

Projektni obrasci

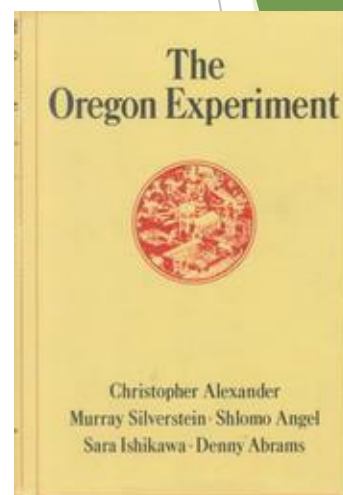
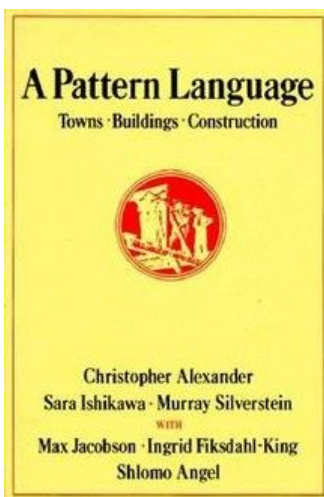
Strategy - Singleton - Adapter

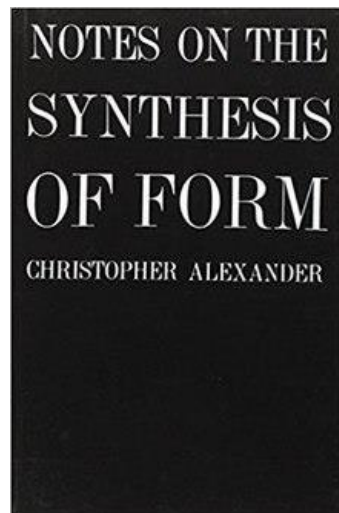
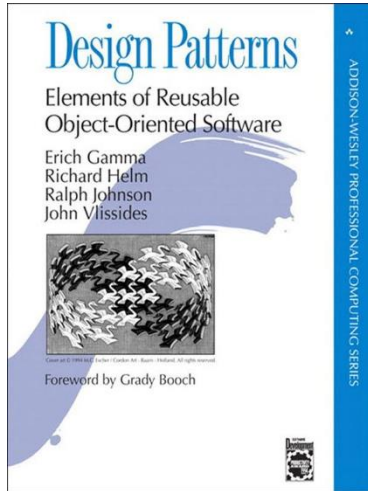
Projektni obrasci

- ▶ Ljudi teže primeni jednostavnih obrazaca koji ponovo mogu da se upotrebe
- ▶ Korišćenje obrazaca ima za cilj dobijanje jednostavnog i *svima* razumljivog rešenja
- ▶ Oduvek postoje i primenjivi su na svaki aspekt ljudskog života

Moderni projektni obrasci

- ▶ Kristofer Aleksander (1977)
 - ▶ Arhitekta
 - ▶ Formalizovao obrasce za upotrebu u opštoj arhitekturi i urbanizmu
 - ▶ Dao osnovu za primenu u drugim naukama





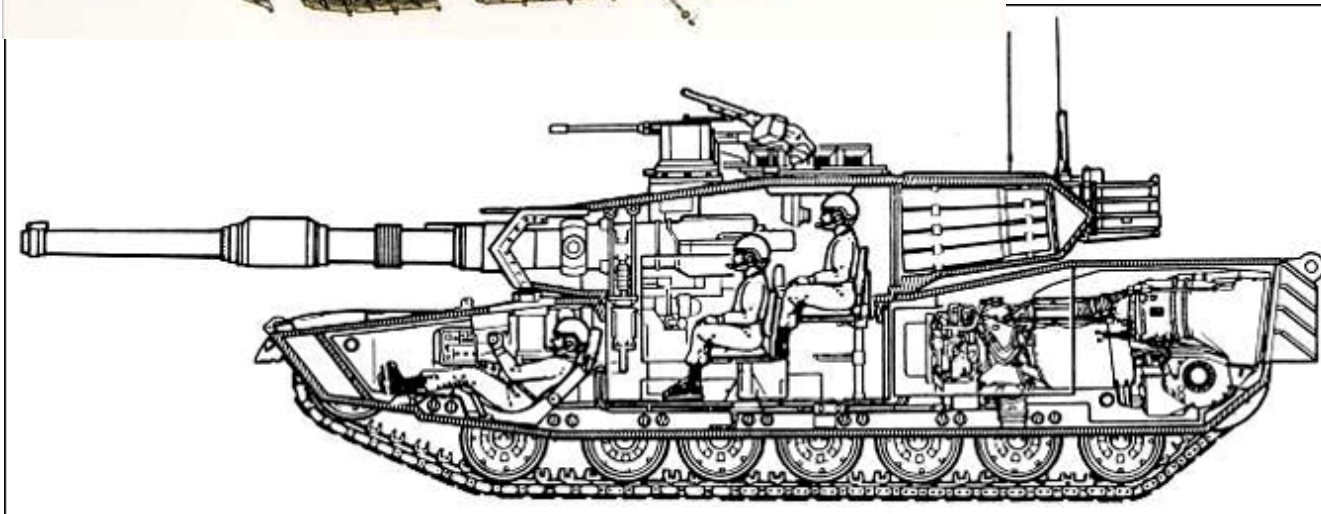
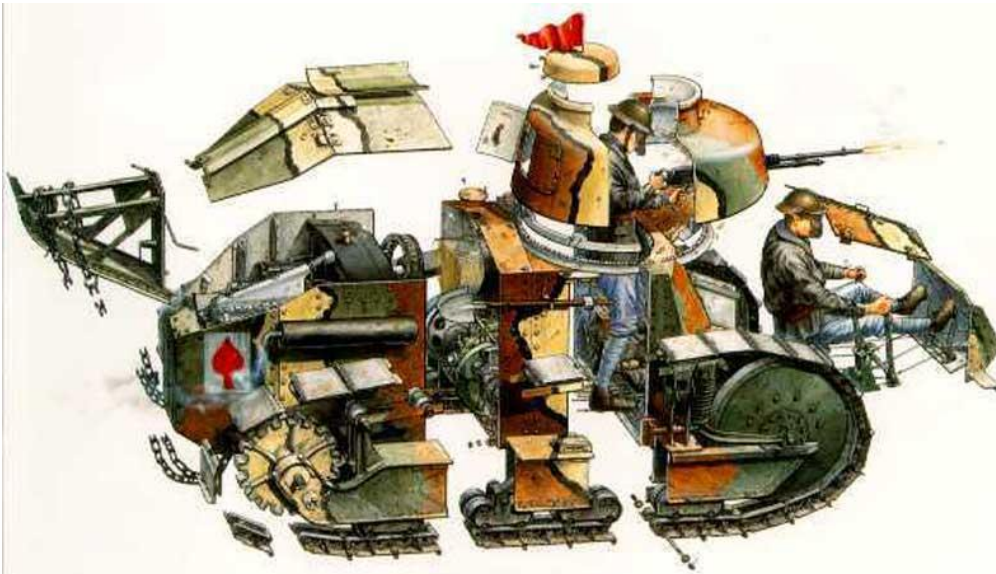
Glavna primena Aleksanderovih obrazaca

