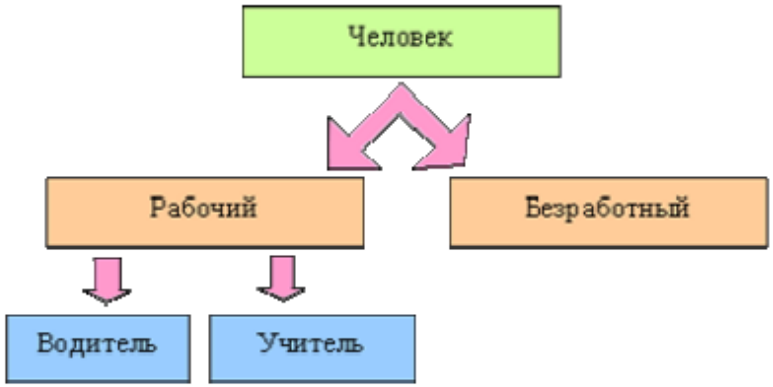
- придание объекту характеристик, которые чётко определяют его концептуальные границы, отличая от всех других объектов.

Механизмы ООП: ***Инкапсуляция***



- использование модификаторов доступа с целью сокрытия частей программного кода от конечного пользователя.

Механизмы ООП: ***Наследование***



- **Наследование** – описание нового класса на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.
  - Позволяет избавиться от дублирования кода.
  - Позволяет добавить новую функциональность в класс.
  - Позволяет описать отношения обобщения

```
class Грузовик extends Автомобиль {
    Кузов кузов;
    процедура загрузить(груз) {
        кузов.загрузить(груз);
        кузов.проверитьПерегруз();
    }
}

Грузовик мойГрузовик = new Грузовик();
мойГрузовик.загрузить(помидоры);
мойГрузовик.добавитьГазу(немного);
```

