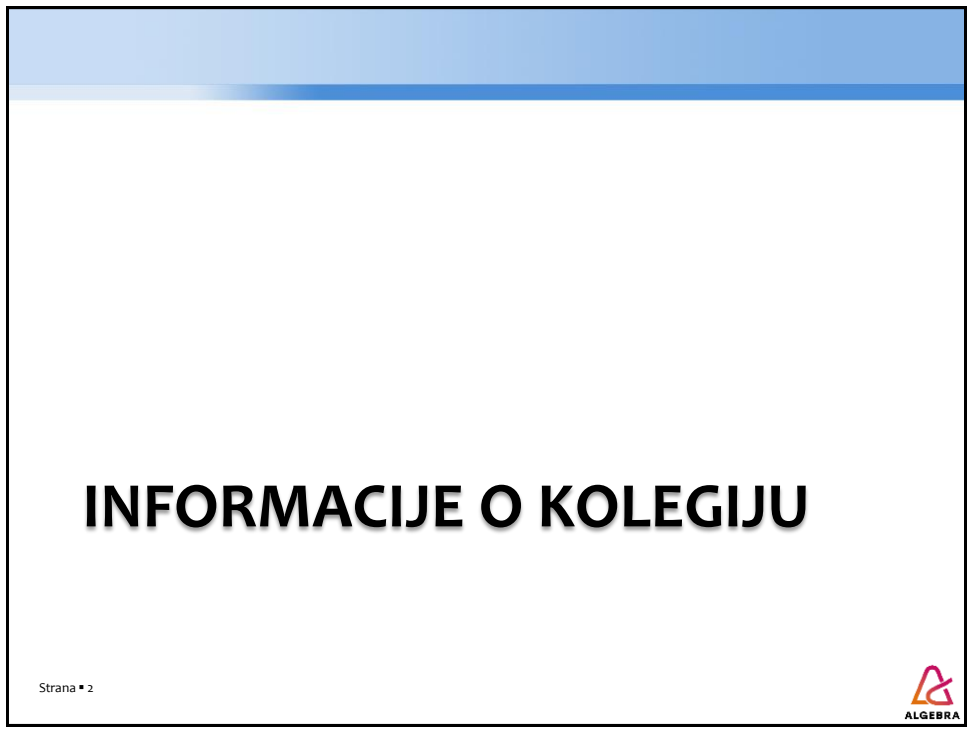

ALGEBRA
PROGRAMIRANJE
Predavanje 01 – Uvod
Uvod

1


INFORMACIJE O KOLEGIJU
Strana ▪ 2
ALGEBRA

2

Nastavnici

- Predavanja:
  - Doc. dr. sc. Goran Đambić
  - goran.djambic@algebra.hr
  - Konzultacije: bilo kada, uz prethodnu najavu e-mailom
- Vježbe:
  - Andrej Lacković (redovni)
  - Toni Steyskal (izvanredni)

O kolegiju

- Ciljevi kolegija:
  - Naučiti se izražavati pomoću algoritama
  - Naučiti temeljne koncepte programiranja
  - Naučiti izraditi jednostavne programe u zadanom programskom jeziku
- Kolegij nosi **6 ECTS** bodova (otprilike 30 sati / bod)
  - 30 sati predavanja (15 tjedana po 2 sata)
  - 45 sati vježbi (15 tjedana po 3 sata)
  - 105 sati samostalnog rada (15 tjedana po 7 sati)

Literatura

- Downey, A.: Think Python: How to Think Like a Computer Scientist
  - https://greenteapress.com/wp/think-python-2e/
  - (HTML) https://greenteapress.com/thinkpython2/html/index.html
  - (PDF) https://greenteapress.com/thinkpython2/thinkpython2.pdf