- ▶ **Software design patterns** - u razvoju softvera
- ▶ **Architectural patterns** - u arhitekturi i urbanizmu
- ▶ **Interaction design patterns** - u HCI
- ▶ **Business patterns** - modeli poslovanja
- ▶ **Pedagogical patterns** - u nastavi

PRIRUČNIK BIROKRATSKIH FRAZA

Sledećih 10 fraza, predstavlja mali priručnik birokratskih fraza, koje će vam omogućiti da održite govor u trajanju od 40 sati, a da, pritom, odabranom auditorijumu ne kažete ništa. Autor se poslužio jednostavnim principom: svaku frazu podeliti na četiri dela, koji se mogu uzajamno kombinovati. Svi segmenti koji se nalaze u koloni **1** na tabeli, mogu se kombinovati po želji sa bilo kojom grupom reči iz kolone **2**, zatim kolone **3** i konačno, kolone **4**.

1	2	3	4
Dame i gospodo	realizacija programa koje smo usvojili	primorava nas na analiziranje	postojećih administrativnih i finansijskih uslova
S druge strane	prostor i mesto koje zauzima obrazovanje kadrova	igra značajnu ulogu u utvrđivanju	osovina budućeg razvoja
Tako isto	konstantni rast kvaliteta i kvantiteta naših aktivnosti	primorava nas na preciziranje i defunisanje	sistema opšte participacije
Ipak, ne zaboravimo da	sadašnja struktura organizacije	nam pomaže u pripremi i kovanju	stavova u skladu sa zadacima koje je utvrdila organizacija
Na taj način	novi model aktivnosti	garantuje veliko učešće u razradi	novih predloga
Svakodnevna praksa pokazuje da	razvoj raznih oblika aktivnosti	doprinosi u značajnoj meri stvaranju	osnova progresističkog obrazovanja
Nije potrebno isticati značaj ovih problema jer	naša stalna briga oko informisanja i propagande	u velikoj meri doprinosi uspostavljanju	sistema obrazovanja kadrova koji je prilagođen potrebama
Bogato i raznovrsno iskustvo i	jačanje i širenje strukture	nam omogućava bolju procenu	uslova koji odgovaraju ubezanju naših aktivnosti
Briga oko organizacije, a pre svega	konsultovanje širokih slojeva aktivista	predstavlja zanimljiv pokušaj isprobavanja	jendog modela razvoja
Visoki ideološki kriterijumi, kao i	pokretanje odlučne kampanje	započinju proces koji ima za cilj usavršavanje	raznih oblika uticaja



