Polimorfizam „under the hood”

KARLO DIMJAŠEVIĆ

Sadržaj

- Polimorfizam općenito
- Vrste polimorfizma
- Uvodni primjer
  - Primjer u C++
  - Primjer u Javi
  - Primjer u Pythonu
- **Polimorfizam „under the hood”**
- Zaključak

Polimorfizam općenito

- višeobličje – više oblika
- van računarstva
  - biologija – javljanje različitih oblika jedinki unutar jedne biljne ili životinjske vrste
  - kemija – odlika nekih spojeva ili elemenata da pri istom kemijskom sastavu kristaliziraju u kristalnim oblicima različite simetrije
  - izvor: Hrvatski jezični portal
- u računarstvu
  - *Stroustrup 2007: provision of a single interface to entities of different types*
  - *Cardelli, Wegner 1985: A polymorphic type is a type whose operations can also be applied to values of some other type, or types.*

Vrste polimorfizma (1)

- implicitni polimorfizam
  - makro naredbe u C-u
- ad-hoc polimorfizam
  - preopterećivanje funkcija
- **parametarski polimorfizam**
  - predlošci i generici
- **hijerarhijski polimorfizam**
  - podtipovi i nasljeđivanje

Vrste polimorfizma (2)

- statički polimorfizam
  - polimorfni poziv poznat već nakon prevođenja (*compile-time*)
  - primjer: C++ i predlošci (*template*)
- dinamički polimorfizam
  - polimorfni poziv poznat tek tijekom izvođenja (*run-time*)
  - primjer: nasljeđivanje razreda ili sučelja (C++, Java i Python), duck typing (Python)

Vrste polimorfizma (3)

Statički polimorfizam

- nakon prevođenja imamo 3 **asemblerske** izvedbe predloška **compare**
  - int
  - string
  - double
- **inline** umeće strojni kod na **mjesto pozivanja** predloška compare

Dinamički polimorfizam

U nastavku...

```
#include <iostream>

using namespace std;

template <class T>
inline int compare(T x, T y) {
    if (x < y) return -1;
    else if (x == y) return 0;
    else return 1;
}

int main (){
    cout << compare(3, 7) << endl;
    cout << compare("cloud", "route") << endl;
    cout << compare(24.5, 23.5) << endl;
}
```

Uvodni primjer

Želimo kreirati osnovno **sučelje BasicAPIClient** koje predstavlja ugovor za rad s API klijentom (CWA, Route, JIRA, ...). Sučelje deklarira određene metode (vidi kod). Konkretni API klijenti trebaju implementirati BasicAPIClient sučelje i ponuditi svoju **konkretnu** implementaciju.

Glavni program ima definiranu funkciju **action** koja prihvaća objekt tipa BasicAPIClient i URL endpoint-a. Funkcija uspostavlja konekciju prema API klijentu, radi HTTP-GET request na zadani URL te potom prekida konekciju.

Cilj: Funkcija action ne treba biti svjesna kako se pojedini koraci izvode (uspostava konekcije, slanje HTTP-GET zahtjeva, ukidanje konekcije).

Rješenje?

Polimorfizam

Polimorfizam i C++ (1)

base_api_client.h

```
#pragma once
#include <iostream>

using namespace std;

class BaseAPIClient {
public:
    virtual ~BaseAPIClient();
    virtual string getServiceName()=0;
    virtual int connect()=0;
    virtual int get(string url)=0;
    virtual int disconnect()=0;
};
```

Polimorfizam i C++ (2)

base_api_client.cpp

```cpp
#pragma once
#include "base_api_client.h"

BaseAPIClient::~BaseAPIClient() {};
```

Polimorfizam i C++ (3)

cwa_api_client.h

```
#include "base_api_client.h"

class CWAAPIClient: BaseAPIClient {
public:
    string getServiceName();
    int connect();
    int get(string url);
    int disconnect();
};
```

Polimorfizam i C++ (4)

cwa_api_client.cpp

```cpp
#include "cwa_api_client.h"
#include <iostream>

using namespace std;

string CWAAPIClient::getServiceName() {
    return "Cloud Web App";
}

int CWAAPIClient::connect() {
    cout << "Connecting to " << this->getServiceName() << "..." << endl;
    cout << "Connected!" << endl;
    return 0;
};

int CWAAPIClient::get(std::string url) {
    cout << "Sending HTTP-GET request to " << this->getServiceName() << ": " << url << endl;
    return 200;
}

int CWAAPIClient::disconnect() {
    cout << "Disconnecting from " << this->getServiceName() << "..." << endl;
    cout << "Disconnected!" << endl;
    return 0;
}
```

