# OOP, instalace SW v Unix like systémech, GPIO Raspberry Pi 4

## OOP obecně a jeho význam

**Objektově orientované programování (OOP)** je programovací paradigma založené na objektech, které obsahují data (atributy) a metody (funkce). Tento přístup umožňuje lepší strukturování kódu, jeho znovupoužitelnost a snadnější údržbu.

### Základní charakteristiky OOP:

1. **Abstrakce** – Skrývání složitosti pomocí jednoduchého rozhraní.
2. **Enkapsulace** – Skrytí implementace objektu a zpřístupnění jen potřebných částí.
3. **Dědičnost** – Možnost odvozování nových tříd z již existujících.
4. **Polymorfismus** – Možnost použití jedné metody pro různé typy objektů.

## Enkapsulace na příkladu v Pythonu

Enkapsulace znamená, že **vnitřní stav objektu je chráněný a nemůže být přímo změněn zvenčí**. Místo přímé manipulace s atributy poskytujeme metody pro jejich změnu.

### Vysvětlení enkapsulace na kódu v Příloze 2:

V kódu je **enkapsulace částečně porušena**, protože atributy jako strana, strana\_a, strana\_b, polomer jsou veřejné a tedy dostupné zvenčí. Pro správnou enkapsulaci by měly být tyto atributy privátní (začínající podtržítkem nebo dvěma podtržítky) a přístup k nim by měl být umožněn pouze přes metody.

Například místo:

self.strana = strana

by bylo vhodnější:

self.\_\_strana = strana # Privátní atribut

a k němu poskytnout metodu:

def get\_strana(self):  
 return self.\_\_strana

Tím by se zajistilo, že hodnota atributu nemůže být změněna přímo, ale pouze prostřednictvím metod, což je základní princip **enkapsulace**.

## Instalace software v Unix like systémech

V Linuxu existuje několik způsobů instalace softwaru:

1. **Použití balíčkovacího systému (terminál):**
   1. Debian/Ubuntu: sudo apt install název\_balíčku
   2. Fedora: sudo dnf install název\_balíčku
   3. Arch Linux: sudo pacman -S název\_balíčku
2. **Grafické rozhraní:**
   1. Software Center (Ubuntu)
   2. Discover (KDE)
3. **Alternativní balíčkovací systémy:**
   1. **Snap**: sudo snap install název\_balíčku
   2. **Flatpak**: flatpak install flathub název\_balíčku
4. **App Store a Google Play:**
   1. iOS používá **App Store**
   2. Android používá **Google Play**

Příklad příkazů pro **apt**:

sudo apt update # Aktualizace seznamu balíků  
sudo apt upgrade # Aktualizace všech balíků  
sudo apt remove název\_balíčku # Odstranění balíčku

## GPIO u Raspberry Pi 4

**GPIO (General Purpose Input/Output)** jsou univerzální piny na Raspberry Pi, které lze programovat pro vstup nebo výstup.

### Rozložení pinů

Najdeš na webu například hledáním "Raspberry Pi 4 GPIO pinout".

* **Napájecí piny:** 3.3V a 5V
* **Zemnící piny:** GND
* **Datové piny:** GPIO2, GPIO3, GPIO17 atd.

### Připojení periferie pro Arduino na GPIO Raspberry Pi

* Arduino využívá **5V logiku**, zatímco Raspberry Pi používá **3.3V logiku** → přímé propojení může poškodit Raspberry Pi.
* Řešení: Použití **logického převodníku napětí**.

## Shrnutí

### **Co se naučit k maturitě:**

1. **OOP:**
   1. Čtyři základní pilíře: **abstrakce, enkapsulace, dědičnost, polymorfismus**.
   2. **Enkapsulace znamená skrytí dat a jejich úpravu přes metody.**
   3. **Příklad v Pythonu** (s privátními atributy a metodami get\_nazev(), get\_obvod()).
2. **Instalace software v Unix-like systémech:**
   1. **apt příkazy:** install, update, upgrade, remove.
   2. **Balíčkovací systémy:** **Snap, Flatpak, App Store, Google Play**.
   3. **Grafické nástroje** (Software Center, Discover).
3. **GPIO Raspberry Pi 4:**
   1. **Co je GPIO:** univerzální piny pro vstup/výstup.
   2. **Rozložení pinů** (napájecí, zemnící, datové).
   3. **Problém s Arduino periferiemi (5V vs. 3.3V)** a nutnost převodníku napětí.

S tímhle bys měl bez problémů povídat 15 minut! 😉