Tehnologije i sistemi eUprave

Predavanja 08

Implementacija

- Poglavlje Implementacija prikazuje rešenja zahteva definisanih Specifikacijom zahteva, a na način definisan Specifikacijom dizajna.
- Nije potrebno prikazati rešenje svakog pojedinačnog zahteva, već samo onih koji su relevantni za problem kojim se bavimo.
- Na početku poglavlja treba barem u jednoj rečenici navesti šta se u tom poglavlju nalazi. Na primer: “Ovo poglavlje prikazuje način na koji su implementirane neke od funkcija sistema za …”

Implementacija

- Poglavlje može biti podeljeno na odeljke kako bi pratilo arhitekturu sistema. Pri tome obavezno najaviti ovu podelu. Na primer, “Objašnjena je implementacija … (nabrojati delove implementiranog sistema) ...”.
- Svaki od ovih odeljaka treba započeti najavom šta taj odeljak sadrži. Na primer, “U ovom poglavlju je predstavljena implementacija klijentskog dela aplikacije.”

Implementacija

- Za objašnjenje implementacije se najčešće koriste prikazi fragmenata programskog kôda uz prateća objašnjenja.
- Svaki fragment (listing) mora biti spomenut u tekstu uz kratko objašnjenje šta se tim kôdom postiže i na koji način.
- Ispod prikaza listinga se mora nalaziti numeracija i njegov naziv.
- Listing mora predstavljati zaokruženu celinu, na primer, celu klasu, jednu metodu, jednu for petlju, blok nekoliko naredbi i slično.
- Pošto je listing izvučen iz nekog konteksta potrebno je uputiti čitaoca šta predstavlja sve ono što u tom listingu nije definisano. Na primer, neke konstante, globalno definisane varijable i slično.

Implementacija

- Programski kôd prikazan na listingu mora biti relevantan za problem koji se rešava.
- Osim toga, programski kôd mora biti reprezentativan.
- Listing ne sme sadržati greške (bug-ove).
- Nazivi klasa, metoda, atributa, itd. koji se pojavljuju na listingu trebaju biti opisni i smisleni.
- Listing treba biti koncizan, optimalan, bez nepotrebnih praznih linija.
- Pri skaliranju listinga voditi računa o čitljivosti jer bi veličina fonta na listingu trebala biti približna veličini fonta u ostatku teksta.

Primer navođenja listinga

Za određivanje broja pozitivnih ocena nekog učenika je implementirana metoda prikazana na listingu 1.

```
public static int brojPozitivnihOcena(Ucenik ucenik) {
        int brojac = 0;
        for (Integer ocena: ucenik.getOcene())
                if (ocena > 1)
                        brojac++;
        return brojac;
}
```

Listing 1 - Određivanje broja pozitivnih ocena učenika

Ovoj metodi se prosleđuje referenca na objekat koji reprezentuje konkretnog učenika. Iteracijama nad kolekcijom svih ocena tog učenika se utvrđuje broj onih ocena koje su veće od 1. Ukupan broj takvih ocena predstavlja povratnu vrednost ove metode.

Primeri loše implementacije

```
public int brojPozitivnihOcena(Ucenik ucenik) {
    int brojac = 0;
    for (Integer ocena: ucenik.getOcene())
        if (ocena > 1)
            brojac++;
```

```
public boolean noviStudent(String ime, String prezime, String brIndeksa) {
    Student s = new Student(ime, prezime, brIndeksa);
    boolean success = StudentDAO.save(s);
    if (success == true)
        return true;
    else
        return false;
}
```

```
public boolean areListsEqual(List list1, List list2) {
    boolean result = false;
    if (list1.size() == list2.size()) {
        for (int i=0; i<list1.size(); i++) {
            Object obj1 = list1.get(i);
            Object obj2 = list2.get(i);
            if (obj1.equals(obj2))
                result = true;
            else
                result = false;
        }
    }
    return result;
}
```

```
public void save(String filename, char[] data) throws IOException {
    FileOutputStream fout = new FileOutputStream(filename, true);
    for (int i=0; i<data.length; i++) {
        fout.write(ch[i]);
    }
}
```

```
public BufferedImage bytes2Image(byte[] bytes) {
    String image = Base64.encodeBase64String(bytes);
    InputStream is = new ByteArrayInputStream(Base64.decodeBase64(image));
    return ImageIO.read(is);
}
```