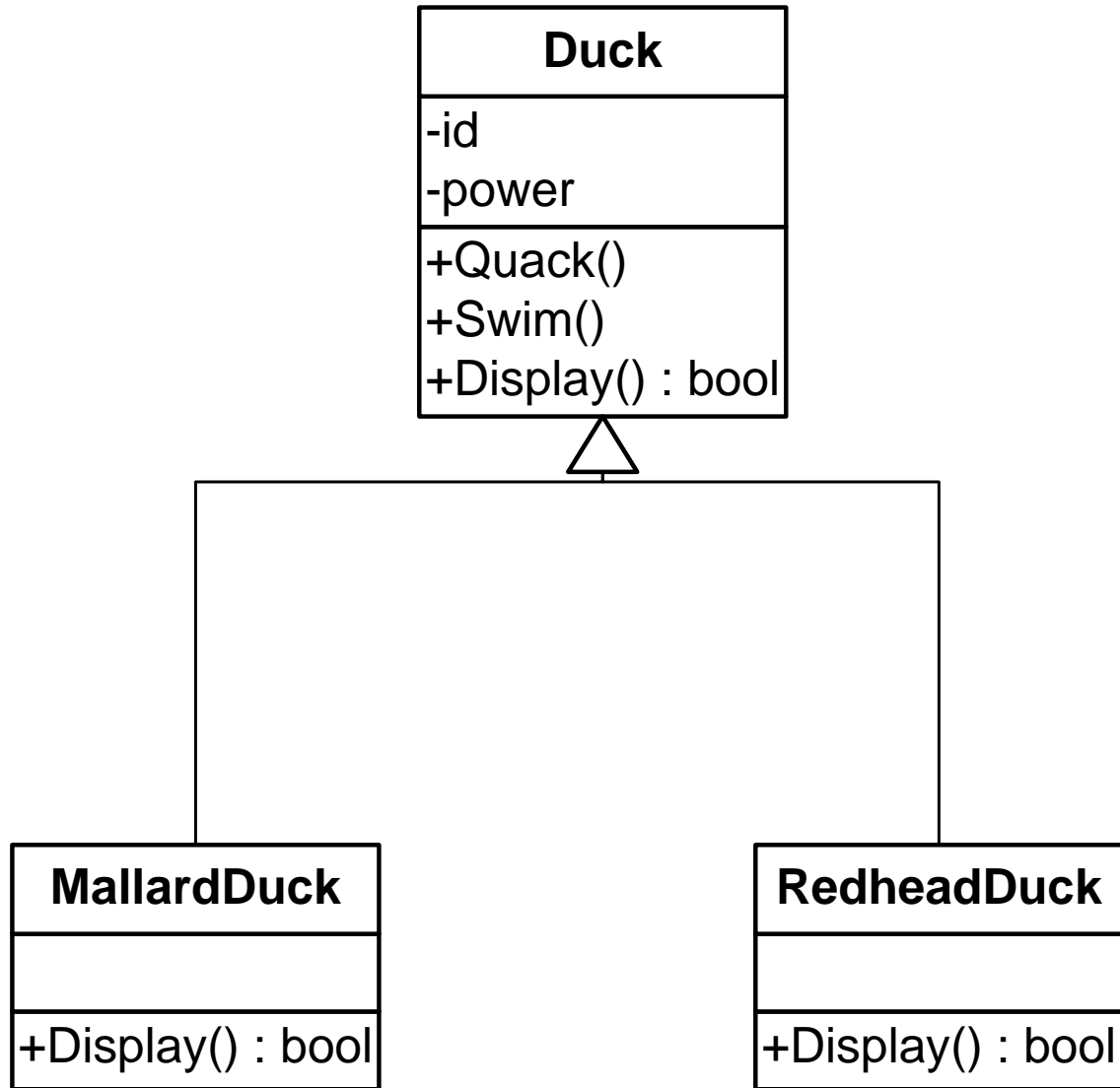
Prvi obrazac - šta je bilo na početku

- ▶ Na početku je postojala jednostavna igra gde je korisnik gađao iz puške plovke koje su plivale i koje su znale da kvaču (tj emituju sebi svojstvene tonove)
- ▶ Sve plovke su modelovane kao objekti iste klase - klase Duck

Duck
-id -power
+Quack() +Swim() +Display()

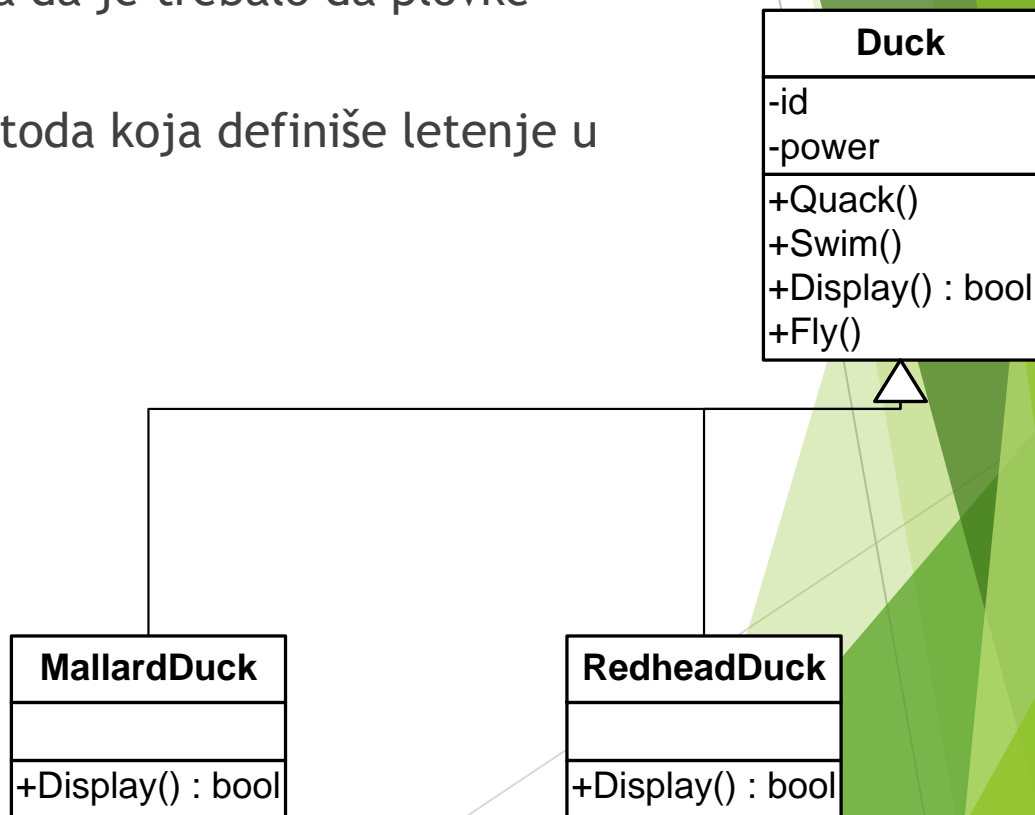
Prvi obrazac - šta je bilo dalje

- ▶ Uspeh igrice je motivisao kreatore da je unaprede tako što će uvesti u igru više različitih vrsta plovki koje će na drugačiji način biti predstavljene na ekranu
- ▶ Klasu Duck nasledilo je mnogo klasa, gde je metoda Display() postala virtuelna i predefinisana je u svakoj od nasleđenih klasa posebno
- ▶ Ducks v2 je brzo razvijen i imao je komercijalnog uspeha



Ducks v2.1

- ▶ Nova funkcionalnost je bila da je trebalo da plovke mogu i da lete
- ▶ Jednostavno je dodata metoda koja definiše letenje u osnovnoj klasi



[Play More Online Games](#)

[Buy Arcade Games for Windows](#)

[Download Free Arcade Games](#)

Duck Hunt

Instructions:

If you grew up in the 80's you
shouldn't need instructions!!

Controls

Use your mouse

This game will take a minute to load, but it is worth it!

