STRATEGY PATTERN

By SmartBoard Team

Agenda

- Scenario & Writing UML
- Discuss with each team's UML
- What if…
- BMVV design
- Strategy pattern
- Applied with BMVV design
- More exercise..

Scenario

- Act yourself as Car company owner

- Car component will consist of
  - Engine Type
  - Fuel used
  - Nitrous (Additional boost)

Scenario (2)

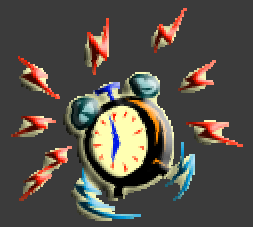
- From its components, we can calculate the fuel burn rate. So, it should include method getBurnFuel(double distance).
  - It comes from Engines power + Nitrous power (if used)
- Each car can used only
  - One type of engine
  - One type of Fuel
  - One type of Nitrous

- You need some program to show fuel burn rate for each car
- Car components can be changed in the future.
- Every team, please write UML Class diagram to according to the requirement above…

Writing UML ~10-15 min

Take your time...

Time’s up....



- Let’s see your UML diagram

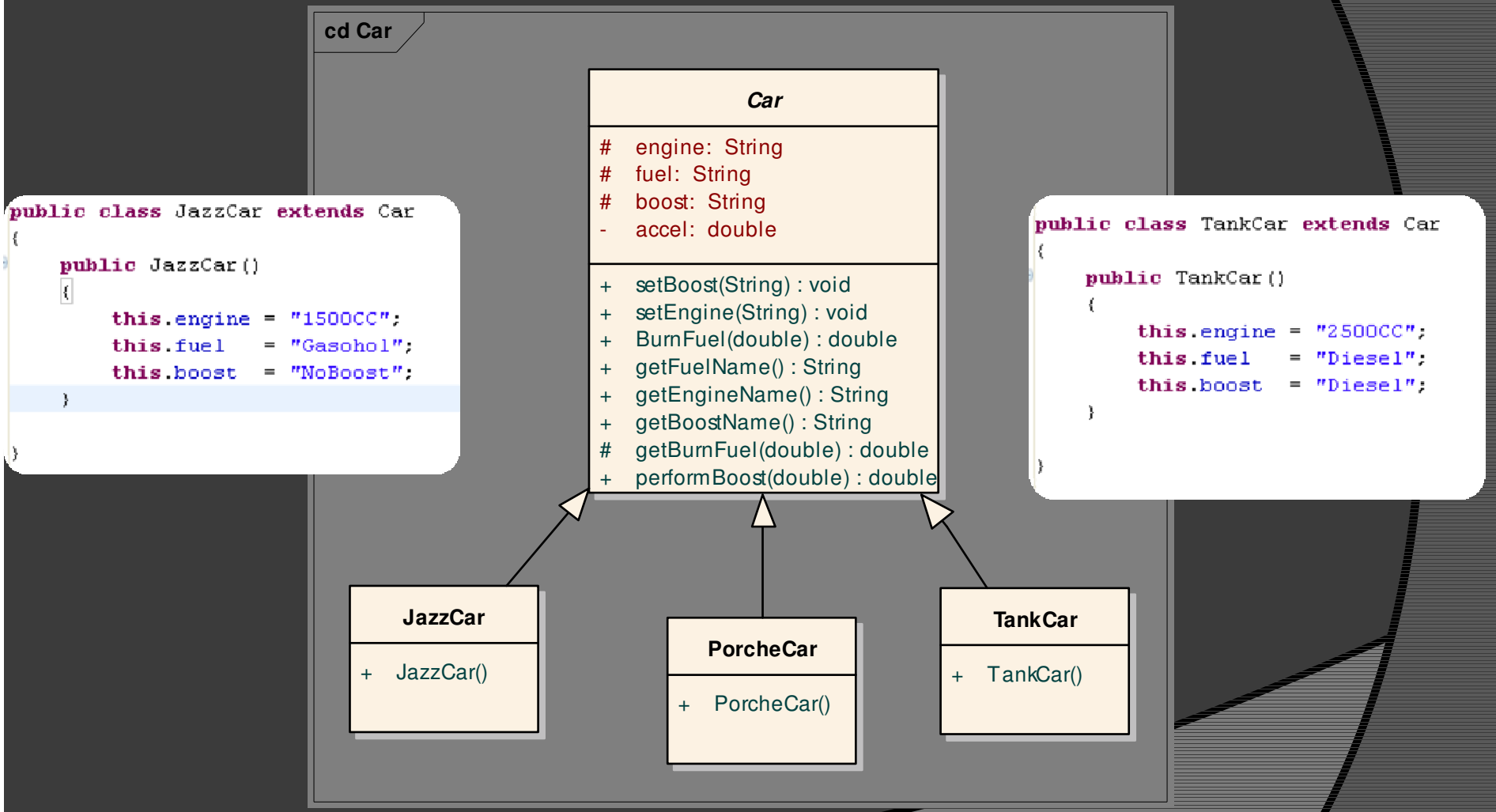
What if..

- More engine type is added
- More fuel type is added
- More nitrous type is added
- Some car can’t use some engine
- No more benzene ??

