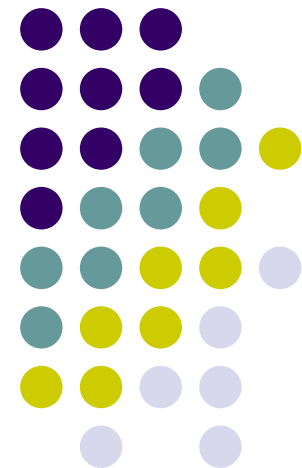


Računarski praktikum 1

Predavanja

Goran Igaly





Gradivo i način polaganja

- Sadržaj kolegija:
 - osnove objektnog programiranja
 - programski jezik C++
- Način polaganja ispita:
 - zadaće
 - kolokviji
 - aktivno sudjelovanje na nastavi

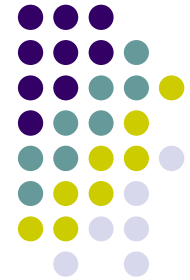


Evolucija programskih jezika

- Prva generacija – strojni jezik
- Druga generacija – assembler
- Treća generacija – Pascal, C, Fortran, Algol, LISP, ...
 - Uvodi se paradigma proceduralnog, strukturnog i modularnog programiranja
 - Pojavljuju se i višenamjenski jezici
 - Objektno-orijentirani jezici kao nova paradigma razvoja (Smalltalk, C++, Java, C#)

Četvrta generacija – 4GL jezici

Jezici specijalizirani za određenu domenu problema



- **Structured Query Language (SQL)**
najpopularniji programski jezik za kreiranje, dohvat, ažuriranje i brisanje podataka iz baze podataka
- **METAFONT** programski jezik za definiranje vektorskih fontova
- **PostScript (PS)** jezik za opisivanje izgleda stranice i programski jezik koji se prvenstveno koristi u elektroničkom i stolnom izdavaštvu.

Četvrta generacija – 4GL jezici (2)

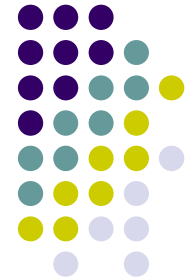


- **Maple** komercijalni matematički softverski paket opće namjene. Sastoji se od programskog jezika sa sučeljem koje omogućuje korisniku onos matematičkih formula na uobičajeni način
- **Mathematica** sustav za računalnu algebru, sadrži i viši programski jezik za manipulaciju izrazima
- **Oracle Forms** okruženje za razvoj aplikacija za rad s bazama podataka. Podržava programski jezik PL/SQL. Ideja: razvoj aplikacije za rad s bazama podataka s minimumom programskog koda.

Četvrta generacija – 4GL jezici (popularna razvojna platforma)



- **Visual Basic (VB)** programski jezik upravljani događajima (event) uz pripadno razvojno okruženje (Microsoft) za korištenje programskog modela COM. VB je zamijenjen razvojnom platformom Visual Basic .NET
- **Component Object Model (COM)** – Microsoftova platforma za razvoj softverskih komponenti (1993)
- Ideja **softverskih komponenti**: softverske komponente, jednako kao hardverske komponente (korištene npr u telekomunikacijama) moraju biti međusobno zamjenjive i pouzdane



Komunikacija s korisnikom

- **Windows Forms** - sučelje za programiranje aplikacija (API), uključeno u Microsoftov .NET Framework, za pristup standardnim elementima Windows sučelja putem Windows API.

```
using System.Windows.Forms;
using System;

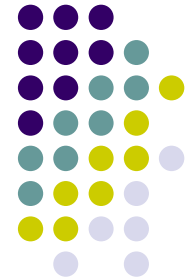
public class DobarDan
{
    [STAThread]
    public static void Main() {
        MessageBox.Show("Dobar dan!");
    }
}
```



