

Objektno orijentirano programiranje - praktikum u .NET okolini

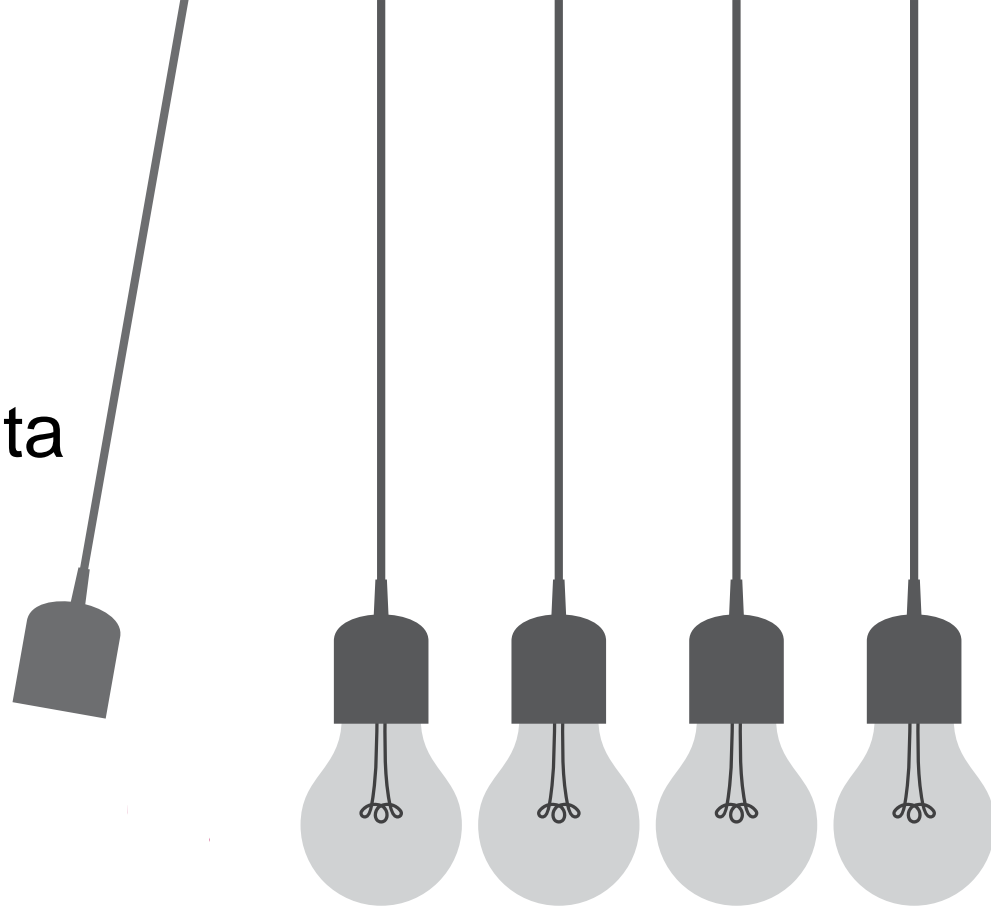
Predavanje 01

Organizacija predavanja i vježbi

Nositelj kolegija:	Đani Pašić	dani.pasic@algebra.hr
	Silvije Davila	silvije.davila@algebra.hr
Izvođenje nastave:	Predavanja	1 sat tjedno - svaki neparni tjedan utorkom - ukupno 15 sati
	Vježbe	3 sata tjedno - po grupama prema rasporedu - ukupno 45 sati

Informacije o kolegiju

- 5 ECTS bodova = 150 sati rada studenta
 - 15 sati predavanja
 - 45 sati vježbi
 - 90 sati rada kod kuće
- Obvezni kolegij



Cilj kolegija

Student će se upoznati s načinima i tehnologijama programiranja Windows aplikacija u .NET objektno-orijentiranoj okolini.

Skupovi ishoda učenja

Oznaka skupa ishoda učenja	Naziv skupa ishoda učenja	ECTS bodova
S1	Osnove izrade desktop aplikacija	3
S2	Osnove izrade bogatog grafičkog sučelja u desktop aplikacijama	2

Ishodi učenja

Skup	Ishod	MINIMALNI ISHODI UČENJA (po uspješnom završetku kolegija, student će moći)	ŽELJENI ISHODI UČENJA (uspješan student bi trebao moći)
S1	I1	Konstruirati desktop rješenje koristeći standardne sastavnice	Konstruirati desktop rješenje koristeći napredne sastavnice
	I2	Preporučiti načine poboljšanja korisničkog iskustva u desktop aplikaciji te ih implementirati	Osigurati pohranu podataka i implementirati korisničko sučelje desktop aplikacije koje će se iscrtavati na temelju pohranjenih podataka
	I3	Preporučiti način realizacije odnosa roditelj-dijete u desktop aplikaciji i implementirati ga	Implementirati naprednu realizaciju odnosa roditelj-dijete u desktop aplikaciji u svrhu poboljšanja korisničkog iskustva
S2	I4	Implementirati prikladnu arhitekturu desktop aplikacije	Implementirati responzivnu desktop aplikaciju uz korištenje prikladne arhitekture
	I5	Primijeniti animacije za postizanje boljeg korisničkog iskustva u desktop aplikaciji	Implementirati dinamičko iscrtavanje korisničkog sučelja desktop aplikacije na temelju dohvaćenih podataka