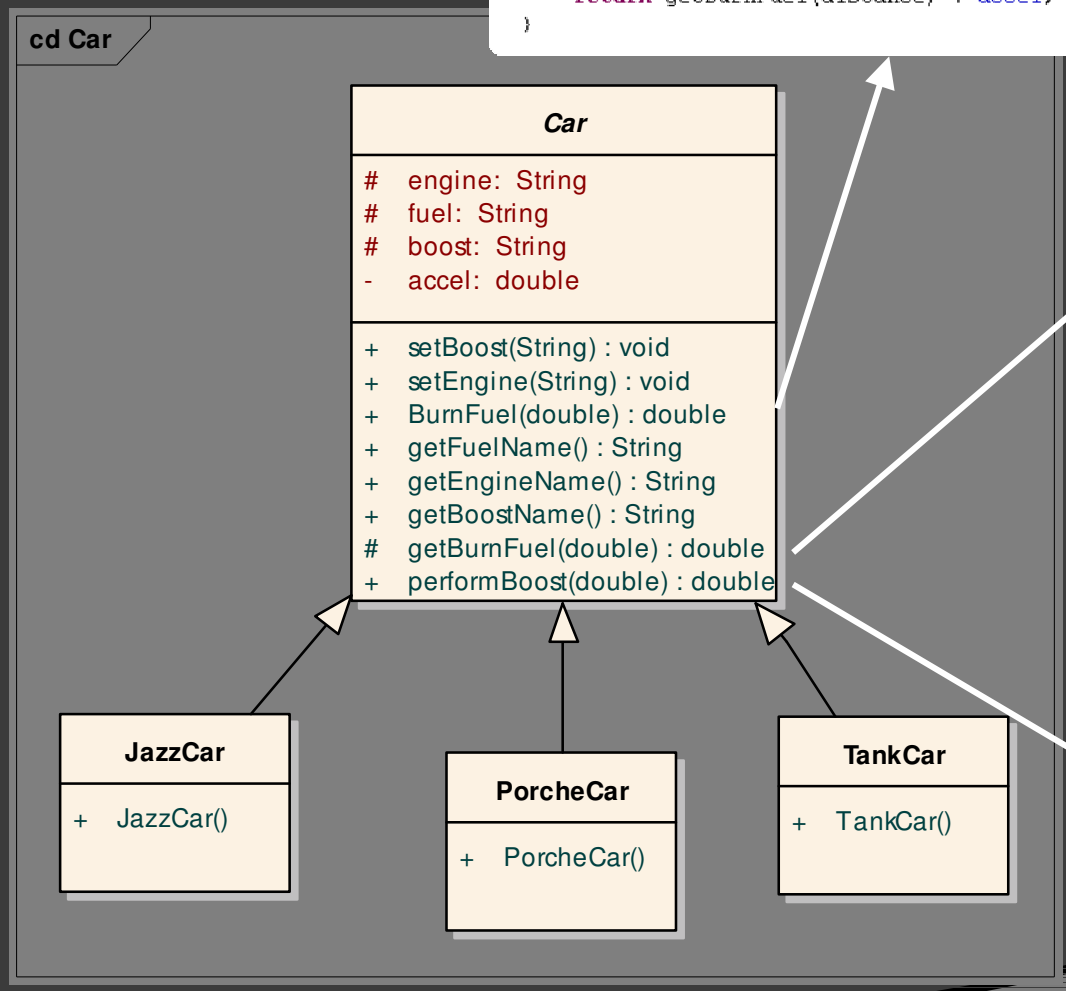
- Can you design support this changes?

Let’s record your problem

BMVV Car company design



BMVV Car company design #2




```
public double BurnFuel(double distance)
{
    return getBurnFuel(distance) + accel;
}
```

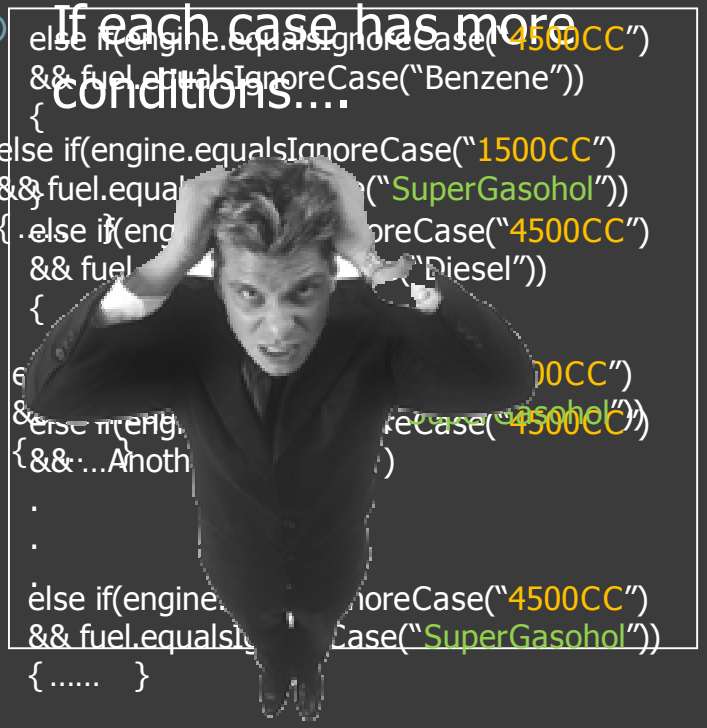
```
protected double getBurnFuel(double distance)
{
if(engine.equalsIgnoreCase("1500CC")
&& fuel.equalsIgnoreCase("Benzene"))
{
    double consume = distance*0.054;
    consume = consume*18/17;
    return consume;
}
else if(engine.equalsIgnoreCase("1500CC")
&& fuel.equalsIgnoreCase("Diesel"))
{  ..............  }
else if(engine.equalsIgnoreCase("1500CC")
 && fuel.equalsIgnoreCase("Gasohol"))
{  ..............  }
else if(engine.equalsIgnoreCase("1900CC")
   && fuel.equalsIgnoreCase("Benzene"))
{  .......  }
..............
.......
...
```

```
public double performBoost(double time)
{
    if(boost.equalsIgnoreCase("Benzene"))
    {
        accel = time*15/7;
    }

    else if(boost.equalsIgnoreCase("Diesel"))
    {
        accel = time*10/7;
    }
```

Problems occurs when...

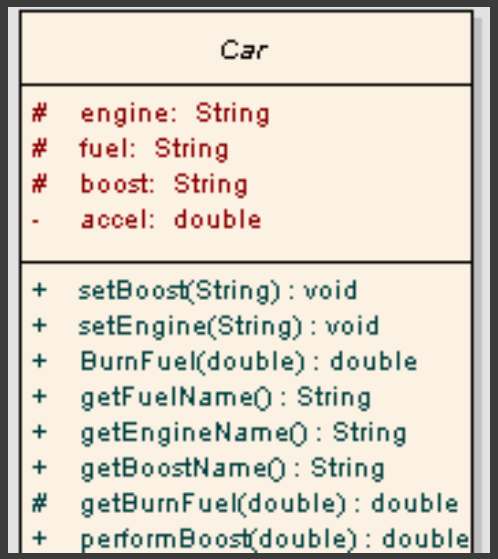
- If 4500cc engine are added
- No more benzene used.
- Super Gasohol is invented.
- If each case has more conditions....