ALGEBRA

5

Potpis

- 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

ALGEBRA

6

Prikupljanje bodova i ocjene

- Na kolegiju je moguće skupiti najviše **100 bodova:**
  - Školske zadaće: najviše **10 bodova**
  - 2 međuispita: najviše **90 bodova**
  - Usmenog ispita nema

$$\sum = 100$$

- Ocjene:
  - 92,01 – 100,00 bodova: izvrstan (5)
  - 75,01 – 92,00 bodova: vrlo dobar (4)
  - 58,01 – 75,00 bodova: dobar (3)
  - 50,01 – 58,00 bodova: dovoljan (2) ← Uz barem 50% bodova na svakom ishodu

ALGEBRA

7

Ishodi učenja

Skup ishoda učenja	Ishod učenja	MINIMALNI ISHODI UČENJA (po uspješnom završetku kolegija, student će moći)	ŽELJENI ISHODI UČENJA (uspješan student bi trebao moći)
S1	I1	Osmisliti jednostavan algoritam prema smjernicama te ga implementirati koristeći osnovne elemente programskog jezika.	Osmisliti jednostavan algoritam prema manje strukturiranim smjernicama te ga implementirati koristeći osnovne elemente programskog jezika.
	I2	Odabrati prikladan kontejner za čuvanje više podataka te primijeniti složene matematičke i logičke operacije na njegove elemente.	Odabrati optimalan kontejner za čuvanje više podataka te primijeniti složenije matematičke i logičke operacije na njegove elemente.
	I3	Osmisliti rješenje jednostavnog problema koristeći funkcije te ih kreirati i upotrijebiti.	Osmisliti rješenje složenijeg problema koristeći funkcije te ih kreirati i upotrijebiti.
S2	I4	Dizajnirati jednostavan korisnički tip podataka te koristiti njegove instance za rješavanje problema.	Dizajnirati jednostavan korisnički tip podataka te koristiti njegove instance za rješavanje problema koristeći složenije programske konstrukte.
	I5	Kreirati rješenje korištenjem raspoloživih memorijskih opcija stoga i hrpe.	Kreirati rješenje složenijeg problema korištenjem raspoloživih memorijskih opcija stoga i hrpe.
	I6	Konstruirati rješenje korištenjem datoteka.	Konstruirati rješenje složenijeg problema korištenjem datoteka.

ALGEBRA

8

Ishodi učenja i provjere znanja

	Ispit	Školska zadaća	Ukupno
Ishod učenja 1			
Minimalni bodovi	11	2	13
Željeni bodovi	5		5
Ishod učenja 2			
Minimalni bodovi	11	2	13
Željeni bodovi	5		5
Ishod učenja 3			
Minimalni bodovi	11	2	13
Željeni bodovi	5		5
Ishod učenja 4			
Minimalni bodovi	11	2	13
Željeni bodovi	5		5
Ishod učenja 5			
Minimalni bodovi	11	2	13
Željeni bodovi	5		5
Ishod učenja 6			
Minimalni bodovi	7		7
Željeni bodovi	3		3
UKUPNO:	**90**	**10**	**100**

ALGEBRA

9

Ispiti

- Na svakom kolegiju vrijedi **pravilo 3 + 1**
  - To znači da student mora položiti ispit iz najviše 4 izlaska
    - 3 redovna – Uključena u cijenu školarine
    - 1 izvanredni – Odlukom o naknadi troškova 4. prijava ispita se naplaćuje
  - 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 ni pismenom niti usmenom dijelu
  - 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).

ALGEBRA

10

Ispiti

- Međuispit 1 će sadržavati zadatke za ishode I1, I2, I3
- Međuispit 2 će sadržavati zadatke za ishode I4, I5, I6
  - Dodatno, studenti će moći popravljati bilo koji ishod učenja s međuispita 1
- Na bilo kojem sljedećem roku, studenti će moći polagati bilo koju kombinaciju ishoda

ALGEBRA

11

Školske zadaće

- Pišu se na nekim vježbama
  - Asistent će dati više detalja
- Donose određen broj bodova
- Ne mogu se naknadno ponavljati
  - Uz dogovor s asistentom moguće je doći u drugi termin istoga tjedna
- Pratiti upute asistenata

ALGEBRA

12

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 (zadaća, seminarskih radova, projekata i sl.)
  - Svaki zadatak, domaća zadaća, projekt itd., poslani 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

ALGEBRA

13

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

ALGEBRA

14

PROGRAMIRANJE

ALGEBRA

15

Što je programiranje?

