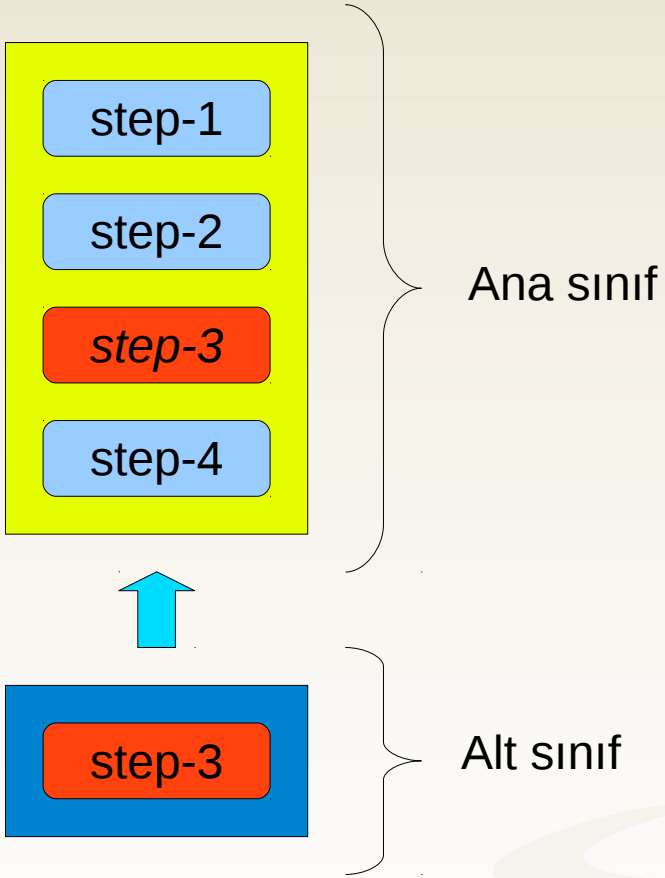
JAVA Eğitimleri

Template Method Örüntüsü

Template Method

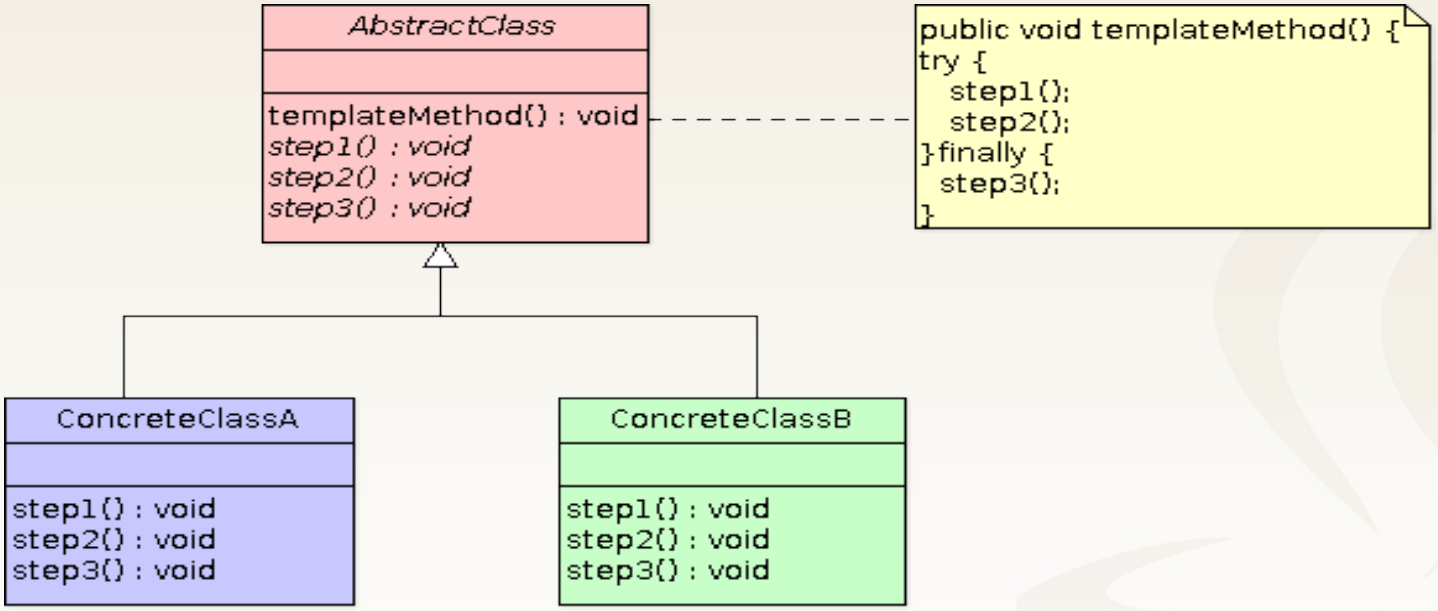


Algoritmanın **ana akışı encapsule** edilerek **spesifik adımların işleyişi**nin alt sınıflar tarafından belirlenmesine imkan tanır

Template Method

- Bir algoritma **genel hatları** ile birden fazla sınıf için aynı olabilir
- Genelde her bir sınıf bu algoritmadaki **adımlardan bazıları**nın farklı biçimde işletilmesini istemektedir
- Algoritmanın ana akışı değişmez biçimde tanımlanarak **sabitlenir** ve **adımları belli bir sıra ile işlettiğ**inden emin olunur
- Değişen adımlar ise **alt sınıflar tarafından implement** edilir

Template Method Sınıf Diagramı



Java Servlets ve Template Method

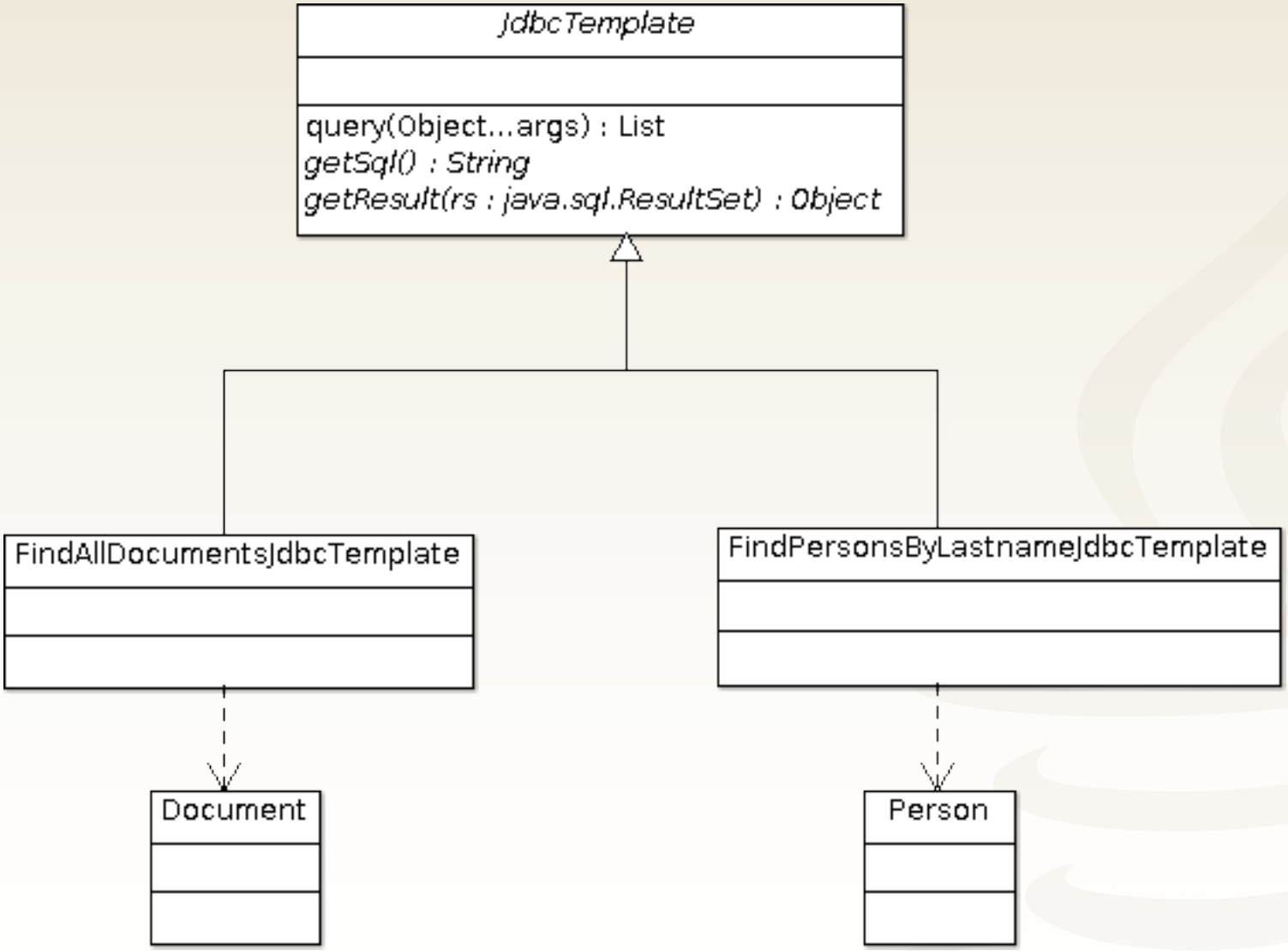
```
public abstract class HttpServlet extends GenericServlet {
protected void service(HttpServletRequest req,
HttpServletResponse resp)
        throws ServletException, IOException
    {
        String method = req.getMethod();

        if (method.equals(METHOD_GET)) {
            long lastModified = getLastModified(req);
            if (lastModified == -1) {
                doGet(req, resp);
            } else {
                long ifModifiedSince =
req.getDateHeader(HEADER_IFMODSINCE);
                if (ifModifiedSince < lastModified) {
                    maybeSetLastModified(resp,
lastModified);
                    doGet(req, resp);
                } else {

resp.setStatus(HttpServletResponse.SC_NOT_MODIFIED);
                }
            }

        } else if (method.equals(METHOD_HEAD)) {
            long lastModified = getLastModified(req);
            maybeSetLastModified(resp, lastModified);
            doHead(req, resp);

        } else if (method.equals(METHOD_POST)) {
            doPost(req, resp);

        }
```

```
          else if (method.equals(METHOD_PUT)) {
            doPut(req, resp);

        } else if (method.equals(METHOD_DELETE)) {
            doDelete(req, resp);

        } else if (method.equals(METHOD_OPTIONS)) {
            doOptions(req,resp);

        } else if (method.equals(METHOD_TRACE)) {
            doTrace(req,resp);

        } else {
            String errMsg =
lStrings.getString("http.method_not_implemented");
            Object[] errArgs = new Object[1];
            errArgs[0] = method;
            errMsg = MessageFormat.format(errMsg,
errArgs);


resp.sendError(HttpServletResponse.SC_NOT_IMPLEME
NTED, errMsg);
        }
    }
}
```