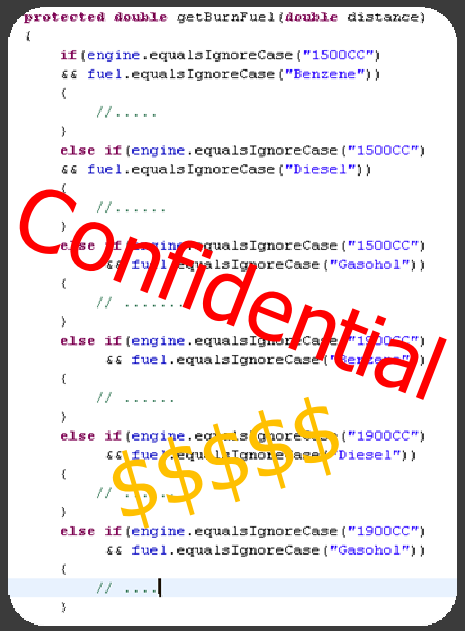
```
protected double getBurnFuel(double distance)
{
    if(engine.equalsIgnoreCase("1500CC")
    && fuel.equalsIgnoreCase("Benzene"))
    {
        //.....
    }
    else if(engine.equalsIgnoreCase("1500CC")
    && fuel.equalsIgnoreCase("Diesel"))
    {
        //......
    }
    else if(engine.equalsIgnoreCase("1500CC")
         && fuel.equalsIgnoreCase("Gasohol"))
    {
        // .......
    }
    else if(engine.equalsIgnoreCase("1900CC")
         && fuel.equalsIgnoreCase("Benzene"))
    {
        // ......
    }
    else if(engine.equalsIgnoreCase("1900CC")
         && fuel.equalsIgnoreCase("Diesel"))
    {
        // ......
    }
    else if(engine.equalsIgnoreCase("1900CC")
         && fuel.equalsIgnoreCase("Gasohol"))
    {
        // ....
    }
```



Unable to encapsulates data

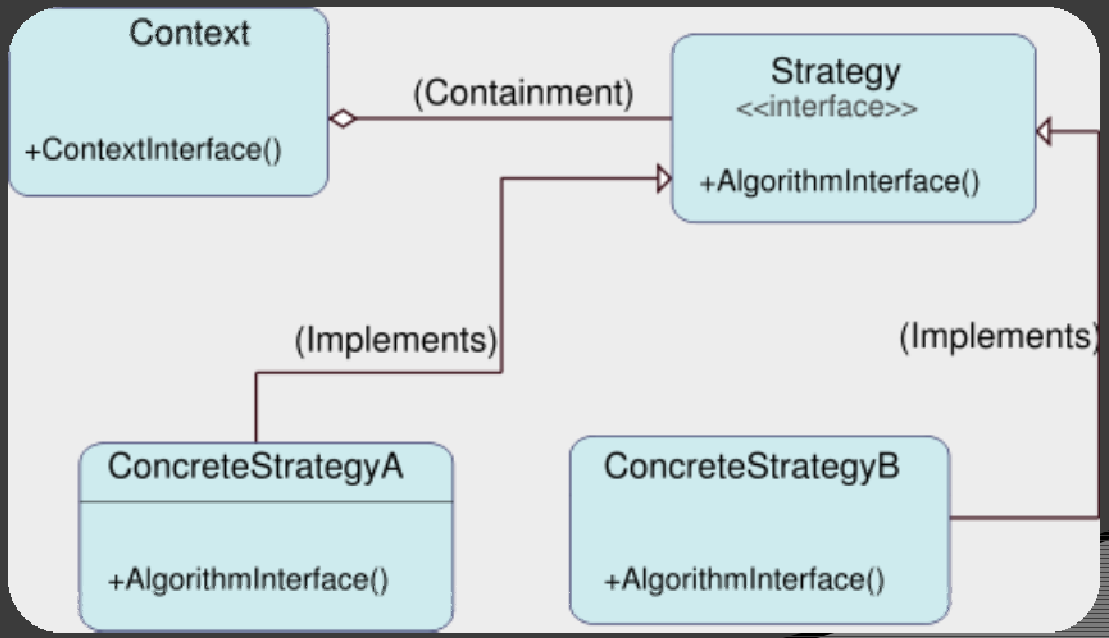
- According to this design, if you company need to outsource your work…





Strategy patterns

- Define a family of algorithms, encapsulate, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it.

