**OOP polymorfismus, E-shop a CMS, HTTPS protokol**

## **Objektově orientované programování (OOP)**

### **Co je OOP a proč ho využíváme?**

OOP (Objektově orientované programování) je programovací paradigma, které umožňuje strukturovat kód pomocí **objektů** a **tříd**. Tento přístup napodobuje reálný svět, kde objekty mají **vlastnosti (atributy)** a **chování (metody)**. Hlavní důvody pro využití OOP:

* **Znovupoužitelnost kódu** – Díky dědičnosti můžeme opětovně využít existující třídy a rozšiřovat je.
* **Lepší organizace kódu** – Kód je přehlednější a lépe škálovatelný.
* **Zapouzdření** – Skrýváme detaily implementace a zpřístupňujeme jen potřebné části.
* **Polymorfismus** – Umožňuje používat stejné rozhraní pro různé datové typy.

### **Základní charakteristiky OOP**

1. **Třídy a objekty** – Třída je šablona, podle které se vytvářejí objekty.
2. **Dědičnost** – Umožňuje, aby jedna třída přebírala vlastnosti a metody jiné třídy.
3. **Zapouzdření** – Ochrana dat a metod uvnitř objektu.
4. **Polymorfismus** – Možnost volat stejnou metodu na různých třídách.

## **Polymorfismus na příkladu v Pythonu**

Polymorfismus znamená, že stejná metoda může mít různé chování v různých třídách. V poskytnutém kódu je polymorfismus vidět v metodách vypocti\_obvod(), vypocti\_obsah() a params(), které jsou definovány v základní třídě Geometricky\_utvar, ale každá dědičná třída (Ctverec, Obdelnik, Kruh) si je implementuje jinak.

class Geometricky\_utvar:  
 def vypocti\_obvod(self):  
 return None  
 def vypocti\_obsah(self):  
 return None

Třída Geometricky\_utvar definuje metody, ale nemá jejich konkrétní implementaci.

class Ctverec(Geometricky\_utvar):  
 def vypocti\_obvod(self):  
 return 4 \* self.strana  
 def vypocti\_obsah(self):  
 return self.strana \*\* 2

V třídě Ctverec jsou tyto metody implementovány konkrétně pro čtverec. To samé platí pro Obdelnik a Kruh.

Díky polymorfismu můžeme stejným způsobem pracovat s různými tvary:

utvary = [Ctverec(5), Kruh(4), Obdelnik(5,2)]  
for u in utvary:  
 print(u.nazev, 's parametry', u.params(), 'ma obvod', u.vypocti\_obvod(), 'a obsah', u.vypocti\_obsah(), '.')

I když voláme vypocti\_obvod() a vypocti\_obsah() na objektech různého typu, Python automaticky spustí odpovídající implementaci pro každý konkrétní objekt. **To je polymorfismus v praxi!**

## **Tvorba e-shopu, CMS Joomla a WordPress**

### **Jak vytvořit e-shop?**

1. **Doména a hosting** – Zaregistrování domény (např. [www.mujshop.cz](http://www.mujshop.cz/)) a zakoupení hostingu.
2. **Výběr platformy** – Možnosti:
   1. **Vlastní řešení** (programování od základu v PHP, Pythonu, nebo Node.js)
   2. **Použití CMS** (např. WordPress + WooCommerce)
   3. **Specializované platformy** (Shoptet, Shopify, Prestashop)
3. **Design a UI** – Moderní a responzivní vzhled, UX optimalizace.
4. **Produkty a platby** – Napojení platební brány (GoPay, PayPal) a správa produktů.
5. **SEO a marketing** – Optimalizace pro vyhledávače, sociální sítě, reklamy.

### **CMS Joomla a WordPress**

* **CMS (Content Management System)** umožňuje snadnou správu obsahu webu.
* **WordPress** – Nejpopulárnější, snadno rozšířitelný, ale méně bezpečný.
* **Joomla** – Robustnější, lepší pro větší projekty, ale složitější na správu.

**Výhody CMS:** ✔ Snadná správa obsahu ✔ Mnoho šablon a pluginů ✔ Široká komunita a podpora

**Nevýhody CMS:** ✘ Zranitelnost vůči útokům ✘ Omezené možnosti přizpůsobení oproti vlastnímu kódu

## **HTTPS protokol**

### **Co je HTTPS?**

HTTPS (HyperText Transfer Protocol Secure) je šifrovaná verze HTTP, která zabezpečuje komunikaci mezi uživatelem a serverem pomocí SSL/TLS.

### **Jak funguje HTTPS požadavek?**

1. **Z webového prohlížeče** – Stačí zadat URL do prohlížeče.
   1. Například: <https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45>
2. **Z terminálu pomocí curl:**

curl "<https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45>"

1. **Z Pythonu pomocí requests:**

import requests  
response = requests.get("https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45")  
print(response.json())

### **Co je URL a parametry GET?**

* **URL (Uniform Resource Locator)** je adresa webové stránky.
* **GET parametry** jsou součástí URL za ? a umožňují předat data serveru.
* Například: ?lat=36.900&lng=-4.45 předává zeměpisnou šířku a délku API.

## **Shrnutí - Co se naučit**

* **Co je OOP a jeho hlavní principy (třídy, dědičnost, zapouzdření, polymorfismus)**
* **Polymorfismus na příkladu v Pythonu**
* **Co je e-shop, jak ho vytvořit, jaké platformy existují**
* **Co je CMS, rozdíly mezi WordPress a Joomla**
* **Co je HTTPS, jak probíhá komunikace mezi klientem a serverem**
* **Jak poslat HTTPS požadavek přes prohlížeč, curl a requests v Pythonu**
* **Co je URL a parametry GET**