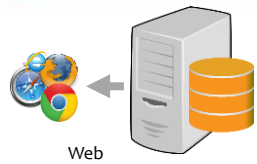
- Programiranje je proces izrade računalnog programa te njegovog testiranja i uklanjanja grešaka
  - Rezultat programiranja je program sastavljen od skupa **podataka** i skupa **naredbi** (instrukcija)
- Tipični koraci kod programiranja:
  1. Napiši programski kôd koji rješava mali dio funkcionalnosti
  2. Pokreni i provjeri kako radi. Ako postoje pogreške, popravi.
  3. Ponavljaj prethodne korake za preostale funkcionalnosti
  4. Program je gotov i spreman za korištenje

ALGEBRA

16

Što možemo programirati?

Desktop aplikacije



Web aplikacije

IoT/AI uređaje

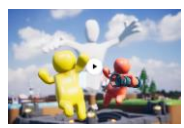
Konzolne aplikacije

Servisne (daemon) aplikacije

Roboti

Autonomna vozila

Mobilne aplikacije



Računalne igre

Slike preuzete s: nasa.gov, amazon.com, hansonrobotics.com, floridapolitics.com

ALGEBRA

17

Programski jezik

- Umjetni jezik namijenjen za izražavanje računanja koje neko računalo može izvršiti (1936, Turing i Church)

Sep 2023	Sep 2022	Change	Programming Language	Ratings	Change
1	1		Python	14.16%	-1.58%
2	2		C	11.27%	-2.70%
3	4	▲	C++	10.65%	+0.90%
4	3	▼	Java	9.49%	-2.23%
5	5		C#	7.31%	+2.42%
6	7	▲	JavaScript	3.30%	+0.48%
7	6	▼	Visual Basic	2.22%	-2.18%
8	10	▲	PHP	1.55%	-0.13%
9	8	▼	Assembly language	1.53%	-0.96%
10	9	▼	SQL	1.44%	-0.57%

TIOBE indeks za rujan 2023; *The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the best programming language or the language in which most lines of code have been written.*

ALGEBRA

18

Prva generacija programskih jezika

- **1GL** - jezici prve generacije
  - Instrukcije i podaci se unose u strojnom jeziku kao nizovi nula i jedinica
  - Primjer izračuna *n*-tog Fibonaccijevog broja (na 32-bitnim x86 procesorima)
    - Zbog čitljivosti prikazano heksadecimalnim zapisom

```
8B542408 83FA0077 06B80000 0000C383 FA027706
B8010000 00C353BB 01000000 B9010000 008D0419
83FA0376 078BD98B C84AEBF1 5BC3
```


Druga generacija programskih jezika

- **2GL** - jezici druge generacije
  - Uvode *mnemonike* (naredbe) koji su čovjeku razumljiviji od nula i jedinica
  - Primjer: "Hello, world!" program za ZX Spectrum

```
        ORG #8000     ; Start address of the routine
START   LD A,2        ; set the output channel
        CALL #1601    ; to channel 2 (main part of TV display)
        LD HL,MSG     ; Set HL register pair to address of the message
LOOP    LD A,(HL)     ; De-reference HL and store in A
        CP 0          ; Null terminator?
        RET Z         ; If so, return
        RST #10       ; Print the character in A
        INC HL        ; HL points at the next char to be printed
        JR LOOP
MSG     DEFM "Hello, world!"
        DEFB 13       ; carriage return
        DEFB 0        ; null terminator
```