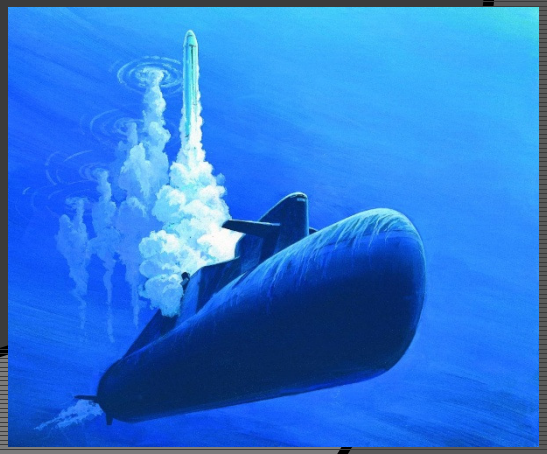


http://en.wikipedia.org/wiki/Strategy_pattern

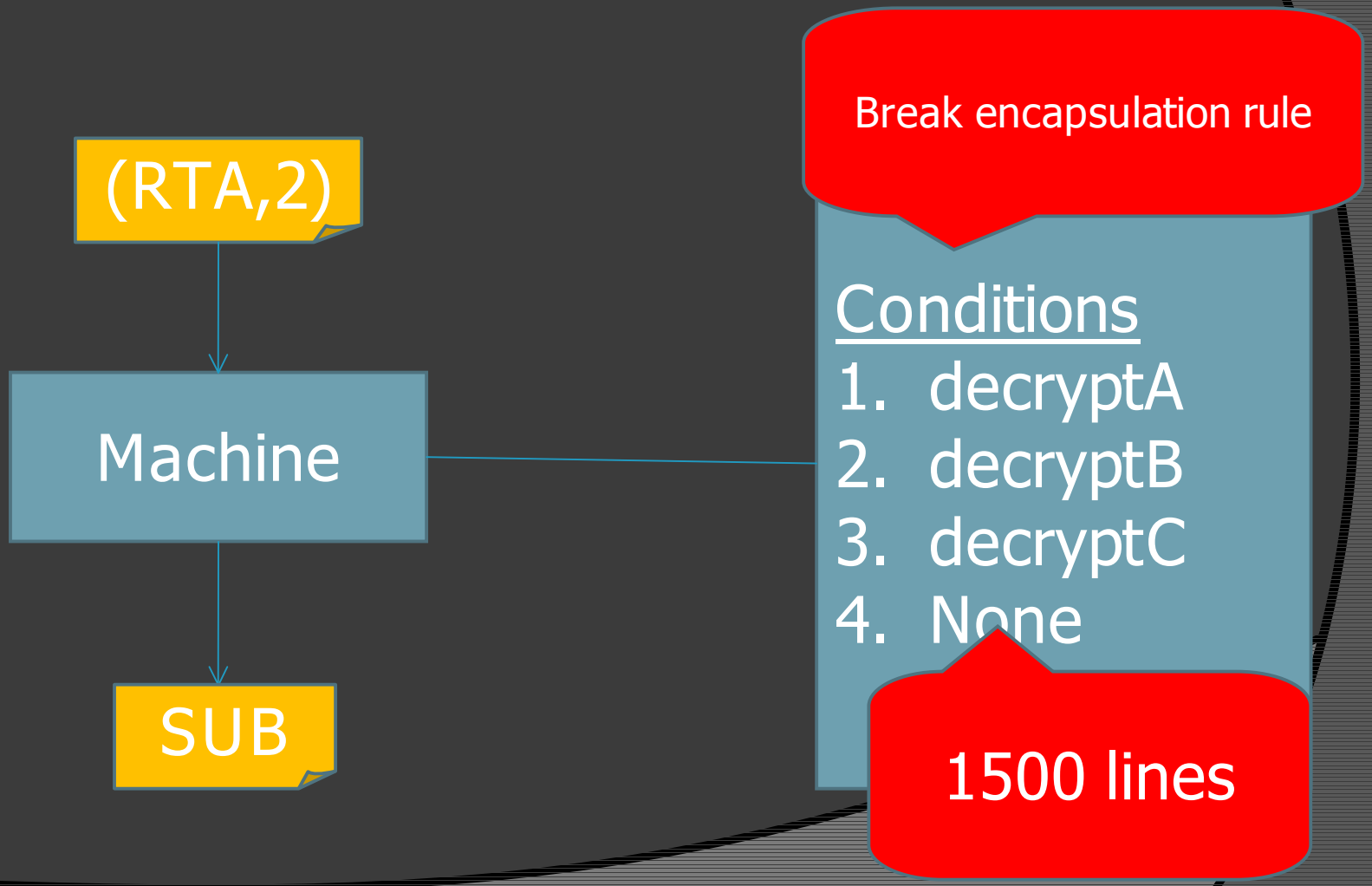
Thai military communication network

Every message will be encrypted.

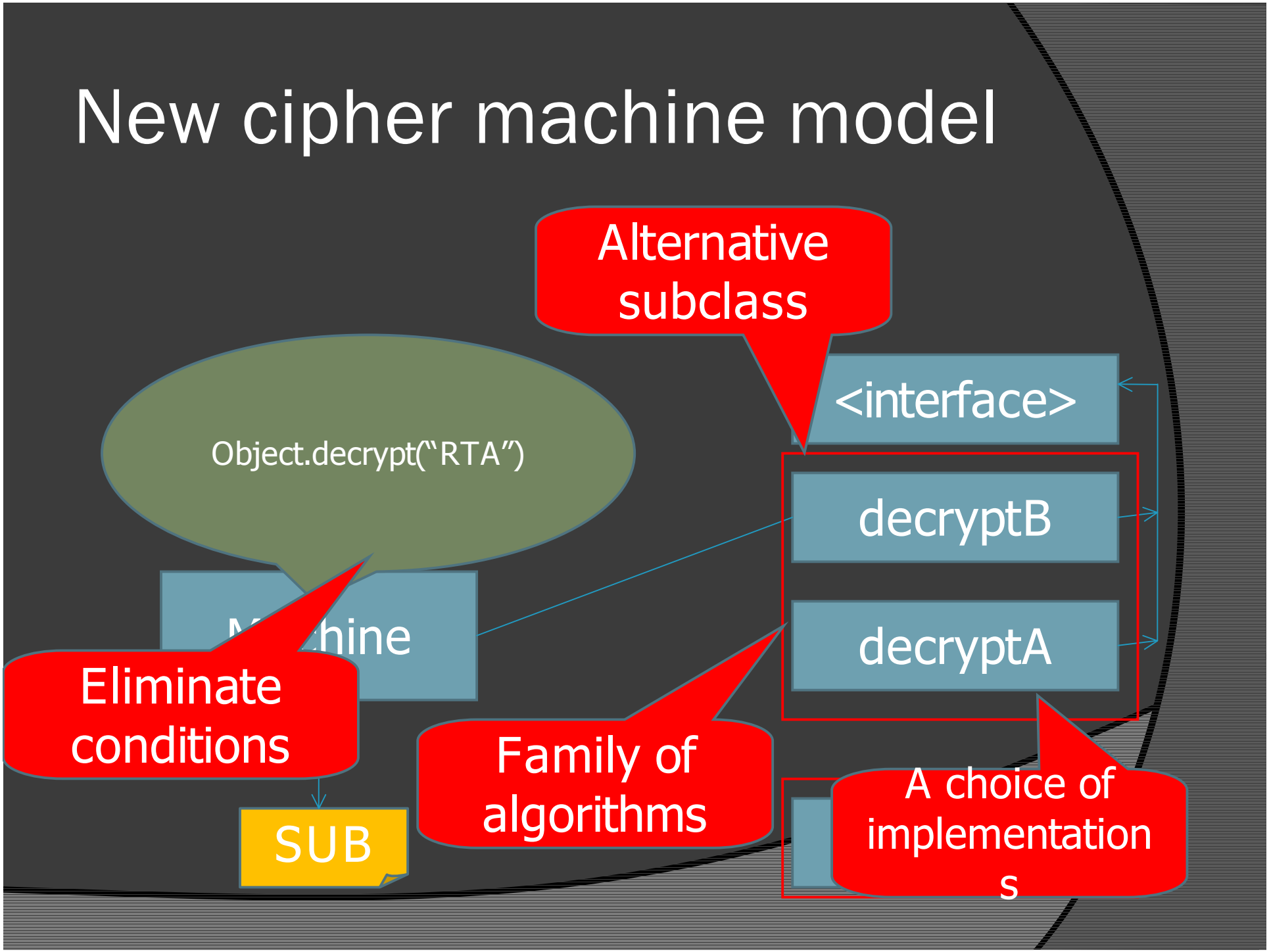
Communication between a station and an unite