LAB ÇALIŞMASI: Template Method

- Java'da veritabanı işlemleri için JDBC API kullanılır. Veritabanında işletilecek SQL ifadeleri için her seferinde yapılması gereken rutin işlemler vardır. Bunlar;
  veritabanı bağlantısı kurulması
  statement nesnesinin oluşturulması
  SQL ifadesinin yazılması
  SQL ifadesinin çalıştırılması
  dönen ResultSet üzerinde iterate edilerek işlem yapılması
  işi biten statement ve veritabanı bağlantı nesnelerinin kapatılması
- Bu işlemlerden sadece **SQL ifadesinin yazılması** ve **dönen ResultSet üzerinde iterate edilerek işlem yapılması** probleme göre farklılık gösterir. Diğer adımlar her seferinde aynıdır
- Ana iskelet kodu sabitleyen ve her seferinde aynı sıra ile işleten, değişen iki adımı ise alt sınıfların tanımlamasına izin veren bir JdbcTemplate sınıfı yazılması istenmektedir

LAB ÇALIŞMASI: Template Method



Template Method Örüntüsünün Sonuçları

- Algoritmanın **ana işleyişi sabitlenmiş** ve **kontrol altına alınmış** olur
- Böylece alt sınıfların ata sınıfın davranışını **uygun olmayan biçimde override** etmelerinin önüne geçilmiş olunur

İletişim

www.harezmi.com.tr

www.java-egitimleri.com

info@harezmi.com.tr

info@java-egitimleri.com

@HarezmiBilisim

@JavaEgitimleri