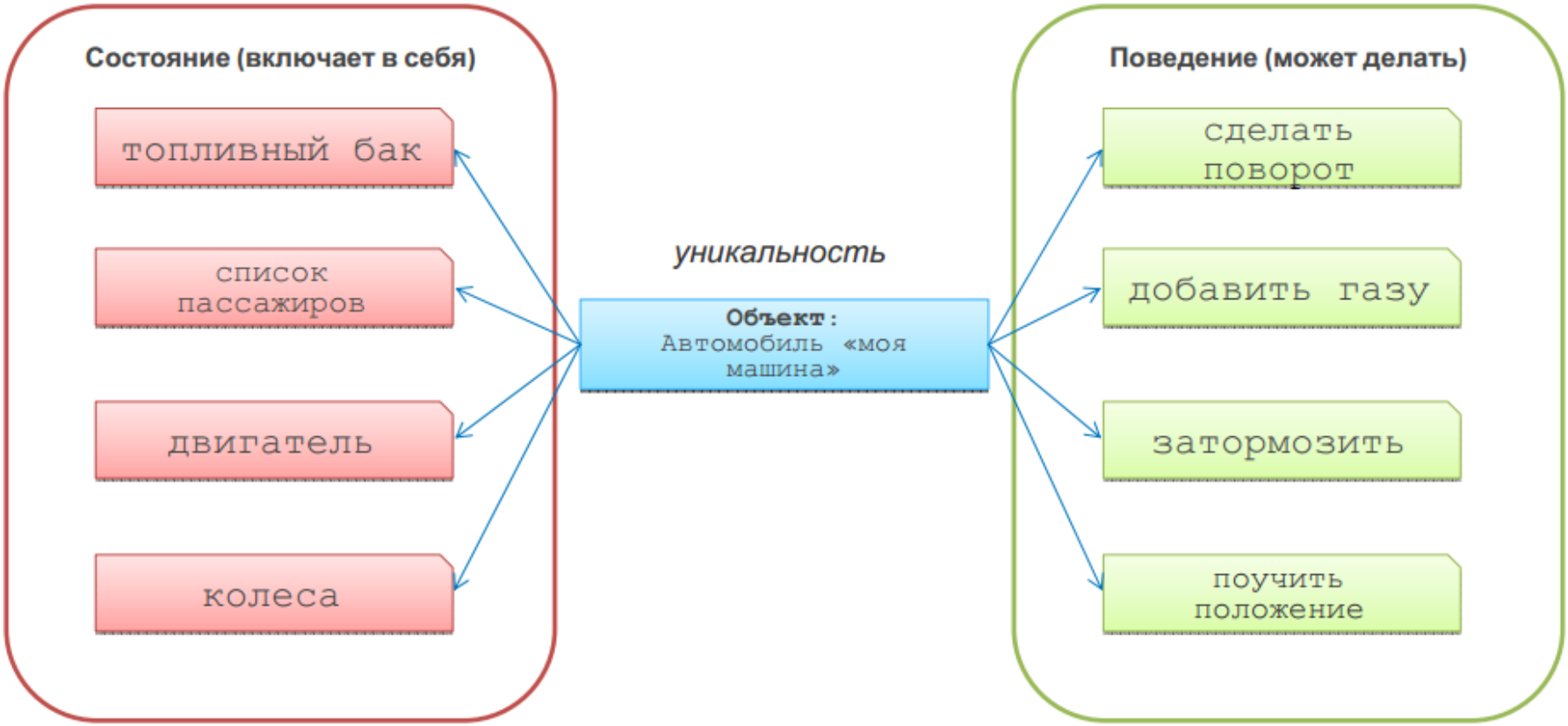
Механизмы ООП: ***Полиморфизм***

- **Полиморфизм** — возможность объектов с одинаковой спецификацией иметь различную реализацию. При этом различные объекты могут быть использованы одинаковым образом.
  - Позволяет писать более абстрактные программы.
  - Позволяет усилить повторное использование кода.
- Реализуется с помощью наследования и интерфейсов.

```
class Автостоянка {
    СписокАвтомобилей автомобили;
    процедура добавить(автомобиль) {
        автомобиль.закрыть();
        автомобиль.включитьСигнализацию();
        автомобили.добавить(автомобиль);
    }
}
Автостоянка стоянка = дом.гдеБлижайшаяАвтостоянка()
стоянка.добавить(мойАвтомобиль);
стоянка.добавить(мойГрузовик);
```

Объект: состояние, поведение, уникальность

Объект в программировании — некоторая сущность в компьютерном пространстве, обладающая определённым состоянием и поведением, имеющая заданные значения свойств (атрибутов) и операций над ними (методов).



Объекты и классы

Класс – это абстрактный тип данных. С помощью класса описывается некоторая сущность (ее характеристики и возможные действия). Например, класс может описывать студента, автомобиль и т.д. *Описав класс, мы можем создать его экземпляр – объект.*

Объект – это уже конкретный представитель класса.

Объекты и классы

Объект –
экземпляр класса.

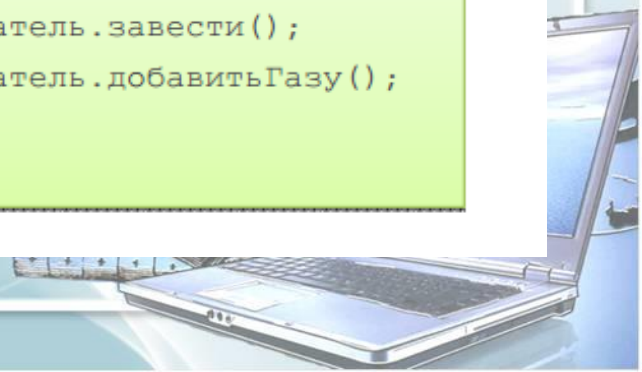
- состояние
- поведение
- уникальность

Класс –
тип объекта.

- определяет набор свойств и интерфейс взаимодействия.
- определяет поведение (реализацию)

```
Автомобиль мой = new Автомобиль();
Автомобиль жены = new Автомобиль();
Топливо топливо = жены.слитьТопливо();
мой.заправить(топливо);
мой.двигаться();
```

```
class Автомобиль {
    Двигатель двигатель;
    процедура двигаться() {
        двигатель.завести();
        двигатель.добавитьГазу();
    }
}
```



Объекты и классы


```
class Автомобиль {
    ТопливныйБак бак;
    СписокПассажиров пассажиры;
    Колеса колеса;
    Двигатель двигатель;

    процедура сделатьПоворот(угол);
    процедура добавитьГазу(уровень);
    процедура затормозить();
    Координаты получитьПоложение();
}
```

```
class Двигатель {
    Свечи свечи;
    Цилиндры цилиндры;
    Карбюратор карбюратор;

    процедура увеличитьОбороты();
    процедура переключитьРежим(режим);
}
```

```
Автомобиль мояМашина = я.купитьМашину();
Двигатель двигательМоейМашины = мояМашина.двигатель;
двигательМоейМашины.переключитьРежим(форсированный);
```