TECH CRASH COURSE
(http://www.techcrashcourse.com/)

C GRAPHICS

- C - Graphics Tutorial (http://www.techcrashcourse.com/2015/08/c-graphics-programming-tutorial.html)
- Draw a Circle (http://www.techcrashcourse.com/2015/08/c-program-draw-circle-graphics.html)
- Draw Bar Graph (http://www.techcrashcourse.com/2015/08/c-program-draw-bar-graph-using-graphics.html)
- Draw 3D Bar Graph (http://www.techcrashcourse.com/2015/08/c-program-draw-3d-bar-graph-using-graphics.html)
- Draw Sine Wave (http://www.techcrashcourse.com/2015/08/c-program-draw-sine-graph-wave-graphics.html)
- Draw Pie Chart (http://www.techcrashcourse.com/2015/08/c-program-draw-pie-chart-using-graphics.html)
- Make a Digital Clock (http://www.techcrashcourse.com/2015/08/c-program-make-digital-clock-using-graphics.html)
- Bouncing Ball Animation (http://www.techcrashcourse.com/2015/08/c-program-bouncing-ball-animation-graphics.html)
- Moving Car Animation (http://www.techcrashcourse.com/2015/08/c-graphics-program-moving-car-animation.html)

Filter Design Pattern

The **Filter pattern** allows us to filter a collection of objects using various conditions(filters) and chaining them in a decoupled way through logical operations. This design pattern is useful when we want to select subset of objects that satisfies one or multiple conditions. It comes under *structural pattern*.

Advantages of Filter Pattern

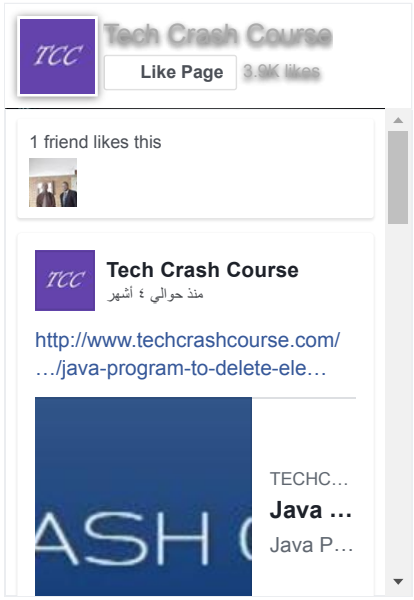
- *It provides a way to filter objects based on certain criteria.*
- *We can add a new filter any time without affecting client's code.*
- *We can select filters dynamically during program execution.*

For Example: Let, EmployeeList be the list of all employees of a company and we want to select all Male employees whose salary is between 1000 to 5000. In this case we need three filters.
Male Filter : Returns List of all Male employees.
LessThanFilter : Returns List of employees whose salary is less than 5000.
GreaterThanFilter : Returns List of employees whose salary is greater than 1000.

Now, we have to perform logical And of above mentioned three filters. Let's see how filter pattern can help us in solving this problem.

Implementation of Filter Design Pattern

Employee.java



```
public class Employee {
    private String name;
    private String gender;
    private String department;
    private int salary;

    public Employee(String name, String gender, String department, int salary){
        this.name = name;
        this.gender = gender;
        this.department = department;
        this.salary = salary;
    }

    public String getName() {
     return name;
    }

    public String getGender() {
        return gender;
    }

    public String getDepartment() {
        return department;
    }

    public int getSalary(){
     return salary;
    }
}
```

We will define Filter interface having checkCondition method which takes a List of employees as input and then returns a subset of employees who satisfy the filter criteria. Every filter criteria class must implement this interface.