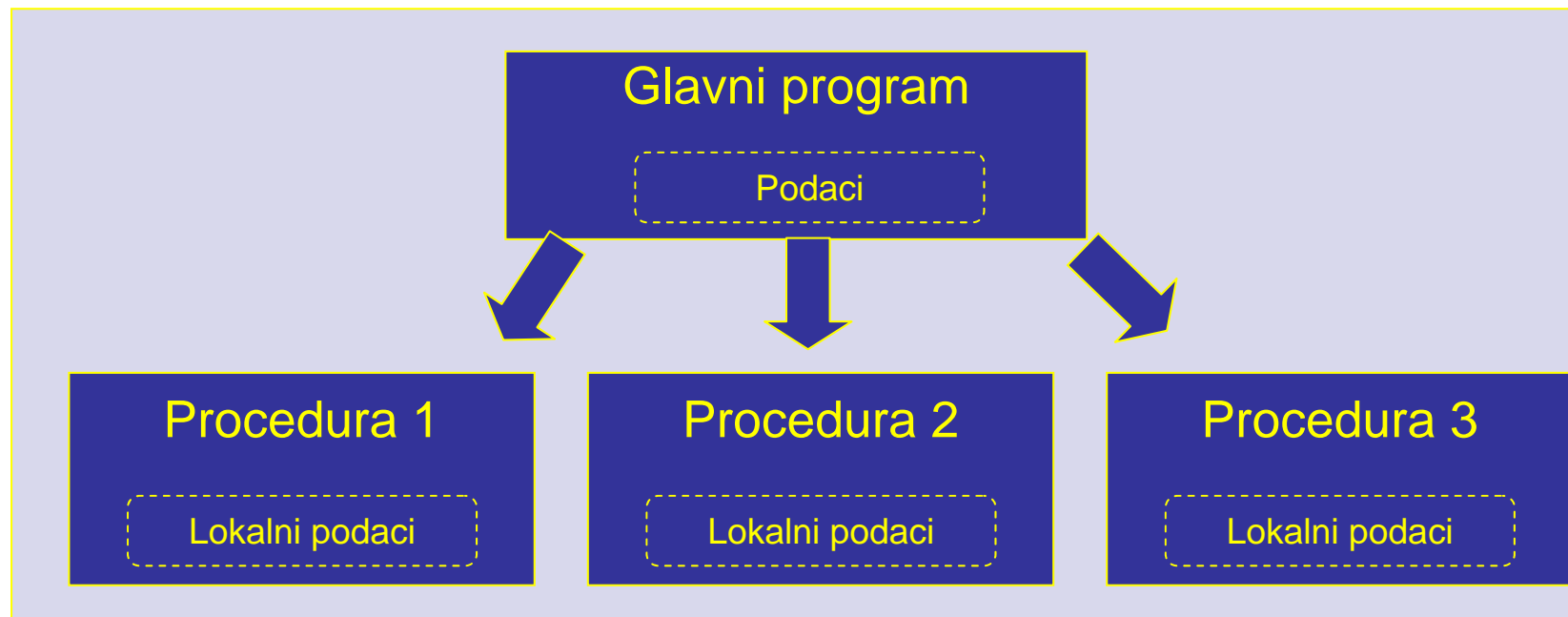
Razvoj tipičnog programera

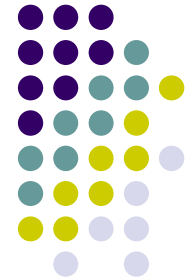
- Započinje učenjem programskog jezika 3GL
 - Basic, Pascal, C
 - učenje strojnog jezika i asemblera je nužno samo za programe za komunikaciju na najnižoj razini
- Koristi se paradigma **nestrukturiranog** programiranja
 - Započinje se pisanjem malih i jednostavnih programa koji se sastoje od slijeda naredbi i djeluju nad zajedničkim skupom podataka

Proceduralna paradigma



- Osnovni korak je uvođenje funkcija koje su izgrađene kao skup naredbi i koje imaju dobro definiran ulaz i izlaz
- Rješavamo se problema ponavljanja programskog kôda
 - Definiramo ga na jednom mjestu i zatim samo pozivamo funkciju kad nam zatreba takva funkcionalnost





Funkcije / procedure

- Primjeri:
 - Matematičke funkcije – `sin`, `cos`
 - Funkcije za U/I – `printf`, `fwrite`
 - Funkcije za rad s grafikom - `DrawLine`,
`DrawCircle`
- Većina razvojnih okruženja dolazi s ugrađenim **mnoštvom** biblioteka takvih funkcija

