

# SPLETNO PROGRAMIRANJE

## pisni izpit, 13. 9. 2013

*Vsaka naloga je vredna 25 točk. Čas pisanja: 75 minut. Literatura je dovoljena.*

*Vsako nalogo rešujte na svojo stran! Programe in izpise komentirajte!*

Ogled izpitov bo v petek, 13.9.2013 ob 13h v LKM.

Ustni izpiti bodo v petek, 13.9.2013 ob 13.30h v kabinetu profesorja (4. nad.).

### **1. Spletne tehnologije**

*Odgovarjajte kratko!*

- a.) Kaj je imenski prostor pri dokumentih XML? Kdaj nujno potrebujemo njegovo opredelitev?
- b.) Kaj je namen arhitekture MVC, kakšne so prednosti in komu koristijo?
- c.) Kaj je to hranjenje podatkov o seji? Kako ga izvajata ASP.NET in JSP/JSF?

### **2. ASP.NET**

V tehnologiji ASP.NET napiši aplikacijo, ki ustreza naslednjim zahtevam:

- a.) ob prvem obisku naj aplikacija prikaže obrazec, v katerem so: vnosno polje, labela (besedilno polje, ki je ob začetku prazno) in potrditveni gumb (vse tri kontrole so strežniške). V vnosno polje lahko uporabnik vnese neko celoštevilsko vrednost  $n$ . Po kliku na gumb naj se proži izpis (točka b).
- b.) Implementirajte validator na strežniški strani, ki zahteva vnos smiselne celoštevilске vrednosti v vnosno polje.
- c.) Ob oddaji obrazca iz alinee a naj aplikacija v labelo izpiše vrednosti od 1 do  $n$  (eno v vsaki vrstici) in njihove kvadrate.

Primer za vnešeni  $n=3$ :

1: 1  
2: 4  
3: 9

### **3. PHP**

Napiši aplikacijo v tehnologiji PHP, ki bo prikazovala nakupovalni seznam za trgovino. Implementiraj naslednje:

- a.) Predpostavi, da je nakupovalni seznam podan v piškotku in sicer v sintaksi: `živilo1/količina1;živilo2/količina2;...;živilon/količinan` (npr. kruh/3;mleko/2 itd.). Na vrhu strani izpiši seznam v obliki tabele (`<table>`).
- b.) Na dnu strani naj vsebuje aplikacija gumb in razdelek `<div>`, ki bo prikazoval status. Prikazani status je odvisen od skupnega števila izdelkov. Če je število izdelkov (vsota vseh količin) manjša ali enaka 10 ali če je na seznamu živilo "kruh", je status: "Pojdi v trgovino!". Sicer je status enak: " $N$  izdelkov je pretežkih za nošenje!" (kjer je  $n$  izračunani parameter).

**OBRNI !**

## **4. Ruby on Rails**

V tehnologiji Ruby on Rails izdelaj naslednjo rešitev:

- a) V programskem jeziku Ruby napišite bločno metodo *oblikuj*, ki deluje nad asociativnim poljem oblike {"nazivi" => [...], "definicije" =>[...]}. Glede na podane parametre metoda izpiše definicijski seznam HTML (torej uporabi značke <dl>, <dt>, <dd> itd.). Pojmi definicijskega seznama so shranjeni v parametru *nazivi*, definicije pa v parametru *definicije*. V vsaki definicijski postavki se podani bločni ukaz izvede tolikokrat, kolikor je zaporedna številka definicije. Primer:

```
> {"nazivi" => ["zajtrk", "kosilo", "večerja"], "definicije" = ["jutranji obrok", "glavni obrok dneva", "zadnji obrok"]}.oblikuj {print "*"}
```

```
zajtrk
  * jutranji obrok
kosilo
  ** glavni obrok dneva
večerja
  *** zadnji obrok
```

- b) V Rails napiši izsek pogleda (view), ki z uporabo zgornje bločne metode oblikuje izpis poljubnega dela kode v zapisu HTML.
- c) Zapiši poti in imena datotek, v katere bi shranili zgornjo metodo in zgornji pogled v skladu s principom MVC, implementiranim v okolju Rails.