Old cipher machine model



New cipher machine model



New cipher machine model

- Decrypt A and B can use signature only a string. Object.decrypt(String);
- But Decrypt C need more parameters.
  C
- So r ve two parame s. Object.decrypt(String, int);
- Ex . ObjectA.decrypt("RTA", Null);
  ObjectB.decrypt("RTA", Null);
  ObjectC.decrypt("RTA", 366);

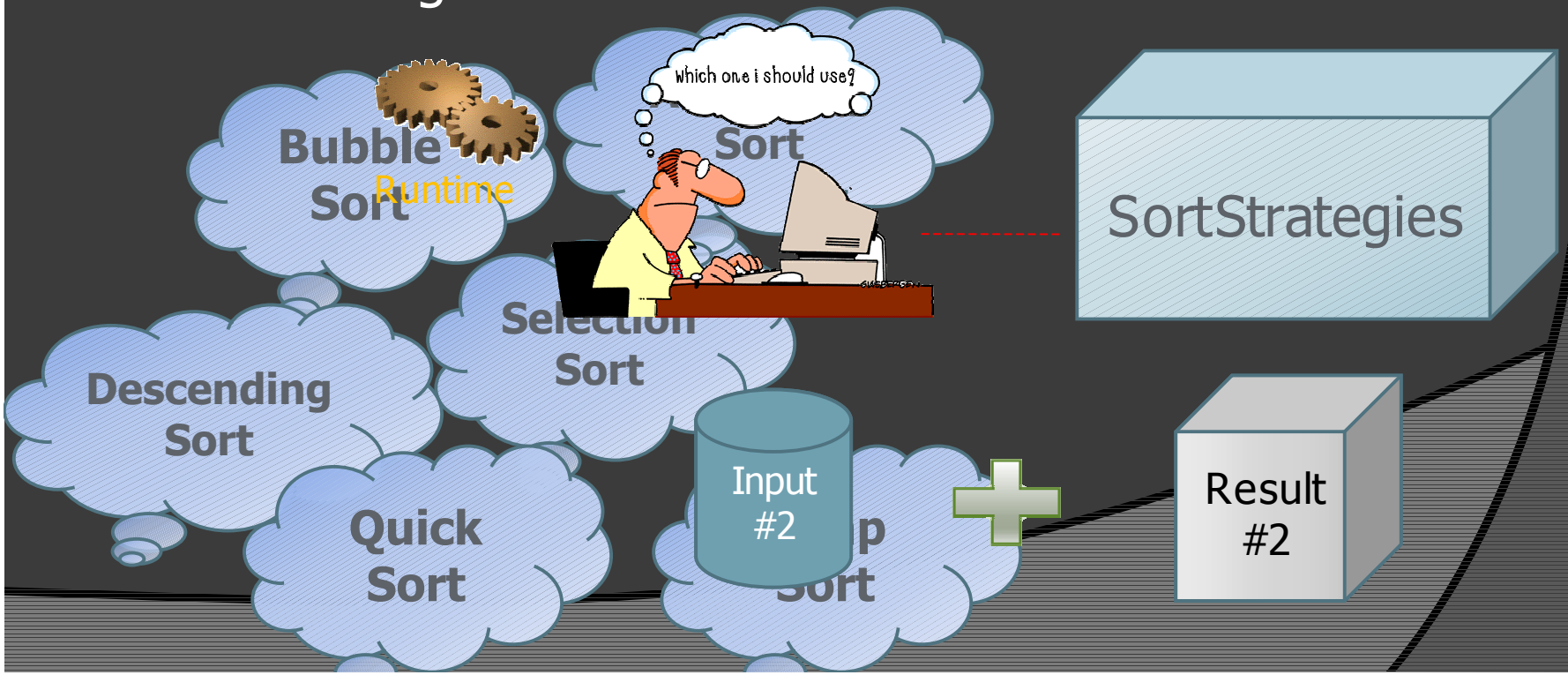
Communication overhead

Consequences

- Families of related algorithms.
- An alternative to subclassing.
- Strategies eliminate conditional statements.
- A choice of implementations.
- Clients must be aware of different strategies.
- Communication overhead between Strategy and Context.
- Increased number of objects.