Demonstracija

- Poglavlje Demonstracija ilustruje korišćenje softverskog rešenja.
- Ovo poglavlje je nalik korisničkom uputstvu.
- Potrebno je prikazati nekoliko ključnih scenarija u korišćenju softvera kojima se rešava problem kojim ste se bavili.
- Scenario treba da prati tok korišćenja softvera, odnosno da redom prikazuje i objašnjava korake tokom rada sa softverom.
- Objašnjenja pisati u trećem licu ili eventualno u pasivu.
- U poglavlju Demonstracija ne treba spominjati tehnologije, softverske biblioteke, niti bilo kakve implementacione detalje (tome je mesto u poglavlju Implementacija).

Demonstracija

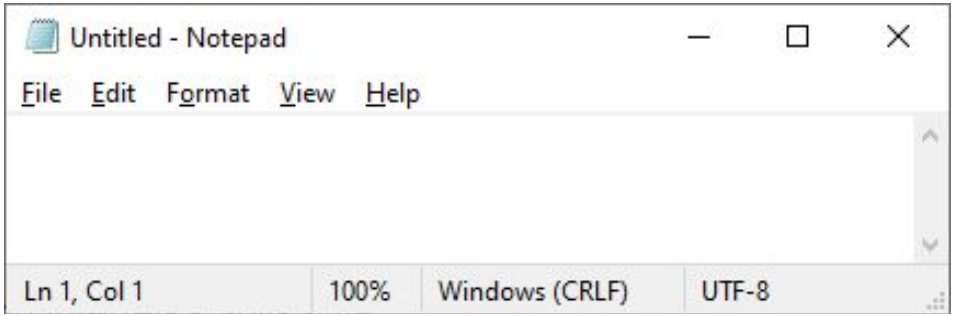
- Na početku poglavlja treba u barem jednoj rečenici spomenuti šta to poglavlje sadrži. Na primer, “Ovo poglavlje prikazuje način korišćenja aplikacije za …”.
- Zatim se čitaocu u koracima objašnjava tipičan način korišćenja softvera.
- Ako sistem sadrži više režima rada ili poseduje klijentske aplikacije za različite platforme, mogu se uvesti odeljci u ovom poglavlju. Prvo je potrebno najaviti podelu (na primer, “Pristup sistemu je moguć putem mobilne i veb aplikacije, što je objašnjeno u narednim odeljcima), a potom je na početku svakog odeljka potrebno naglasiti šta taj odeljak prikazuje (na primer, “Ovaj odeljak prikazuje način korišćenja sistema putem mobilne aplikacije.”)

Demonstracija

- Za demonstraciju rada sistema od pomoći mogu biti screenshot-ovi korisničkog interfejsa, koje je potrebno spomenuti u tekstu i ispod svakog screenshot-a navesti redni broj slike i njen naziv.
- Ako softver ne poseduje GUI, umesto screenshot-ova treba prikazati primere ulaznih i izlaznih podataka.
- U oba slučaja voditi računa o tome da prikazan sadržaj bude tako skaliran da se relevantni detalji mogu pročitati bez potrebe za zumiranjem.

Opis korisničkog interfejsa na primeru Notepad-a

Prilikom pokretanja aplikacije korisniku se prikazuje prozor kao što je ilustrovano na slici 10.



Slika 10 - Izgled korisničkog interfejsa aplikacije

U gornjem delu prozora se nalazi glavni meni, u centralnom delu prozora je polje za unos teksta, a u donjem delu prozora je statusna linija sa osnovnim informacijama, kao što su pozicija kursora, stepen uvećanja prikaza dokumenta, karakteri za prelom reda, kodna stranica dokumenta i slično.

Primeri loše formulisanih rečenica

- Korisnik ulazi u aplikaciju.
- Aplikacija prebacuje korisnika na početnu stranicu.
- Klikom na link aplikacija će odvesti korisnika na formu za unos.
- Moraju se uneti sve tražene vrednosti inače aplikacija neće raditi.
- About us služi da se kaže malo više o aplikaciji.
- Email mora biti u podrazumevanm formatu.
- Korisnike u aplikaciju može unositi samo administrator.
- Ako podaci ne prođu validaciju aplikacija će izbaciti grešku.
- Ako su podaci validni idu na server.

Primeri loše pripremljenih screenshot-ova