Treća generacija programskih jezika

- **3GL** - jezici **treće** generacije
  - Dodatni nivo **apstrakcije**, potreban ***prevoditelj***
  - Primjer: "Hello, World!" za Delphi

```
{$APPTYPE CONSOLE}
begin
   Writeln('Hello, world!');
end.
```


ALGORITMI I PROGRAMI

Algoritmi i programi

- Algoritam je niz dobro definiranih koraka za rješavanje problema
- Da bismo mogli napisati program, prvo moramo osmisliti algoritam
  - Algoritam je neovisan o programskom jeziku
  - Program je implementacija algoritma

ALGEBRA

23

Algoritmi i programi

- Primjerice, što je rezultat rada sljedećeg algoritma:
  1. Postavi `max` na 0.
  2. Svaki broj `x` u listi `L` usporedi s `max`. Ako je `x` veći, onda postavi `max` jednak `x`.
  3. Ispiši `max`.
- Algoritme zapisujemo u pseudokôdu

ALGEBRA

24

Algoritmi i programi

- Prethodni algoritam možemo implementirati u bilo kojem programskom jeziku:

```python
L = [ 6, 2, 9, 1, 5]
max = 0

for x in L:
    if x > max:
        max = x

print(max)
```

Python

```cpp
int L[] = { 6, 2, 9, 1, 5 };
int max = 0;

for (int x : L) {
   if (x > max) {
      max = x;
   }
}

cout << max << endl;
```

C++

```perl
my @L = (6, 2, 9, 1, 5);
my $max = 0;

foreach my $x (@L) {
   if ($x > $max) {
      $max = $x
   }
}

print $max;
```

Perl

ALGEBRA

25

UVOD U PYTHON

ALGEBRA

26

Kratka povijest

- Otac programskog jezika Python je Guido van Rossum
- Prva verzija je objavljena 1991. godine na temelju jezika ABC
- Često se opisuje izrazom „batteries included”
  - Što god želite raditi, vjerojatno već imate gotovu biblioteku
- Vrlo popularan za skriptiranje i podatkovnu znanost
- Verzije:
  - Python 2.0 objavljen 2000. godine
    - Od 2020. se ne razvija više (*discontinued*), zadnja verzija je 2.7.18
  - Python 3.0, objavljen 2008. godine
    - Trenutna verzija je 3.11.5

ALGEBRA

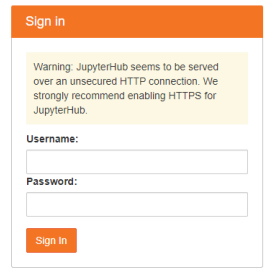
Kako početi

- Koristit ćemo programsko okruženje Jupyter Notebook
  - Alternative su PyCharm, IDLE, Atom, Visual Studio Code …
- U njemu je integrirano:
  - Editor za pisanje programskog kôda
  - Interpreter
    - Programeri pišu izvorni kôd (engl. *source code*), operacijski sustav razumije izvršni kôd (engl. *executable code*)
    - Interpreter u hodu prevodi izvorni kôd u izvršni kôd
  - Dokumentiranje tekstom i slikama

ALGEBRA

Algebrin Jupyter Notebook

- Algebrin Jupyter Notebook je dostupan:
  - S vanjske mreže: http://vss01.vua.cloud:8888
  - Iz Algebrine mreže: http://pythonr.vua.cloud:8888
  - Ako zagusti: http://193.198.186.130:8888/hub/login



- Korisničko ime je ono od Infoeduke, a lozinka Lozinka1234
  - Na prvim vježbama se logirate i mijenjate lozinku

ALGEBRA

Rad s *notebookovima*

- Otvorimo /Programiranje/Predavanja/Primjer.ipynb

- Svaki *notebook* se sastoji od više ćelija (engl. *cell*)
  - Trenutni *notebook* sadrži jednu ćeliju
  - Lijevi klik aktivira ćeliju (aktivna ćelija je obrubljena)
  - Ćelija može sadržavati nešto od navedenog:
    - *Markdown* (tekst, slike, linkovi …)
    - Programski kôd