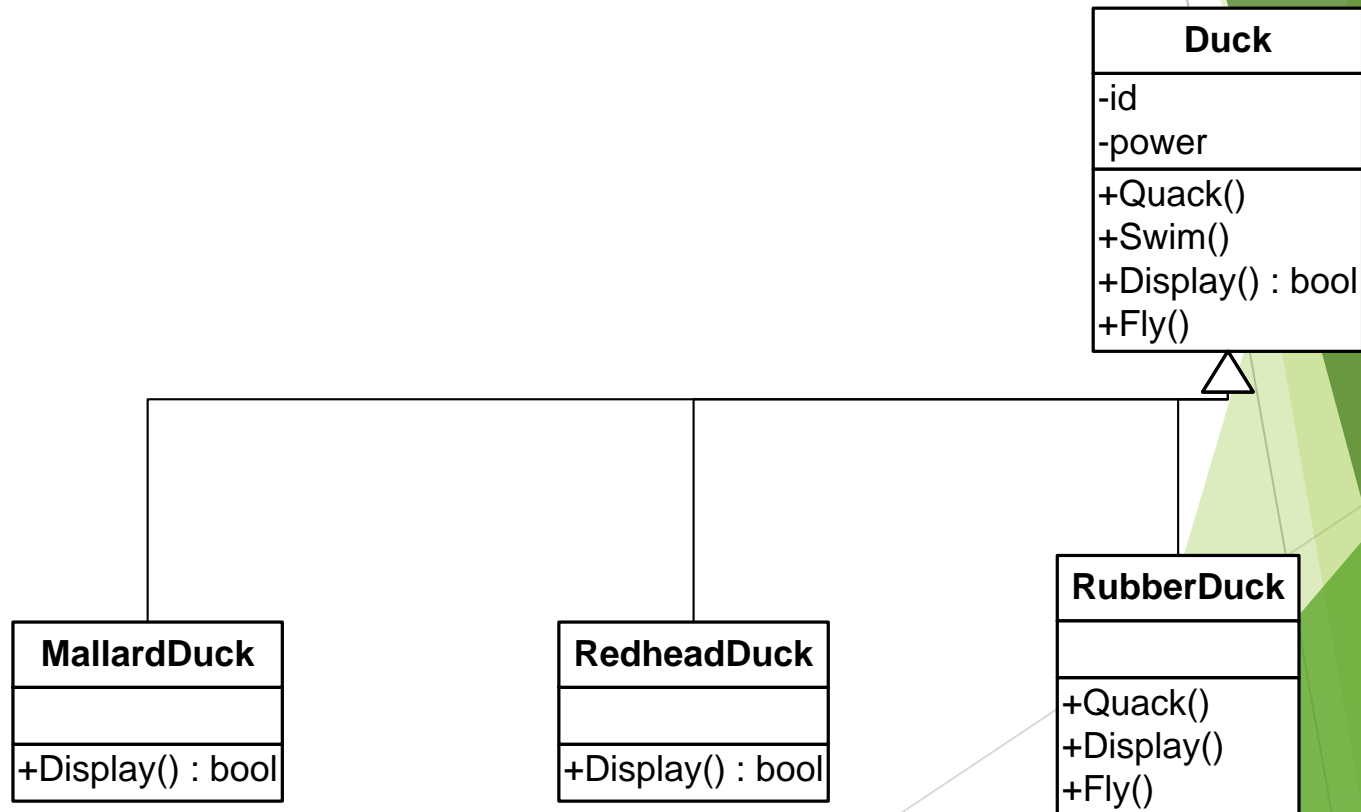


Ducks v2.2

- ▶ Odlučeno je da se pojave novi elementi - laj što su na primer gumene plovke - mamci.
- ▶ One imaju sasvim drugačija ponašanja za razliku od čivih plovki - ne kvaču nego puštaju drugačije zvuke
- ▶ Ne lete

