

Podrška objektno orijentisanom programiranju u jezicima C++, Objective C, Java, C#, Ada i Ruby

Katarina Popović, Dušan Pantelić, Dejan Bokić, Nikola
Stojević

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Maj 2019

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Sadržaj

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Uvod

- ▶ Programska paradigma
- ▶ Princip jedinstvene odgovornosti
- ▶ Enkapsulacija
- ▶ Nasleđivanje
- ▶ Polimorfizam
- ▶ Apstrakcija

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

C++

- ▶ C++ je delimično objektno orijentisan jezik
 - ▶ Main funkcija izvan klase
 - ▶ Koncept globalne promenljive
 - ▶ Postojanje friend funkcija
- ▶ Enkapsulacija u C++
 - ▶ public, protected i private sekcije
- ▶ Nasleđivanje u C++
 - ▶ public, protected i private nasleđivanje
 - ▶ virtuelno nasleđivanje
- ▶ Polimorfizam u C++
 - ▶ polimorfizam u vreme kompilacije
 - ▶ polimorfizam u vreme izvršavanja
- ▶ Apstrakcija u C++

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Objective C

```
@interface Employee : NSObject {
2   double salary; @public int age;}
@property(nonatomic, readwrite) double salary;
4 - (void)display;
@end # '-' za metode instance, '+' za klasne metode(static)
6 @implementation Employee
@synthesize salary;
8 - (void)display { NSLog(@"Employee salary is %f", salary); }
@end # (id) tip koji je kompaktilan svakom objektu
10 @interface Driver : Employee { NSString* truck; }
- (id)initWithTruck:(NSString*)model;
12 @end # self oznacava tekuci objekat
@implementation Driver
14 - (id)initWithTruck:(NSString*)model {
    truck = model; return self; }
16 - (void)display { NSLog(@"Driver salary is %f", salary); } @end
int main(int argc, const char * argv[]) {
18     NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    Employee *empl = [[Driver alloc] initWithTruck:@"Mercedes"];
20     empl.salary = 5.0;  empl->age = 33;
    [empl display];
22     [pool drain];
    return 0;
}
```

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura



Java - primer koda sa enkapsulacijom, nasledjivanjem, polimorfizmom

```
public class Employee {
    private int salary;
    #this je referenca na tekuci objekat
    public Employee(int salary) { this.salary = salary;}
    public int getSalary(){ return salary;}
    public void setSalary(int newSalary) { salary = newSalary;}
    public void display() {
        System.out.println("Hello i'm employee!");
    }
    public static void main(String[] args) {
        Employee Marko = new Driver(600,"Mercedes");
        Marko.display();}
}

class Driver extends Employee {
    String truck = "FAP";
    #super vrsi poziv konstruktora bazne klase
    public Driver(int salary,String truck) {
        super(salary); this.truck = truck;}
    public void display() {
        System.out.println("My truck is "+truck+"!");
    }
    public void display(String x) {
        System.out.println("My truck is "+truck+x+"!");
    }
}
```

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura



Apstrakcija

Apstraktne klase ili interfjese

```
public abstract class Employee {  
    public abstract void display(); ...  
    interface Employee {  
        public void display(); #podrazumevano apstraktna  
        default void work(){System.out.println("Working"); }  
    }  
}
```

Table: Vidljivost različitih modifikatora pristupa.

Modifikator	Klasa	Paket	Podklasa	Svet
public	Da	Da	Da	Da
protected	Da	Da	Da	Ne
podrazumevani	Da	Da	Ne	Ne
private	Da	Ne	Ne	Ne

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

C#

▶ C#

- ▶ C# je jednostavan, moderan, objektno-orijentisan jezik
- ▶ Nudi punu podršku objektno orijentisanom programiranju
- ▶ Ne podržava druge paradigme ali koristi svoje imperativne strukture

▶ Enkapsulacija u C#

- ▶ public, private, protected, internal i protected internal sekcije

▶ Nasleđivanje u C#

- ▶ može da bude direktno ili indirektno
- ▶ samo jednostruko nasleđivanje je podržano