Tematske cjeline

Tjedan nastave	Cjelina	Tjedan nastave	Cjelina
1. Tjedan	Uvod u kolegij. Uvod u Windows Formse.	9. Tjedan	
2. Tjedan		10. Tjedan	
3. Tjedan	Kontejnerske kontrole. Dijaloške forme.	11. Tjedan	Uvod u Windows Presentation Foundation (WPF).
4. Tjedan		12. Tjedan	
5. Tjedan	Korištenje web servisa iz Windows Forms aplikacija. Asinkroni rad u Windows Forms aplikacijama.	13. Tjedan	Razmještaj elemenata generalno. Razmještaj panela pomoću Panela.
6. Tjedan		14. Tjedan	
7. Tjedan	Korisničke kontrole. Globalizacija i lokalizacija. Kontrole za ispis (engl. <i>print</i>).	15. Tjedan	Resursi, animacije i stilovi. Lokalizacija. Korisničke kontrole.
8. Tjedan			

Literatura

OBAVEZNA LITERATURA

- Beerbohm, M. (2019) Visual C#.NET: Windows Forms Programming with C#. Independently published.
- Yuen, S. (2020) Mastering Windows Presentation Foundation: Build responsive UIs for desktop applications with WPF. 2nd edn. Birmingham: Packt Publishing.

PREPORUČENA LITERATURA

- Nathan, A. (2010) WPF 4: Unleashed. Carmel: Sams.

Za potpis treba?

Za stjecanje prava na potpis potrebno je prisustvovati nastavi u postotku propisanom Pravilnikom o studijima i studiranju

Dolaznost na predavanja i vježbe	
najmanje 50% fizičke prisutnosti na predavanjima	najmanje 60% fizičke prisutnosti na vježbama

Tko ne dobije potpis, mora sljedeće godine ponovno upisati kolegij, platiti upis kolegija te nema pravo polaganja ispita.

Polaganje kolegija



- Kolegij ima definiranih 5 ishoda učenja koji su raspoređeni u 2 skupa ishoda učenja.
- **Da bi student položio kolegij mora po svakom ishodu učenja ostvariti minimalno 50% bodova raspoloživih za taj ishod učenja.**

Kako je to raspoređeno po ishodima učenja

Skup	Ishod	MAX (Projektni zadatak)
S1	I1	20
	I2	20
	I3	20
S2	I4	20
	I5	20
	Ukupno	100

Ocjenjivanje

Broj osvojenih bodova	Ocjena
0,00 – 50,00	1 (nedovoljan)
50,01 – 58,00	2 (dovoljan)
58,01 – 75,00	3 (dobar)
75,01 – 92,00	4 (vrlo dobar)
92,01 – 100,00	5 (izvrstan)

Ispiti

- Na svakom kolegiju vrijedi **pravilo 3 + 1**
 - To znači da student mora položiti ispit iz najviše 4 izlaska
 - 3 redovna izlaska – Uključena u cijenu školarine
 - 1 izvanredni izlazak – Odlukom o naknadi troškova 4. prijava ispita se plaća
 - Vremenski rok za položiti ispit je **12 mjeseci** od dana upisa kolegija
 - Ako student u 12 mjeseci ne položi kolegij, **mora ponovno upisati kolegij te ponovno polagati sve skupove ishoda učenja kako je definirano kolegijem**
- **Vodite računa o rokovima prijave i odjave ispita na IE.**
 - Ako niste prijavili ispit na vrijeme, ne možete pristupiti obrani projektnog zadatka
 - Ako je student prijavio više ispitnih rokova iz istog kolegija, pri dobivanju ocjene kojom je zadovoljan, dužan je odjaviti svaki sljedeći rok koji je iz tog kolegija prijavio. U suprotnom, studentu se u Infoeduku unosi nedovoljan (1).

Projektni zadatak...

- Na Infoeduci pod Nastavni materijali -> OOP - praktikum u .NET okolini -> Ostali dokumenti kolegija:
 - VUA-OOPNET-Projektni_Zadatak.pdf
 - worldcup-sfg-io.zip

Akademski standard ponašanja

- U komunikaciji (pisanoj i usmenoj) pridržavati se pravila poslovne komunikacije primjerene akademskoj razini.
- Potrebno je držati se jasno definiranih rokova za predaju zadataka (projekt).
 - Svaki projekt poslan nakon definiranog roka neće se ocjenjivati.
- Samo oni studenti koji mogu potvrditi svoje pohađanje, smatrat će se prisutnima.
 - Potpisivanje drugih studenata ili registracija njihovom karticom nije dopušteno i može biti predmet stegovnog postupka. Nastavnik će obrisati prisustvo ako utvrdi da je student prijavljen, a da nije prisutan na nastavi.