Polimorfizam i C++ (5)

main.cpp

```cpp
#include "cwa_api_client.h"
#include "route_api_client.h"

using namespace std;

template<typename BaseAPIClient>
void action(BaseAPIClient client, string url) {
    cout << endl;
    client.connect();
    int statusCode = client.get(url);
    cout << "Action completed with status code: " << statusCode << endl;
    client.disconnect();
}

int main() {
    CWAAPIClient cwa;
    RouteAPIClient route;

    action(cwa, "/api/v1/machines");
    action(route, "/api/v1/routes");

    return 0;
}
```

Polimorfizam i C++ (6)

./compileAndRun.sh

```
Connecting to Cloud Web App...
Connected!
Sending HTTP-GET request to Cloud Web App: /api/v1/machines
Action completed with status code: 200
Disconnecting from Cloud Web App...
Disconnected!

Connecting to Route...
Connected!
Sending HTTP-GET request to Route: /api/v1/routes
Action completed with status code: 200
Disconnecting from Route...
Disconnected!
```

Polimorfizam i Java (1)

BasicAPIClient.java

```java
package hr.intis.cloud.examples.api;

public interface BasicAPIClient {

    String getServiceName();

    int connect();

    int get(String url);

    int disconnect();
}
```

Polimorfizam i Java (2)

CWAAPIClient.java

```java
package hr.intis.cloud.examples.api;

public class CWAAPIClient implements BasicAPIClient {

    public CWAAPIClient() {
    }

    @Override
    public String getServiceName() {
        return "Cloud Web App";
    }

    @Override
    public int connect() {
        System.out.printf("Connecting to %s...\n", getServiceName());
        System.out.println("Connected!");
        return 0;
    }

    @Override
    public int get(String url) {
        System.out.printf("Sending HTTP-GET request to %s: %s\n", getServiceName(), url);
        return 200;
    }

    @Override
    public int disconnect() {
        System.out.printf("Disconnecting from %s...\n", getServiceName());
        System.out.println("Disconnected!");
        return 0;
    }
}
```

Polimorfizam i Java (3)

Main.java

```java
package hr.intis.cloud.examples;

import hr.intis.cloud.examples.api.BasicAPIClient;
import hr.intis.cloud.examples.api.CWAAPIClient;
import hr.intis.cloud.examples.api.RouteAPIClient;


public class Main {

    public static void action(BasicAPIClient client, String url) {
        System.out.println();
        client.connect();
        int statusCode = client.get(url);
        System.out.printf("Action completed with status code: %d\n", statusCode);
        client.disconnect();
    }

    public static void main(String[] args) {
        BasicAPIClient cwa = new CWAAPIClient();
        BasicAPIClient route = new RouteAPIClient();

        action(cwa, "/api/v1/machines");
        action(route, "/api/v1/routes");
    }
}
```

Polimorfizam i Java (4)

./compileAndRun.sh

```
Connecting to Cloud Web App...
Connected!
Sending HTTP-GET request to Cloud Web App: /api/v1/machines
Action completed with status code: 200
Disconnecting from Cloud Web App...
Disconnected!

Connecting to Route...
Connected!
Sending HTTP-GET request to Route: /api/v1/routes
Action completed with status code: 200
Disconnecting from Route...
Disconnected!
```

Polimorfizam i Python (1)

route_api_client.py

```python
class RouteAPIClient:

    def get_service_name(self):
        return "Route"

    def connect(self):
        print(f"Connecting to {self.get_service_name()}...")
        print("Connected!")
        return 0

    def get(self, url):
        print(f"Sending HTTP-GET request to {self.get_service_name()}: {url}")
        return 200

    def disconnect(self):
        print(f"Disconnecting from {self.get_service_name()}...")
        print("Disconnected!")
        return 0
```

Polimorfizam i Python (2)

main.py

```python
from cwa_api_client import CWAAPIClient
from route_api_client import RouteAPIClient


def action(client, url):
    print()
    client.connect()
    status_code = client.get(url)
    print(f"Action completed with status code: {status_code}")
    client.disconnect()


def main():
    cwa = CWAAPIClient()
    route = RouteAPIClient()

    action(cwa, "/api/v1/machines")
    action(route, "/api/v1/routes")


main()
```