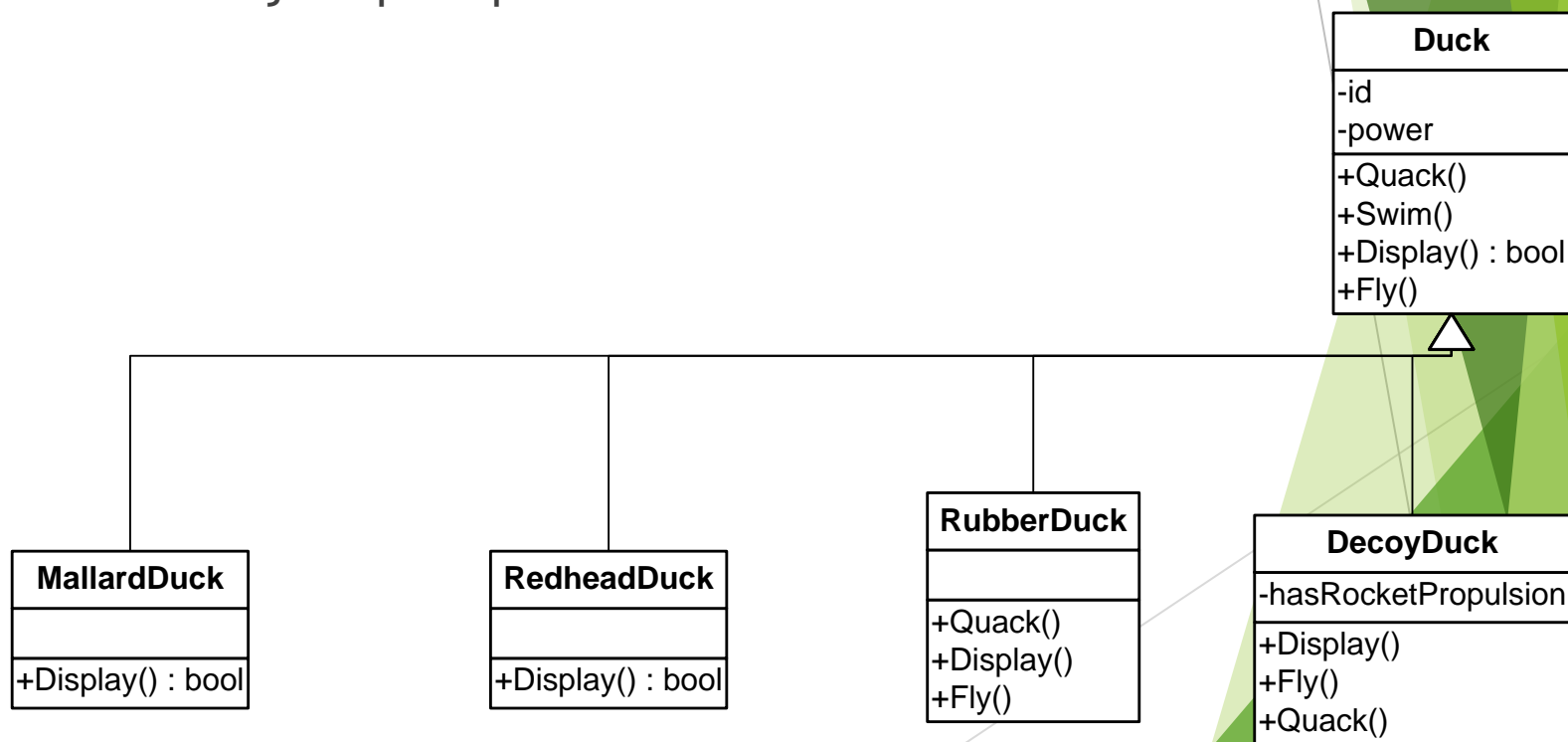
► Šta je moralo da se menja

- Quack mora da postane virtuelna fukcija da bi je RubberDuck predefinisala
- Fly mora da postane virtulena da bi mogla da se predefiniše
- Fly se predefiniše da ne radi ništa
 - Ovo baš i nije dobro (duplira se kod)
 - Biće još gore



Ducks v2.2a

- Pojavlja se potreba za drvenim plovkama mamcima na koje može da se montira raketni pogon
- Drvene plovke ne puštaju nikakav zvuk
- Mada može da im se montira sirena
- A sigurno će trebati još tipova plovki