Filter.java

```
import java.util.List;

public interface Filter {
    public List<Employee> checkCondition(List<Employee> employees);
}
```

GenderFilter.java

Popular Posts

(http://www.techcrashcourse.com/2016/02/print-triangle-pattern-of-prime-numbers.html)

C program to print triangle pattern of prime numbers (http://www.techcrashcourse.com/2016/02/print-triangle-pattern-of-prime-numbers.html)

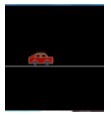
C Program to Print Lower Triangular Matrix (http://www.techcrashcourse.com/2015/03/c-program-lower-triangular-matrix.html)

```
import java.util.List;
import java.util.ArrayList;

public class GenderFilter implements Filter {
    private String gender;

    public GenderFilter(String gender){
        this.gender = gender;
    }

    @Override
    public List<Employee> checkCondition(List<Employee> employees){
        List<Employee> filterredEmployees = new ArrayList<Employee>();
        for(Employee e : employees){
            if(e.getGender().equalsIgnoreCase(gender)){
                filterredEmployees.add(e);
            }
        }
        return filterredEmployees;
    }
}
```

DepartmentFilter.java

```
import java.util.List;
import java.util.ArrayList;

public class DepartmentFilter implements Filter {
    private String department;

    public DepartmentFilter(String department){
        this.department = department;
    }

    @Override
    public List<Employee> checkCondition(List<Employee> employees){
        List<Employee> filterredEmployees = new ArrayList<Employee>();
        for(Employee e : employees){
            if(e.getDepartment().equalsIgnoreCase(department)){
                filterredEmployees.add(e);
            }
        }
        return filterredEmployees;
    }
}
```

SalaryGreaterThanFilter.java



(http://www.techcrashcourse.com/2015/08/c-graphics-program-moving-car-animation.html)

C Program for Moving Car Animation Using C Graphics (http://www.techcrashcourse.com/2015/08/c-graphics-program-moving-car-animation.html)

(http://www.techcrashcourse.com/2015/08/c-program-bouncing-ball-animation-graphics.html)

C Program for Bouncing Ball Animation Using C Graphics (http://www.techcrashcourse.com/2015/08/c-program-bouncing-ball-animation-graphics.html)

C Program to Print Even Numbers Between 1 to 100 using For and While Loop (http://www.techcrashcourse.com/2015/11/c-program-to-print-all-even-numbers-between-1-to-100-using-loop.html)

(http://www.techcrashcourse.com/2016/02/c-program-to-print-triangle-pyramid-star-pattern.html)

C program to print triangle, pyramid, geometrical shapes and star patterns (http://www.techcrashcourse.com/2016/02/c-program-to-print-triangle-pyramid-star-pattern.html)

C++ Program to Calculate Grade of Student Using Switch Case (http://www.techcrashcourse.com/2017/01/cpp-program-calculate-grade-of-student.html)

C Graphics Programming Tutorial (http://www.techcrashcourse.com/2015/08/c-graphics-programming-tutorial.html)

```
import java.util.List;
import java.util.ArrayList;

public class SalaryGreaterThanFilter implements Filter {
    private int value ;

    public SalaryGreaterThanFilter(int value){
        this.value = value;
    }

    @Override
    public List<Employee> checkCondition(List<Employee> employees){
        List<Employee> filterredEmployees = new ArrayList<Employee>();
        for(Employee e : employees){
            if(e.getSalary() > value){
                filterredEmployees.add(e);
            }
        }
        return filterredEmployees;
    }
}
```

SalaryLessThanFilter.java

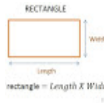
```
import java.util.List;
import java.util.ArrayList;

public class SalaryLessThanFilter implements Filter {
    private int value ;

    public SalaryLessThanFilter(int value){
        this.value = value;
    }

    @Override
    public List<Employee> checkCondition(List<Employee> employees){
        List<Employee> filterredEmployees = new ArrayList<Employee>();
        for(Employee e : employees){
            if(e.getSalary() < value){
                filterredEmployees.add(e);
            }
        }
        return filterredEmployees;
    }
}
```

AndFilter.java takes a List of Filters and then perform the logical and of all filters. The output of one filter becomes the input of other in chained manner. It is used for implementing composite criteria like, "List of employees who are Male and works in CSE department".