Polimorfizam i Python (3)

python main.py

```
Connecting to Cloud Web App...
Connected!
Sending HTTP-GET request to Cloud Web App: /api/v1/machines
Action completed with status code: 200
Disconnecting from Cloud Web App...
Disconnected!

Connecting to Route...
Connected!
Sending HTTP-GET request to Route: /api/v1/routes
Action completed with status code: 200
Disconnecting from Route...
Disconnected!
```

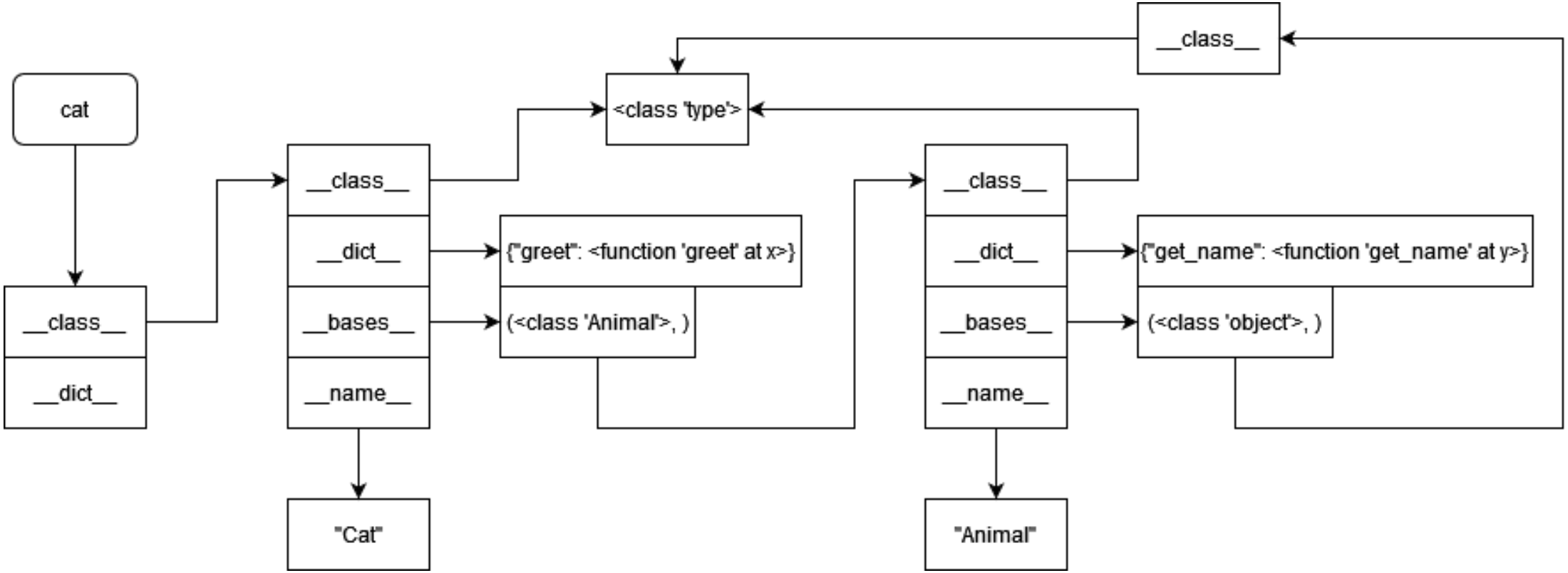
Polimorfizam „under the hood”

Kako polimorfizam funkcionira? (2)