1.2 :Multiplier
8 :Kill Spree



RELOAD

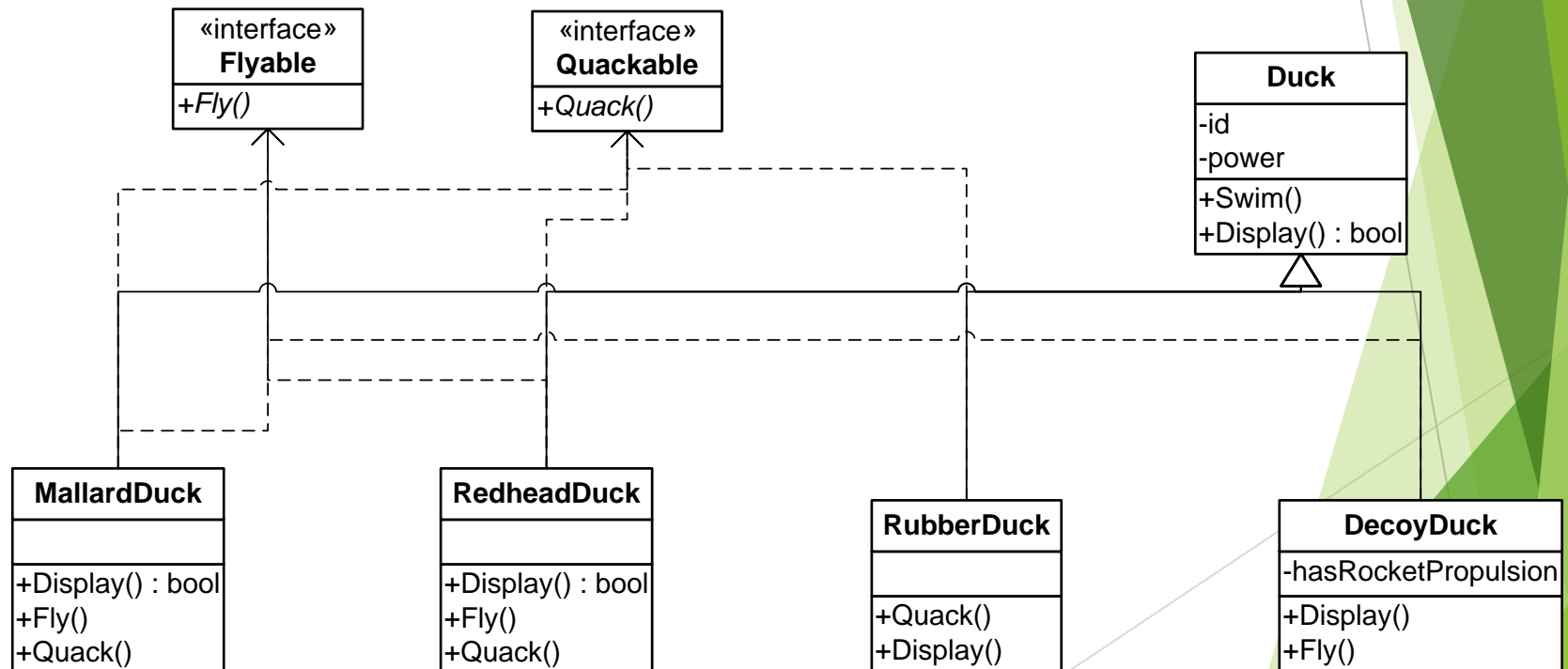


1

01:06
\$ 9950

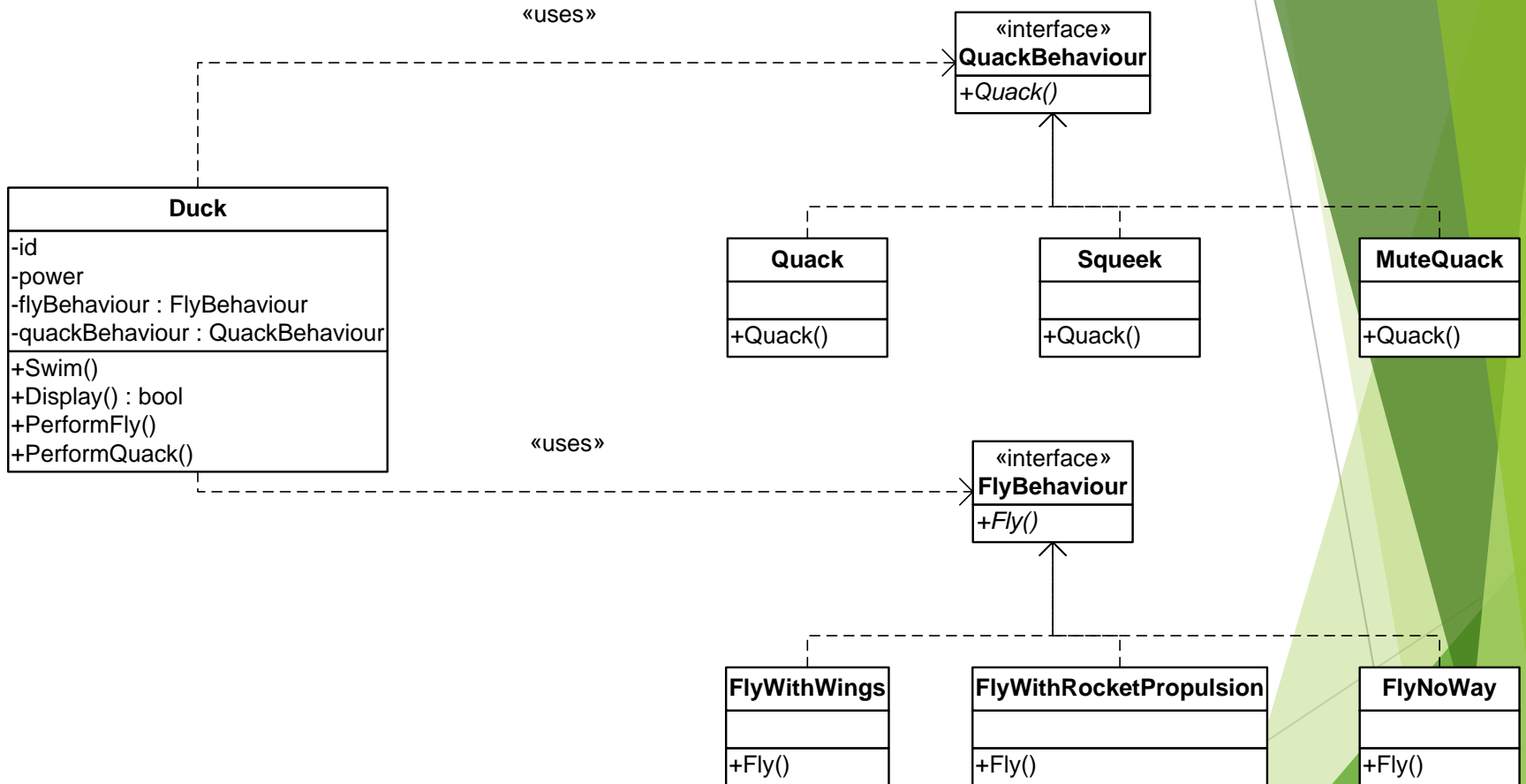
Obrazac 1 - instant rešenje

- ▶ Hajde da uvedemo interfejsse za Fly i Quack
- ▶ Ovde se tek duplira kod

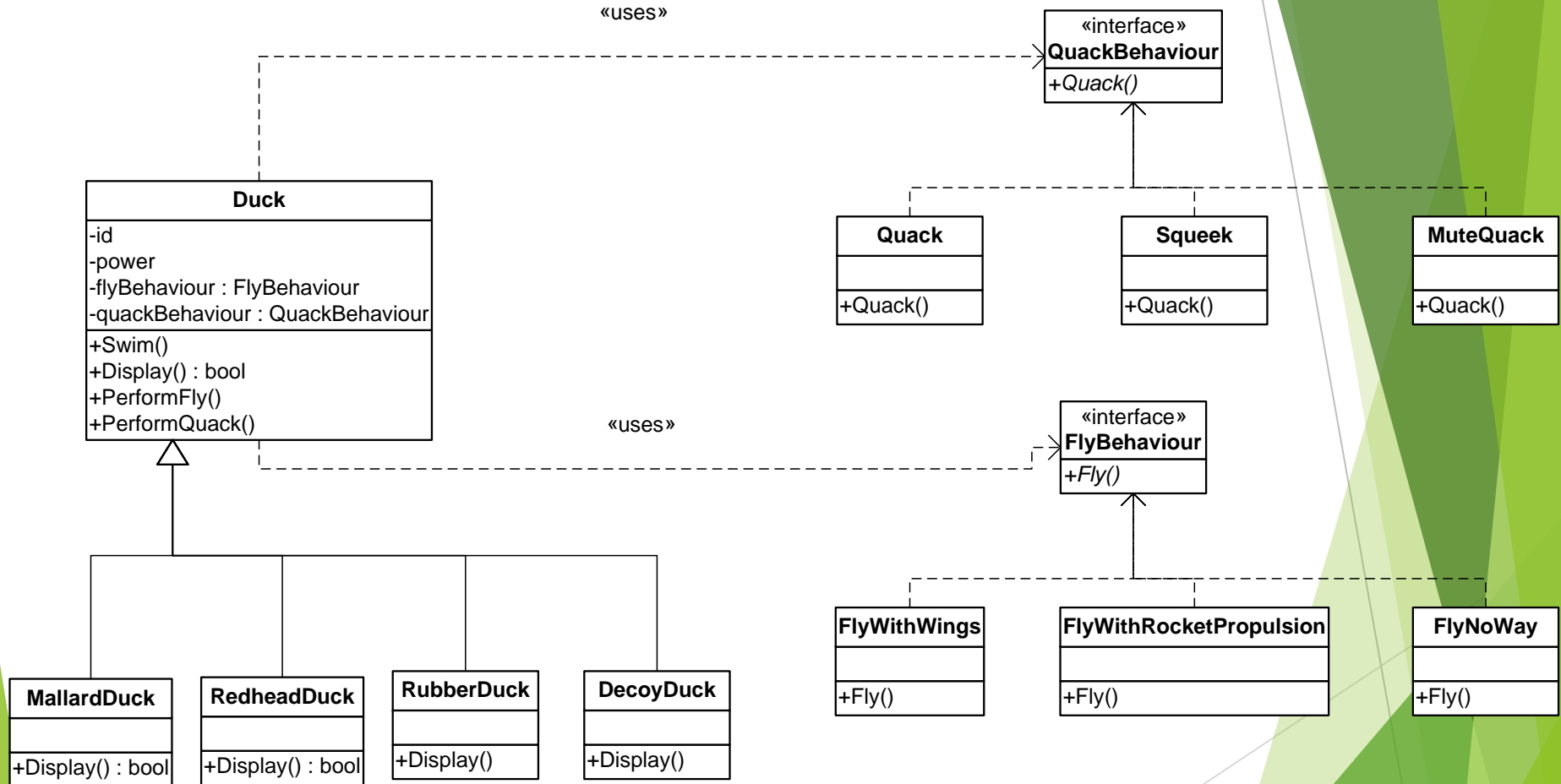


Šta raditi

- ▶ Znamo da su metodi Fly i Quack delovi originalne klase Duck, koji su vremenom postali promenljivi
- ▶ Da bi ih modelovali kako treba, izdvojićeno ih iz klase u interfejse, ali ćemo zato uvesti klase koje implementiraju posebne metode letenja i kvakanja
- ▶ U klasi Duck stavićemo reference na interfejs deo objekata



Ceo sistem

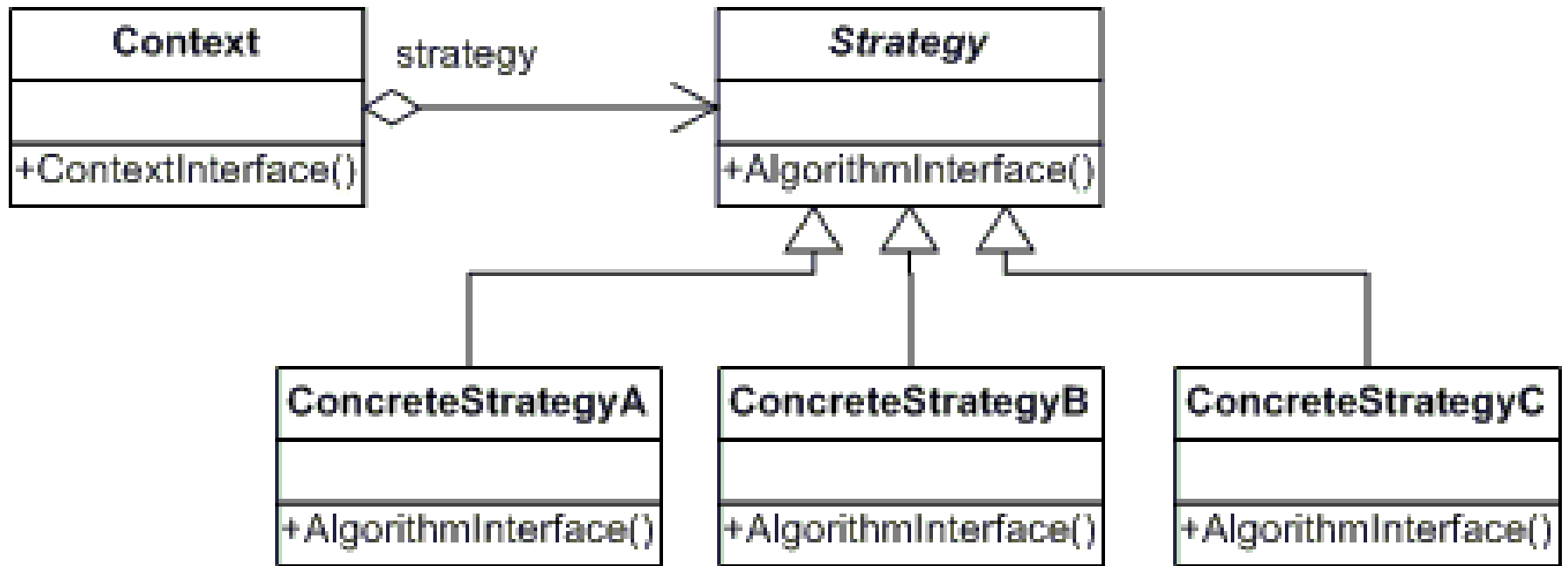




Prvi obrazac se zove Strategy

- ▶ Definiše familiju algoritama, enkapsulira svaki od njih, i čini ih međusobno dostupnim. Strategija dozvoljava algoritmu može biti promenljiv nezavisno od klijenata koji ga koriste.
- ▶ [strategy.cs](#)

UML Diagrama klase



Elementi

► Strategy (SortStrategy)

- deklarise zajednicki interfejs za sve podrzane algoritme. Objekti klase kontekst koriste ovaj interfejs da pozovu algoritme definisane ConcreteStrategy familijom klasa.

► ConcreteStrategy (QuickSort, ShellSort, MergeSort)

- implementira algoritam koristeći interfejs klase Strategy

► Context (SortedList)