AndFilter.java

(http://www.techcrashcourse.com/2015/08/c-graphics-programming-tutorial.html)

C Program to Print Odd Numbers Between 1 to 100 using For and While Loop (http://www.techcrashcourse.com/2015/11/c-program-to-print-all-odd-numbers-between-1-to-100-using-loop.html)



(http://www.techcrashcourse.com/2015/03/c-program-to-calculate-area-of-rectangle.html)

C Program to Calculate Area and Perimeter of a Rectangle (http://www.techcrashcourse.com/2015/03/c-program-to-calculate-area-of-rectangle.html)

```
import java.util.List;

public class AndFilter implements Filter {
    private List<Filter> filterList;

    public AndFilter(List<Filter> filterList){
        this.filterList = filterList;
    }

    @Override
    public List<Employee> checkCondition(List<Employee> employees){
        List<Employee> filterredEmployees = employees;
        for(Filter filter : filterList){
            filterredEmployees = filter.checkCondition(filterredEmployees);
        }
        return filterredEmployees;
    }
}
```

FilterPatternExample.java first creates a list of employees and the filter them using specific filter classes to gets subset of employees satisfying particular condition(s).

```
import java.util.List;
import java.util.ArrayList;

public class FilterPatternExample {
    public static void main(String[] args) {
        List<Filter> filterList = new ArrayList<Filter>();
        List<Employee> employeeList= new ArrayList<Employee>();
        employeeList.add(new Employee("Jack", "Male", "CSE", 1000));
        employeeList.add(new Employee("Rose", "Female", "ECE", 2000));
        employeeList.add(new Employee("George", "Male", "CSE", 3000));
        employeeList.add(new Employee("Mike", "Male", "ECE", 4000));
        employeeList.add(new Employee("Julia", "Female", "CSE", 5000));

        // Define various filters
        Filter departmentFilter = new DepartmentFilter("CSE");
        Filter maleFilter = new GenderFilter("Male");
        Filter lessThanFlter = new SalaryLessThanFilter(5000);
        Filter greaterThanFilter = new SalaryGreaterThanFilter(1000);

        System.out.println("List of Male Employees");
            printEmployeesName(maleFilter.checkCondition(employeeList));

        System.out.println("List of CSE Employees");
            printEmployeesName(departmentFilter.checkCondition(employeeList));

        System.out.println("List of Male Employees whose salary is" +
            " greater than 1000 and less than 5000");
        filterList.add(lessThanFlter);
        filterList.add(greaterThanFilter);
        filterList.add(maleFilter);
        Filter andFilter = new AndFilter(filterList);
        printEmployeesName(andFilter.checkCondition(employeeList));
    }

    private static void printEmployeesName(List<Employee> empList){
        for(Employee e : empList){
            System.out.print(e.getName() + ", ");
        }
        System.out.println("\n");
    }
}
```

Output

```
List of Male Employees
Jack, George, Mike,

List of CSE Employees
Jack, George, Julia,

List of Male Employees whose salary is greater than 1000 and less than 5000
George, Mike,
```

Related Topics

Strategy Design Pattern (http://www.techcrashcourse.com/2015/10/strategy-design-pattern-in-java.html)
Builder Design Pattern (http://www.techcrashcourse.com/2015/10/builder-design-pattern-in-java.html)
Abstract Factory Design Pattern (http://www.techcrashcourse.com/2015/10/abstract-factory-design-pattern.html)
Flyweight Design Pattern (http://www.techcrashcourse.com/2015/10/flyweight-design-pattern-in-java.html)
Facade Design Pattern (http://www.techcrashcourse.com/2015/10/facade-design-pattern-in-java.html)
Command Design Pattern (http://www.techcrashcourse.com/2015/10/command-design-pattern-in-java.html)
Memento Design Pattern (http://www.techcrashcourse.com/2015/10/memento-design-pattern-in-java.html)
Mediator Design Pattern (http://www.techcrashcourse.com/2015/10/mediator-design-pattern-in-java.html)
Observer Design Pattern (http://www.techcrashcourse.com/2015/10/observer-design-pattern-in-java.html)
Singleton Design Pattern (http://www.techcrashcourse.com/2015/10/singleton-design-pattern.html)
List of Design Patterns (http://www.techcrashcourse.com/2015/10/design-patterns-tutorial-in-java.html)

← Previous (http://www.techcrashcourse.com/2015/10/bridge-design-pattern.html)

Next → (http://www.techcrashcourse.com/2015/10/composite-design-pattern-in-java.html)

Posted by Tech Crash Course (https://www.blogger.com/profile/17353939485345171416)

Newer Post (http://www.techcrashcourse.com/2015/10/composite-design-pattern-in-java.html) Home (http://www.techcrashcourse.com/) Older Post (http://www.techcrashcourse.com/2015/10/builder-design-pattern-in-java.html)