Pravila ponašanja na nastavi – fizička prisutnost

- Na nastavu se dolazi na vrijeme.
- Pri ulasku u učionicu student prilazi do stola i prijavljuje se na nastavu karticom te sjeda na dostupno mjesto za rad.
- Ometanje nastave i neaktivno sudjelovanje na nastavi nije dozvoljeno.
 - Repetitivno kršenje ovog pravila sankcionira se prijavom Stegovnom povjerenstvu.



Uvod u Windows Formse

Windows Forms

- Skup klasa, kontrola i alata za **razvoj Windows aplikacija**
- Dio .NET programskog okvira
- Naglasak na brzom razvoju (engl. *RAD – rapid application development*)

Pomoćne tehnologije

- Graphic Device Interface + (**GDI+**)
 - **API** za iscrtavanje grafičkog sučelja
- ADO.NET i Entity Framework za pristup podacima

Alternativa Windows Forms aplikacijama

- Windows Presentation Foundation (**WPF**) aplikacije
 - Naglasak na grafičkim mogućnostima
 - Umjesto GDI+ za iscrtavanje koriste **DirectX**

Visual Studio

- Integrirana razvojna platforma
 - Služi za razvoj svih tipova aplikacija
- Sadržava niz korisnih alata:
 - **Vizualne dizajnere** za Windows forme s povuci-pusti (engl. *drag-and-drop*) kontrolama
 - **"Kodno-svjesne" editore** (engl. *code-aware*) koji uključuju kompletiranje naredbi, provjeru sintakse i druge *IntelliSense* mogućnosti
 - **Integrirano prevođenje i otkrivanje grešaka**

Arhitektura rješenja

- Ishodišni element u izradi aplikacija u Visual Studiju naziva se rješenje (engl. *Solution*)
- Svako rješenje sadržava jedan ili više projekata (engl. *project*)
- Pretvaranje projekta u izvršni kôd:
 - **Build -> Build Solution** stvara izvršni kôd
 - **Debug -> Start Debugging** stvara izvršni kôd i pokreće aplikaciju u debug načinu rada
 - **Debug -> Start Without Debugging** stvara izvršni kôd i pokreće aplikaciju

Dijelovi radne okoline

- Radna okolina Visual Studija sastoji se od niza dijelova, od kojih su najbitniji:
 - Središnji dio:
 - Okvir **Design**
 - Okvir **Code**
 - Okvir **Solution Explorer** (Ctrl + Alt + L)
 - Okvir **Properties** (Alt + Enter ili F4)
 - Okvir **Toolbox** (Ctrl + Alt + X)

Forme

- **Forme** (engl. *Forms*) su osnovni građevni blokovi korisničkog sučelja u desktop aplikacijama
- Forma predstavlja **kontejner** koji sadrži **kontrolne**:
 - Omogućuje prezentiranje aplikacije na poznat i konzistentan način
 - Može primiti korisnički unos preko tipkovnice ili miša
 - Može prikazivati podatke u svojim kontrolama
- Jednostavnije aplikacije koriste jednu, a složenije i više formi
- Izgled formi definira se u Design okviru, a instance formi kreiraju se i prikazuju za vrijeme izvršavanja (engl. *run-time*)

Klasa **Form**

- Sve funkcionalnosti formi implementirane su u ugrađenoj klasi **Form**
- Kad dodamo novu formu u projekt, nasljeđujemo tu klasu
 - Klasa je **partial** jer je definirana u dvije datoteke:
 - **.Designer.cs** koristi dizajner i tu mi ne pišemo ništa
 - **.cs** koristimo mi i tu dizajner ne piše ništa

Svojstva Windows formi

- Dostupna su nam brojna svojstva kojima definiramo izgled i ponašanje forme:
 - Pomoću okvira **Properties**
 - U kôdu pomoću **IntelliSense** alata
- Primjerice:
 - Možemo u dizajneru promijeniti naziv forme, podesiti svojstva **Text**, **Size**, **StartPosition**
 - Možemo kroz kôd promijeniti svojstvo **BackColor**
 - Možemo ga podesiti ili na imenovanu boju ili koristiti **Color.FromArgb()**

Rad s kontrolama

- Kontrole dodajemo na formu:
 - Iz **Toolboxa**
 - Kroz kôd
- Forma sve kontrole drži u kolekciji **Controls**
 - Sve što je tamo, forma će iscrtati
- Primjerice, možemo na formu dodati dva gumba i podesiti im **Text, Size, BackColor, Location**:
 - Jedan iz Toolboxa
 - Drugi kroz kôd:
`Button btnZatvori = new Button();`
`this.Controls.Add(btnZatvori);`

Rad s događajima

- Forme i kontrole mogu „okinuti“ niz **događaja**
 - Mi se **pretplaćujemo** na one koji nas zanimaju
 - Definiramo metode koje želimo da budu pozvane kad se događaj dogodi
- Primjerice, pretplata na Click događaj:
 - Kroz okvir Properties (Events)
 - Ili kroz kôd

```
public Form1()
{
    Button btn = new Button
    {
        Text = "Close"
    };
    btn.Click += Btn_Click;
    this.Controls.Add(btn);
}

private void Btn_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```