- konfiguriše se pomoću ConcreteStrategy objekata
- čuva referencu na objekat klase Strategy
- može da definiše interfejs koji objektu klase Strategy dozvoljava pristup podacima

Primer iz stvarnog sveta

- ▶ Ovaj primer iz stvarnog sveta prikazuje upotrebu obrasca Strategy koji enkapsulira različite algoritme sortiranja u formi objekata. Ovo klijentu omogućava da dinamički menja strategije sortiranja (ovde su podržani Quicksort, Shellsort i Mergesort).
- ▶ [strategyRW.cs](#)

Singleton - Definicija

- ▶ Obezbeđuje da određena klasa ima maksimalno jedan aktivni objekat, i obezbeđuje da on bude dostupan.
- ▶ [singleton.cs](#)

UML Dijagram klasa



Elementi

▶ Singleton (LoadBalancer)

- ▶ definiše operaciju Instance koja obezbeđuje korisniku pristup jedinom kreiranom objektu
- ▶ definiše konstruktor koji je odgovoran za kreiranje i održavanje svog jedinog objekta

Singleton implementation



Conventional implementation





Primer iz stvarnog sveta

- ▶ Ovaj primer prikazuje primenu obrasca Singleton na objekat klase LoadBalancing. Kreiraće se samo jedna instanca klase zato što server može dinamički da menja stanje iz on-line u off-line. Zbog toga će se zahtev prosleđivati kroz jedan objekat koji će voditi računa o stanju cele web farme.
- ▶ [singletonRW.cs](#)