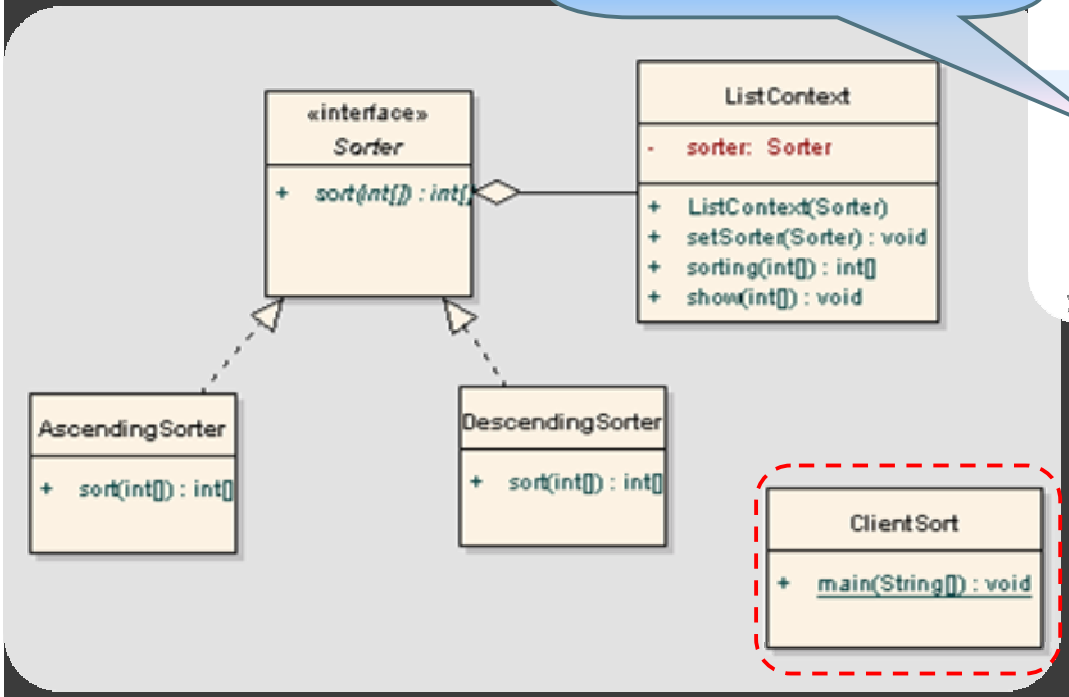
Another example : Sort

- Assume that, you need to write a sort program
- There’re many kinds of sort..
- Strategy is what we group the many algorithms that do the same things and make it interchangeable at run-time



Another example : Sort #2

Change the algorithm at runtime




```
public static void main(String[] args)
{
    ListContext listContext =
        new ListContext(new AscendingSorter());
    int[] listNumber = {9,1,5,2,8 };

     Sort for AscendingSorter
    listNumber = listContext.sorting(listNumber);
    System.out.println("Ascending Sorting");
    listContext.show(listNumber);

    // Sort for DescendingSorter
    listContext.setSorter(new DescendingSorter());
    listNumber = listContext.sorting(listNumber);
    System.out.println("Descending Sorting");
    listContext.show(listNumber);

}
```

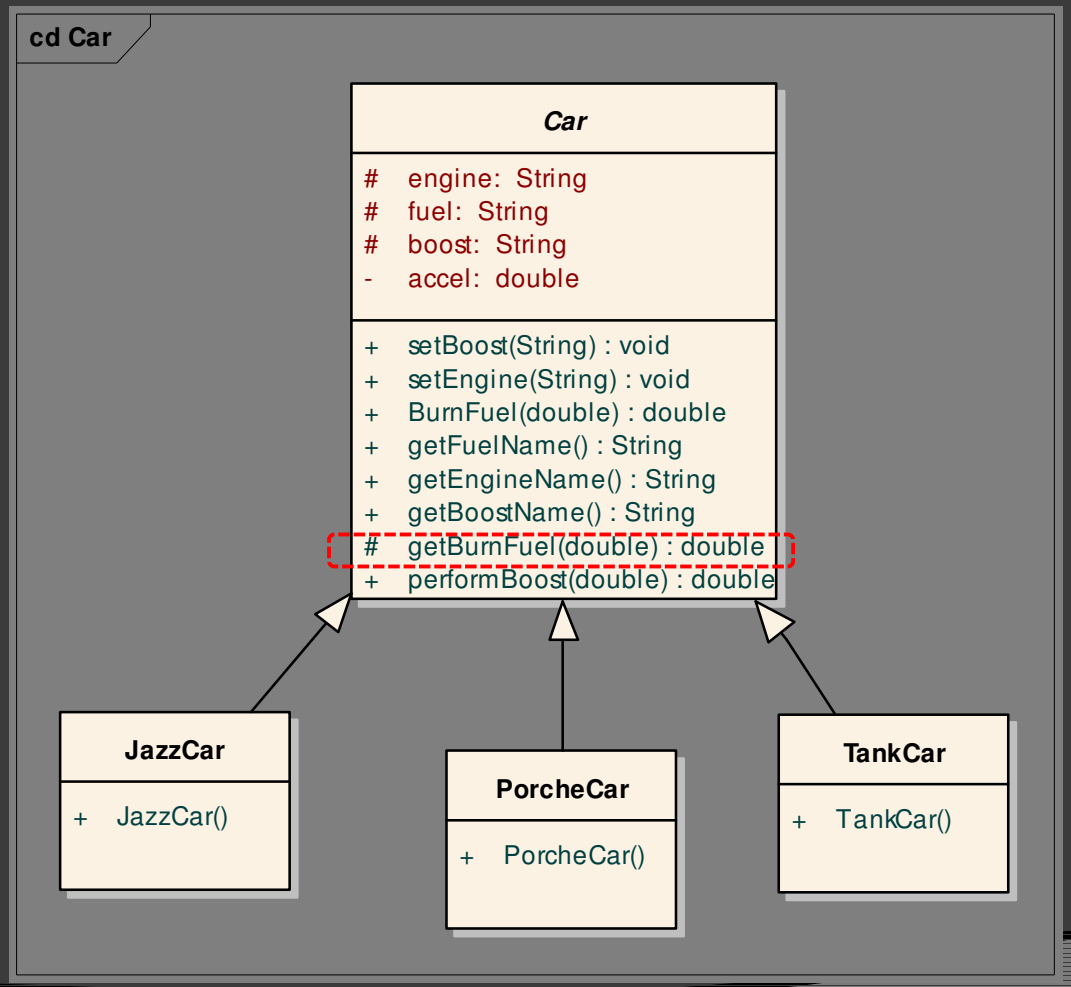
Problems | Javadoc | Declaration | Console

<terminated> ClientSort [Java Application] C:\Progra

```
Ascending Sorting
[1, 2, 5, 8, 9]
Descending Sorting
[9, 8, 5, 2, 1]
```

Apply strategy with BMVV design

- Indicates the problem..



