

# Objektově orientované programování

## Přednáška č. 4

Ing. Lukáš Slánský

Ústav elektrotechniky a informatiky, Univerzita Pardubice

26. 10. 2006



# Obsah

- 1 Zavedení objektů pomocí postulátů
- 2 Vztahy mezi objekty
  - Příklad skládání objektů



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- 1 Objekt obsahuje *atributy* (proměnné)



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- 1 Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ❶ Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ❷ Objekt obsahuje *metody* (procedury, funkce)



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ① Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ② Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ① Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ② Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ③ Objekt má schopnost *přijmout a zpracovat zprávu*.





# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ① Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ② Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ③ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní*



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ① Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ② Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ③ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní* – veřejná množina zpráv.



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ① Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ② Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ③ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní* – veřejná množina zpráv.
  - ▶ *Protokol*



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ❶ Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ❷ Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ❸ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní* – veřejná množina zpráv.
  - ▶ *Protokol* – zobrazení (jednoznačné přiřazení) rozhraní do množiny metod.



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ❶ Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ❷ Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ❸ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní* – veřejná množina zpráv.
  - ▶ *Protokol* – zobrazení (jednoznačné přiřazení) rozhraní do množiny metod.
- ❹ Objekt může obsahovat jiné objekty.



# Definice objektu pomocí postulátů

**Definice:** Objekt je samostatná programová struktura, která je vytvořena podle axiomů:

- ❶ Objekt obsahuje *atributy* (proměnné)
  - ▶ = privátní (neveřejná) paměť objektu, která uchovává aktuální stav objektu.
- ❷ Objekt obsahuje *metody* (procedury, funkce)
  - ▶ = privátní (neveřejné) operace nad atributy, které mění stav objektu.
- ❸ Objekt má schopnost *přijmout a zpracovat zprávu*.
  - ▶ *Rozhraní* – veřejná množina zpráv.
  - ▶ *Protokol* – zobrazení (jednoznačné přiřazení) rozhraní do množiny metod.
- ❹ Objekt může obsahovat jiné objekty.
  - ▶ Může jim posílat zprávy, pomocí kterých jiné objekty řídí.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt





# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého.





"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.



"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty –



"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty – *dědění*.



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty – *dědění*.
  - ▶ Spolupráce ve smyslu



# Vztahy mezi objekty

"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty – *dědění*.
  - ▶ Spolupráce ve smyslu dopravní prostředek-auto.



"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty – *dědění*.
  - ▶ Spolupráce ve smyslu dopravní prostředek-auto.
  - ▶ Vnitřní objekt (dopravní prostředek) může existovat i bez vnějšího (auto).



"Objekt může obsahovat jiné objekty".

- Objekt může obsahovat odkaz (ukazatel) na vnější objekt – *asociace*.
  - ▶ Spolupráce ve smyslu auto-řidič.
  - ▶ Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.
- Objekt může mít jiný objekt jako svou autonomní část – *agregace*.
  - ▶ Spolupráce ve smyslu auto-motor.
  - ▶ Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.
- Objekt může mít jiný objekt jako základní část své podstaty – *dědění*.
  - ▶ Spolupráce ve smyslu dopravní prostředek-auto.
  - ▶ Vnitřní objekt (dopravní prostředek) může existovat i bez vnějšího (auto). Vnější objekt bez vnitřního existovat nemůže.



# Příklad skládání objektů

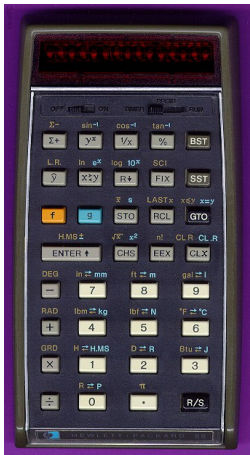
Navrhněte kalkulačku, která bude pracovat v obrácené polské notaci. Bude umět základní matematické funkce (+, -, \*, /).





# Příklad skládání objektů

Navrhněte kalkulačku, která bude pracovat v obrácené polské notaci. Bude umět základní matematické funkce (+, -, \*, /).



HP-55



# Příklad skládání objektů

Navrhněte kalkulačku, která bude pracovat v obrácené polské notaci. Bude umět základní matematické funkce (+, -, \*, /).



HP-55

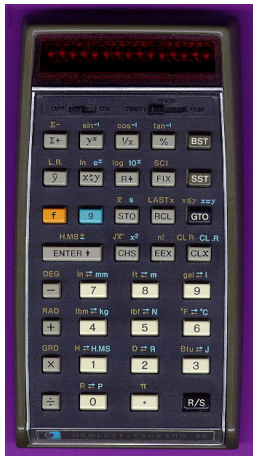


TI-59



# Příklad skládání objektů

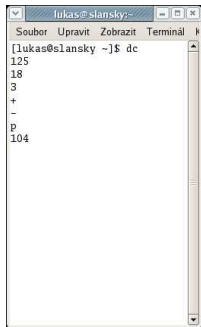
Navrhněte kalkulačku, která bude pracovat v obrácené polské notaci. Bude umět základní matematické funkce (+, -, \*, /).



HP-55



TI-59



Unixový program dc 

# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) –



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) – agregace.



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) – agregace.
- Interaktivní část (displej a klávesnice).





# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) – agregace.
- Interaktivní část (displej a klávesnice).
  - ▶ Neumí skoro nic, jen komunikovat s procesorem.



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) – agregace.
- Interaktivní část (displej a klávesnice).
  - ▶ Neumí skoro nic, jen komunikovat s procesorem.
  - ▶ Volná vazba na procesor (můžu ho vyměnit za jiný) –



# Příklad skládání objektů II.

Základní struktura programových bloků:

- Výpočetní část (simulace jednoduchého procesoru).
  - ▶ Nedělá žádný vstup z klávesnice ani výstup na obrazovku.
  - ▶ Jako svou část obsahuje střadač (zásobník) – agregace.
- Interaktivní část (displej a klávesnice).
  - ▶ Neumí skoro nic, jen komunikovat s procesorem.
  - ▶ Volná vazba na procesor (můžu ho vyměnit za jiný) – asociace.

