
Sa proceduralnog na OO programiranje: polimorfizam

- ❖ Dinamički polimorfizam, u vreme izvršavanja:

```
int fprintf (FILE* stream, char* format, ...);
```

- ❖ Apstrakcija *izlaznog znakovnog toka* (*output character stream*): ima operacije izbacivanja (“ispisa”) jednog po jednog znaka (sekvencijalno):

```
int putc (int ch, FILE* stream);
```

- ❖ Konkretna implementacija zavisi od toga šta je izlazni tok, odnosno šta se krije iza pokazivača *stream* tipa *FILE**: može biti “konzola”, fajl na lokalnom disku, fajl na udaljenom računaru itd. (ovaj polimorfizam implementiran je u sistemskom pozivu operativnog sistema, ne u samom jeziku C/C++ tj. njegovoj standardnoj biblioteci; ali je i tamo implementiran na jeziku C)
- ❖ Svaka konkretna vrsta izlaznog toka ima svoju implementaciju svake operacije (npr. *putc*)
- ❖ Iza pokazivača se može kriti različit izlazni tok, i to se može menjati *dinamički*, za vreme izvršavanja programa (u jednom pozivu operacije *fprintf* jedan, u drugom drugi izlazni tok)
- ❖ Potrebno je *dinamičko vezivanje* (*dynamic binding*): umesto u vreme prevođenja, adresa pozvanog potprograma određuje se dinamički, u vreme izvršavanja — potprogram koji se poziva se vezuje za poziv dinamički, u vreme izvršavanja, a ne statički u vreme prevođenja

Sa proceduralnog na OO programiranje: polimorfizam

❖ Drugi primer: pravimo program (na jeziku C) za igranje šaha na računaru. Skica delova:

```
enum FigureKind { pawn, bishop, knight, rook, queen, king };  
enum FigureColor { white, black };
```

```
struct Figure {  
    FigureKind kind;  
    FigureColor color;  
    unsigned posCol, posRow; // Current position  
    ...  
};
```

```
int canMoveTo (Figure* fig, unsigned col, unsigned row);
```

❖ Kako implementirati funkciju *canMoveTo* (da li data figura može da se pomeri na dato polje)?