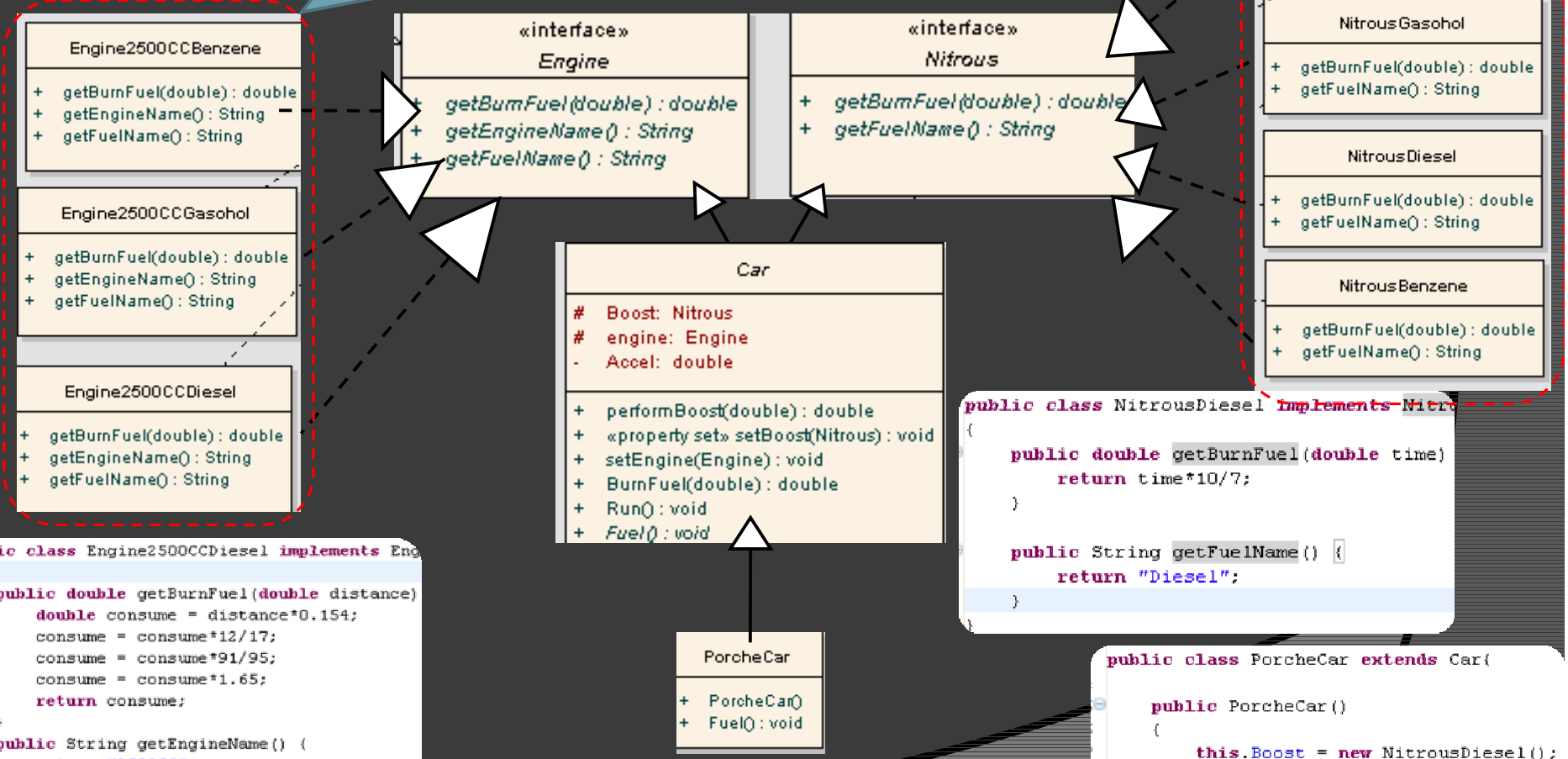
Move it to the new class...