ALGEBRA

Osnove radnje: dodavanje *markdown* ćelije

- U Jupyter Notebooku tekst uređujemo (formatiramo) koristeći opisni jezik Markdown
- Postupak je sljedeći:
  - Kreirajmo novi *notebook* zadavanjem *New* → *Python 3*
  - Kao naziv upišimo *Test*
  - Iz izbornika *Cell* odaberemo *Cell Type* i zadamo *Markdown*
  - Upišemo:

```
## Hello, world!
Ovo je moja prva ćelija.
<img
src="https://i.pinimg.com/originals/8f/05/c6/8f05c62be67c
f8d0f6494c460e7ec0dd.gif" />
```

  - Zadamo naredbu Run (ili Control+Enter)

Osnove radnje: dodavanje ćelije s kôdom

- Postupak je sljedeći:
  - Aktiviramo zadnju ćeliju
  - Iz izbornika *Insert* zadamo *Insert Cells Below*
  - Iz izbornika *Cell* odaberemo *Cell Type* i zadamo *Code*
  - Upišemo:

```
tekst1 = "Hello"
tekst2 = "world"
zajedno = tekst1 + tekst2
print(zajdeno)
```

  - Zadamo naredbu Run (ili Control+Enter)

Sintaksne pogreške

- Sintaksne pogreške otkriva interpereter i na njih nas upozorava

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-90f17a66f08b> in <module>
      2 tekst2 = "world"
      3 zajedno = tekst1 + tekst2
----> 4 print(zajdeno)

NameError: name 'zajdeno' is not defined
```

- Da bismo mogli izvršiti program, moramo ispraviti sve pogreške ovog tipa
  - Preporuka je ispravljati greške od prve prema zadnjoj
  - Često kad ispravimo prvu ostale nestanu

Drugi pokušaj

- Pokušajmo ispraviti pogrešku:

```
tekst1 = "Hello"
tekst2 = "world"
zajedno = tekst1 + tekst2
print(zajedno)
```

- Ponovnim pokretanjem ćelije program se izvršava
  - Popravili smo sve sintaksne pogreške
- Što ipak nije u redu s našim programom i zašto nas interpreter nije upozorio?

Bugovi

- Pogreške pri izvođenju (engl. *run-time errors*) se još nazivaju i ***bugovi***
  - *Bugovi* su logičke pogreške
- Interpreter nas ne može upozoriti da smo napravili *bug*
  - Primjerice, željeli smo napisati program koji oduzima dva broja, ali smo greškom napisali „+" umjesto „-"
    - Što se tiče interpretera, program je ispravan – to što ne radi ono što bismo mi htjeli je naš problem, a ne interpreterov ☺
- *Bugovi* su očekivani i pojavljuju se u svakom programu
  - Proces *debuggiranja* služi uklanjanju *bugova*

Treći pokušaj – uspjeh

- Ispravljanjem *buga* dobivamo finalni program koji ispravno radi:

```
tekst1 = "Hello"
tekst2 = "world"
zajedno = tekst1 + " " + tekst2
print(zajedno)
```


Tipovi podataka i varijable

- **Tip podataka** je skup vrijednosti i operacija nad njima
- Za svaki tip podataka točno se zna **koliko će prostora zauzimati** u memoriji varijable tog tipa
  - Prostor se mjeri u bajtovima

ALGEBRA

37

Primjeri

- Kakav je podatak vaše prezime?
  - Tekstualni
- Kakav je podatak vaš datum rođenja?
  - Datumski
- Kakav je podatak trenutna godina?
  - Brojčani, cjelobrojni
- Kakav je podatak površina predavaonice?
  - Brojčani, decimalni

ALGEBRA

38

TODO lista do sljedećeg predavanja

- Spojiti se na Algebrin Jupyter Notebook
- Pročitati prezentaciju s predavanja i isprobati sve primjere

ALGEBRA