Adapter - Definicija

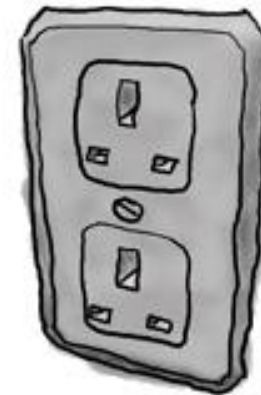
- ▶ Konvertuje interfejs jedne klase u interfejs kakav očekuje druga klasa. Obrazac Adapter omogućava da klase, koje inače ne bi mogle zbog različitih interfejsa, funkcionišu zajedno.
- ▶ [adapter.cs](#)



ADAPTER
US \leftrightarrow UK

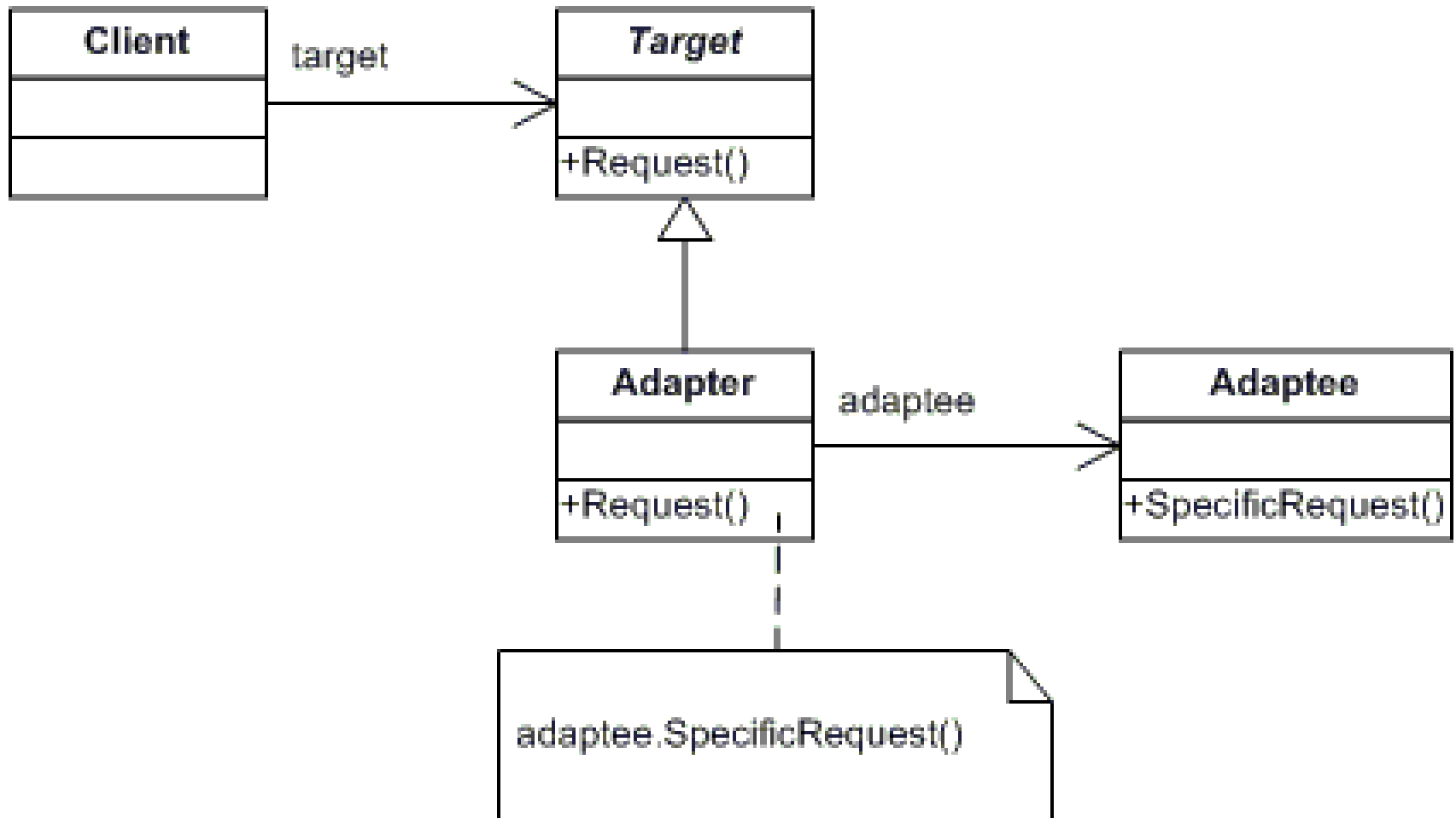


US PLUG



UK WALL OUTLET

UML Dijagram klasa



Elementi

- ▶ **Target (ChemicalCompound)**
 - ▶ definiše specifični interfejs koji koristi klasa Client.
- ▶ **Adapter (Compound)**
 - ▶ adaptira interfejs klase Adaptee prema interfejsu klase.
- ▶ **Adaptee (ChemicalDatabank)**
 - ▶ definiše postojeći interfejs koji treba adaptirati.
- ▶ **Client (AdapterApp)**
 - ▶ vodi računa o objektima koji zahtevaju interfejs klase Target

Primer iz stvarnog sveta

- ▶ Ovaj primer iz stvarnog sveta prikazuje upotrebu banke podataka hemijskih jedinjenja. Složena hemijska jedinjenja pristupaju banci podataka na osnovu interfejsa koji im je omogućio interfejs klase Adapter.
- ▶ adapterRW.cs

Primena

- ▶ Setimo se prvog primera sa plovkama gde je upotrebljen obrazac Strategy
- ▶ Naš sistem u prvom primeru je radio sa plovkama i ceo sistem zna kako sa njima da radi
- ▶ U međuvremenu, odlučimo da u ugricu dodamo i ćurke koje su već definisane u drugom sistemimu
- ▶ Hajde da ih adaptiramo u plovke