Reusability – mogućnost ponovnog korištenja programskog kôda

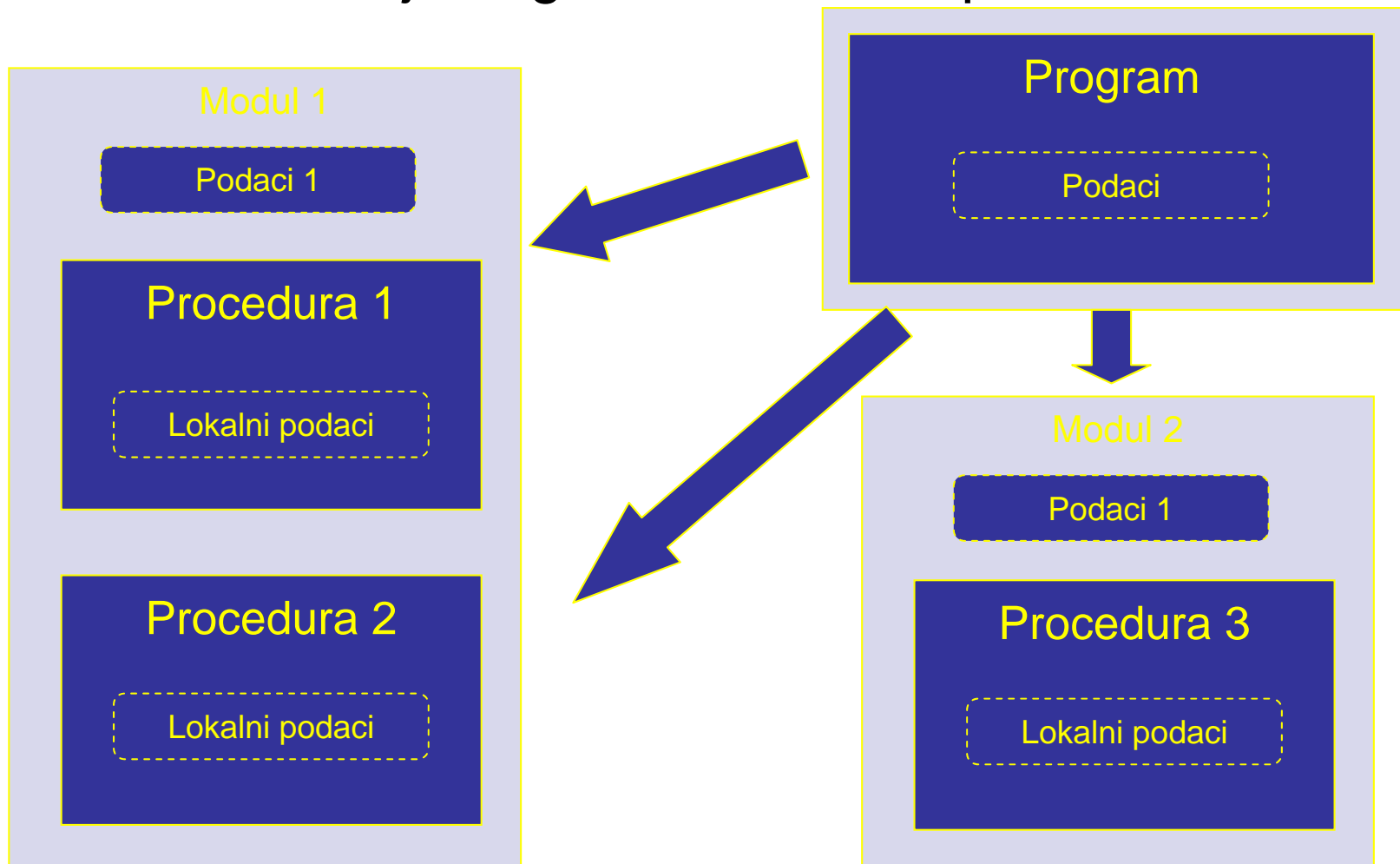


- “*write once, use everywhere*”
- Ali, nije sve tako jednostavno
- Javlja se problem **organizacije** velikog broja funkcija u smislene cjeline



Modularno programiranje

- Procedure srodne funkcionalnosti grupiraju se u module koji mogu imati vlastite podatke



Modul kao dobro definirana i zatvorena funkcionalna jedinica



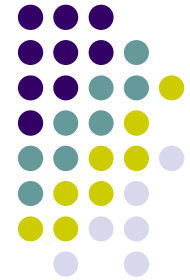
- Programski jezici imaju vrlo “tanku” (i često nejasnu) definiciju modula
 - Npr u C-u modul može biti:
 - .h datoteka s pripadajućom .cpp datotekom
 - biblioteka funkcija koja se statički povezuje (*linka*) u naš program
 - Windows DLL biblioteka
- Koncept modula je najčešće implementacijski detalj programskog okruženja u kojem radimo i nije potpuno integriran u sam programski jezik

Modul kao dobro definirana i zatvorena funkcionalna jedinica



- Osnovni problem je što funkcije i podaci ugrađeni u modul nisu povezani u konkretne programske entitete koji bi predstavljali razumljiv i jasan koncept za krajnje korisnike modula !
- Standardni `<stdio.h>` sadrži niz struktura i funkcija a mi bismo željeli imati “nešto” kao: `File`, `BinaryFile`, `TextFile`, ...

Korištenje modula



- Moduli “niske razine” (npr. U/I funkcije u C-u, funkcije za grafičko iscrtavanje u Windows-ima) se svakodnevno (i uspješno) iskorištavaju u projektima diljem svijeta. Ovakvi moduli su jasno definirani, s dobrom dokumentacijom i “izglačani” dugotrajnim korištenjem



Stvarna razlika od asemblera?

- MASM (Microsoft Macro Assembler) ima varijable, mogućnost definiranja potprograma i mogućnost kreiranja modula
- C - jednostavnija sintaksa i moćniji skup osnovnih naredbi
- Traži se potpuno nova **paradigma** programiranja
- “Odgovor” je **objektno-orijentirana paradigma**