Apply strategy with BMVV design

- Re-c

Need getBurnFuel() from different factor

So as Nitrous



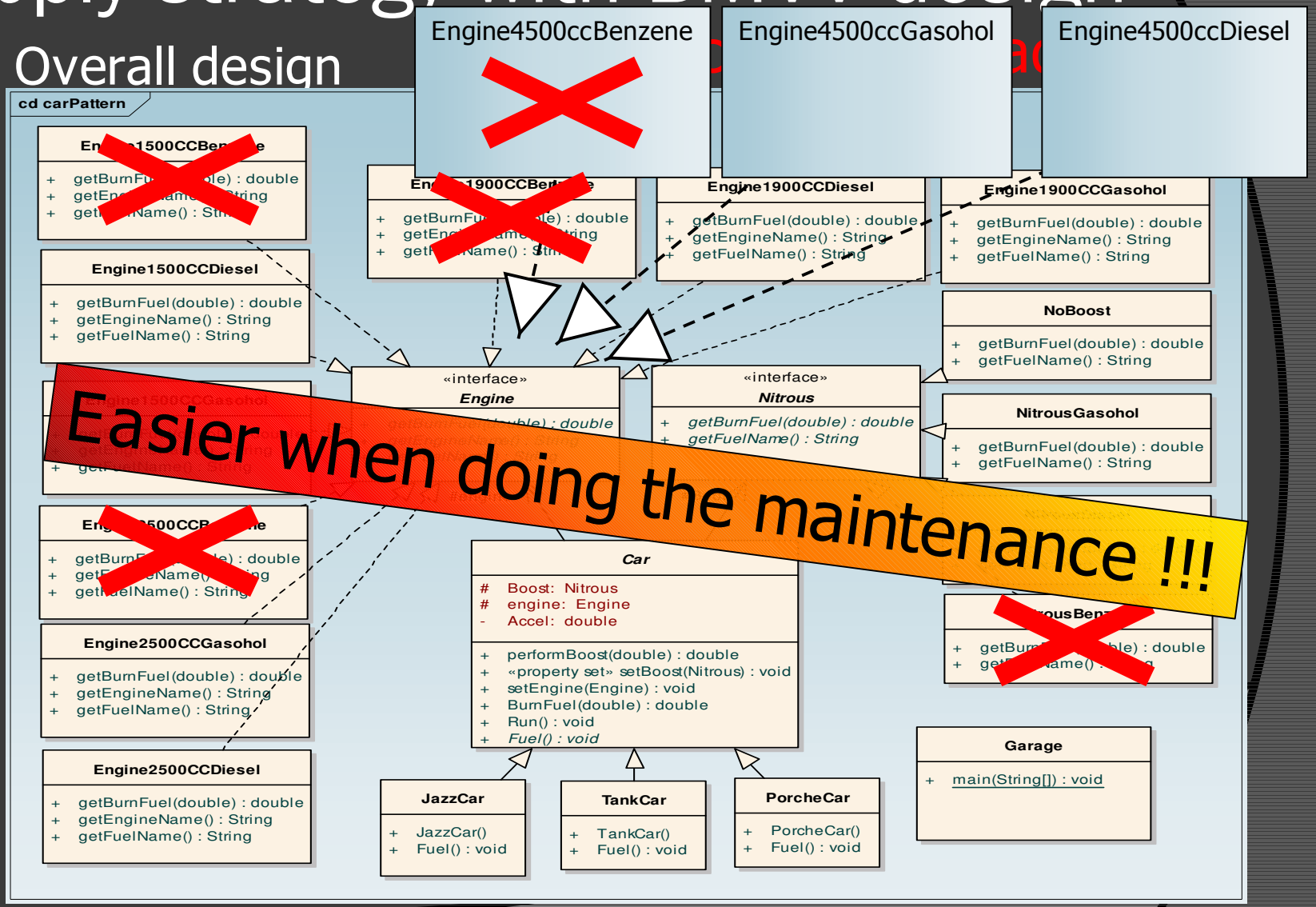

```
public class Engine2500CCDiesel implements Eng

    public double getBurnFuel(double distance)
        double consume = distance*0.154;
        consume = consume*12/17;
        consume = consume*91/95;
        consume = consume*1.65;
        return consume;
    }
    public String getEngineName() {
        return "2500CC";
    }
    public String getFuelName() {
        return "Diesel";
    }
```

```
public class NitrousDiesel implements Nitr
{
    public double getBurnFuel(double time)
        return time*10/7;
    }

    public String getFuelName() {
        return "Diesel";
    }
}
```

```
public class PorcheCar extends Car{

    public PorcheCar()
    {
        this.Boost = new NitrousDiesel();
        this.engine =
            new Engine2500CCDiesel();
    }
    @Override
    public void Fuel() {
```

Apply strategy with BMVV design

- Overall design



Apply strategy with BMVV design

- How it works at runtime

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Car newCar = new PorcheCar();
    Car Tank = new TankCar();
    Car Jazz = new JazzCar();

    System.out.println("Porche Burn Fuel without boost >> " + newCar.BurnFuel(100));
    System.out.println("Tank Burn Fuel without boost >> " + Tank.BurnFuel(100));
    System.out.println("Jazz Burn Fuel without boost >> " + Jazz.BurnFuel(100));

    Jazz.setEngine(new Engine2500CCBenzene());
    newCar.performBoost(10);
    Tank.performBoost(10);

    System.out.println("---  Modified & Add boost ---------");
    System.out.println("P  Burn Fuel With boost >> "+newCar.BurnFuel(100));
    System.out.println(  urn Fuel With boost >> "+Tank.BurnFuel(100));
    System.out.print  ed Jazz Burn Fuel >> "+Jazz.BurnFuel(100));
```

It will call BurnFuel(double) according to its engine

Jazz is changing its engine

```
public JazzCar()
{
    engine = new Engine1500CCGasohol();
    Boost = new NoBoost();
}
```

```
public PorcheCar()
{
    this.Boost = new NitrousBenzene();
    this.engine = new Engine2500CCBenzene();
```

```
public TankCar()
{
    engine = new Engine2500CCDiesel();
    Boost = new NitrousDiesel();
}
```

Problems | Javadoc | Declaration | Console

```
<terminated> Garage [Java Application] C:\Program Files\Java\jre1.5.0_11\bin\javaw.ex
Porche Burn Fuel without boost >> 15.098674922600619
Tank Burn Fuel without boost >> 17.181250773993806
Jazz Burn Fuel without boost >> 4.564086687306502
-------- Modified & Add boost ---------
Porche Burn Fuel With boost >> 29.384389208314907
Tank Burn Fuel With boost >> 31.466965059708095
Modified Jazz Burn Fuel >> 15.098674922600619
```

An action adventure game

Start

Solutions of exercise

Q/A

References

- www.viewimages.com
- www.thailandstrategy.com
- http://www.maxituning.pl/archiwum/0412/pics/technika/nos2.jpg
- www.mysticcobra.net/system/
- http://jpfamily.net/Coolpix/NHRA/slides/
- http://www.robinstewart.com/products/graphics/gs_icon.gif
- www.freewarebox.com/screenshot_2476_clock.html
- http://tv.pcworld.hu/apix/0612/Data%20thief.png
- http://www.offthemarkcartoons.com/cartoons/