- razmatrat ćemo **Python**

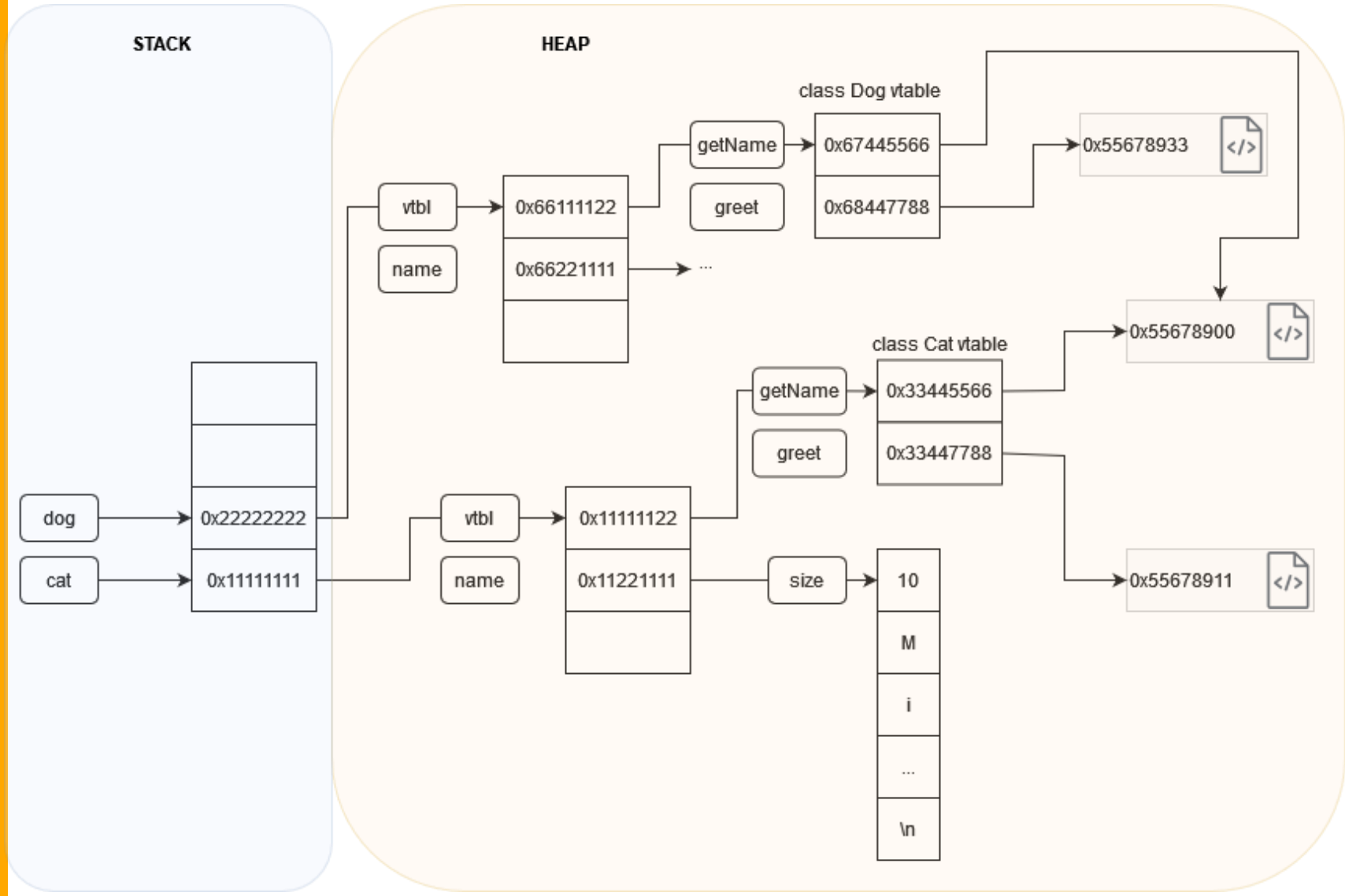
cat.greet()

- što se događa?
  - asocijativni pristup rječniku objekta cat
  - u slučaju neuspjeha, asocijativni pristup rječniku razreda Cat
  - u slučaju neuspjeha, asocijativni pristup rječniku razreda Animal
  - ...
  - općenito, rekurzivan poziv do vrha hijerarhije nasljeđivanja - *<class 'object'>*



Asocijativni rječnik

Virtualne tablice



Zaključak

- **statički** polimorfizam vs. **dinamički** polimorfizam
  - manja vremenska složenost - odredište polimorfnog poziva poznato nakon prevođenja (*compile-time*)
  - manja prostorna složenost - nema strukturnih podataka
    (virtualne tablice u C++ i Javi, asocijativni rječnik u Pythonu)
  - korištenje predložaka izaziva veći izvršni kod
  - nakon prevođenja nestaje fleksibilnost poziva
- „stari kod poziva novi kod”
- asocijativno polje vs. virtualna tablica
  - fleksibilnija primjena (*duck typing*)
  - mogućnost dodavanja metoda tijekom izvođenja
  - veća vremenska i prostorna složenost

Hvala na pažnji!

https://github.com/Karlito16/Polymorphism-Under-The-Hood