▶ Polimorfizam u C#

- ▶ polimorfizam vremena kompiliranja
- ▶ polimorfizam vremena izvođenja

▶ Apstrakcija u C#

- ▶ Apstraktni tip se konstruiše dodavanjem abstract u deklaraciji

Ada

- ▶ Ada ne sledi model klase zasnovan na jednom konstruktoru
 - ▶ Odvojene karakteristike tipova unutar paketa
 - ▶ Moguće postojanje funkcija i procedura u paketu
- ▶ Enkapsulacija
 - ▶ Privatnost se određuje na nivou paketa
 - ▶ private, limited private
- ▶ Nasleđivanje
 - ▶ Implementirano nasleđivanje po nivoima i hijerarhijsko
 - ▶ Višestruko nasleđivanje je moguće implementirati
- ▶ Polimorfizam
 - ▶ Pomoću nasleđivanja, apstraktnih tipova i podtipova
- ▶ Apstrakcija
 - ▶ Apstraktni tip se konstruiše dodavanjem abstract u deklaraciji

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Ada - primer koda

```
package Employees is
2   type Employee is tagged
    record
4       Name: String;
    end record;
6   procedure Set_Name(Obj: in out Employee; Name: String);
    function Print(E: Employee) return String;
8 end Employees;

10 package Drivers
    type Driver is new Employees.Employee with record
12     Driver_ID: Integer;
    end record;
14   procedure Set_D(Obj: in out Employee; Name: String);
    overriding function Print(D: Driver) return String;
16 end Drivers;
```

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Ruby

- ▶ Kreiranje klasa u jeziku Ruby
 - ▶ Standardni metod initialize, ponaša se kao konstruktor
- ▶ Enkapsulacija
 - ▶ public, protected i private
 - ▶ attr_accessor(čitanje i izmena), attr_reader(čitanje) i attr_writer(izmena)
- ▶ Nasleđivanje
 - ▶ Implementirano nasleđivanje po nivoima i hijerarhijsko
 - ▶ Višestruko nasleđivanje nije podržano
- ▶ Polimorfizam
 - ▶ Pomoću nasleđivanja
 - ▶ Duck typing
- ▶ Apstrakcija
 - ▶ Nema direktnu podršku
 - ▶ Moguće implementiranje sličnog ponašanja pomoću nasleđivanja

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura

Ruby - primer koda

```
1 class Employee
2   attr_accessor :name
3   def initialize(name)
4     @name = name
5     print()
6   end
7   def print
8     puts "Employee: #{@name}."
9   end
10 end

12 class Driver < Employee
13   def initialize(name)
14     @name = name
15     print()
16   end
17   private
18   def print
19     puts "Driver: #{@name}."
20   end
21 end

22 emp = Employee.new("John")
```

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura



Literatura

- ▶ Introduction to Ada. on-line at:
<https://learn.adacore.com/courses/intro-to-ada/index.html>
- ▶ Object C apple documentation. on-line at: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ObjectiveC>
- ▶ Ruby - Object Oriented. on-line at:
https://www.tutorialspoint.com/ruby/ruby_object_oriented.htm
- ▶ Gary Bennet, Brad Lees and Mltchell Fisher. Objective-C for Absolute Beginners: iPhone, iPad and Mac Programming Made Easy. Apress, Berkely, CA, USA, 3rd edition, 2016
- ▶ AdaCore experts. High-Integrity Object-Oriented Programming in Ada. AdaCore(www.adacore.com), 1.2 release edition, 2011. on-line at:
<http://extranet.eu.adacore.com/articles/HighIntegrityAda.pdf>
- ▶ Hal Fulton. The Ruby Way. Sams Publishing, 2001.
- ▶ Cay S Horstmann. Core Java SE 9 for the Impatient. Addison-Wesley Professional, 2017.
- ▶ Aayushi Johari. Object Oriented Programming - Java OOPs Concepts With Examples, 2018. on-line at: <https://www.edureka.co/blog/object-oriented-programming/>
- ▶ Stephen Prata. C++ Primer Plus (5th Edition) (Primer Plus (Sams)). Sams, Indianapolis, IN, USA, 2004

Podrška objektno
orijentisanom
programiranju u
jezicima C++,
Objective C,
Java, C#, Ada i
Ruby

Katarina
Popović, Dušan
Pantelić, Dejan
Bokić, Nikola
Stojević

Uvod

C++

Objective C

Java

C#

Ada

Ruby

Literatura