

Predmet „Uzorci dizajna“

Ak. god. 2020./2021.

Zadaća 2. Iznajmljivanje električnih vozila

Naziv projekta: {LDAP_korisničko_ime}_zadaca_2

Ishodišni direktorij projekta: {LDAP_korisničko_ime}_zadaca_2

Naziv rješenja: {LDAP_korisničko_ime}_zadaca_2.zip

Prije predavanja projekta potrebno je napraviti Clean na projektu (obrisati sve pomoćne i izvršne datoteke kao što su .class, .jar, .exe, .dll). Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom {LDAP_korisničko_ime}_zadaca_2.zip i predati u Moodle. Uključiti izvorni kod, primjere datoteke(a) podataka, popunjenu datoteku dokumentacije (u ishodišnom direktoriju projekta). Potrebno je podesiti konfiguraciju projekta tako da se kod C# stvara izvršna .exe datoteka (ne .dll i sl.), a kod Java izvršna .jar datoteka.

Uvod.

Tvrtka iznajmljuje različite vrste električnih vozila (bicikl, romobil, motor, automobil i sl.). Tvrtka za svaku vrstu vozila ima određeni broj primjeraka što čini vozni park. Svaka vrsta vozila ima svoje osobine: vrijeme punjenja prazne baterije (0%-100%) u satima, domet u km.

Tvrtka ima više lokacija na kojima se mogu unajmiti i vraćati vozila. Svaka lokacija ima ograničeni broj mjesta za punjenje pojedine vrste vozila budući da se vozila moraju puniti nakon povratka. Svaka lokacija ima inicijalni broj raspoloživih primjeraka pojedine vrste vozila. Broj raspoloživih primjeraka pojedine vrste vozila ne može biti veći broja mjesta za punjenje te vrste vozila. [Svaki vozilo potrebno je pridružiti jednoznačni cjelobrojni identifikator.](#)

Tvrtka je provela reorganiziranje svog poslovanja. Uvedena je nova organizacijska struktura koja se temelji na organizacijskim jedinicama pri čemu postoji jedna koja je ishodišna (nema nadređenu) i ona predstavlja tvrtku. Jedna organizacijska jedinica može sadržavati druge organizacijske jedinice kao i lokacije. Jedna organizacijska jedinica može biti samo u jednoj organizacijskoj jedinici. Svaka organizacijska jedinica (osim ishodišne) mora biti sastavni dio neke organizacijske jedinice. Jedna lokacija može biti samo u jednoj organizacijskoj jedinici. Svaka lokacija mora biti sastavni dio neke organizacijske jedinice. Svi podaci koji su označeni u datotekama kao id (organizacijske jedinice, lokacije, vozila, itd) su cijeli brojevi.

Korisnik može unajmiti odabranu vrstu vozila samo ako postoji raspoloživo vozilo na odabranoj lokaciji. Između raspoloživih vozila na odabranoj lokaciji bira se ono vozilo koje je imalo manji broj najma. U slučaju istog broja najma kod više vozila najma bira se ono koje ima manji ukupan broj prijeđenih km. U slučaju istog broja prijeđenih km konačan izbor je vozilo s manjim id. Korisnik može imati samo jedan aktivni najam vozila iste vrste vozila. Dok jedno vozilo određene vrste vozila ne vrati korisnik ne može unajmiti drugo vozilo te vrste vozila. Unajmljeno vozilo može se vratiti na odabranu lokaciju ukoliko ima slobodnog mjesta za tu vrstu vozila na lokaciji. [Vozilo koje je vraćeno u neispravnom stanju ne može se dalje iznajmljivati.](#) Korisniku se bilježi broj vraćanja vozila u neispravnom stanju.

Kod vraćanja vozila s najma preuzima se njegov ukupan broj prijeđenih kilometara te se izračunava potrošnja baterije na bazi prijeđenog broja km (tog najma) i dometa vrste vozila u km. Nakon vraćanja vozila treba mu napuniti bateriju. Punjenje baterije provodi se linearno tako da se potrebno vrijeme punjenja baterije vozila izračunava na temelju trenutnog stanja baterije (u %) i vremena punjenja prazne baterije. Vozilo nije moguće unajmiti ako nije puna baterija.

Postoji cjenik za svaku vrstu vozila koji se temelji na najmu, broju sati najma i broju prijeđenih kilometara. Kod vraćanja vozila izračunava se trošak najma (najam, broj započelih sati najma, broj prijeđenih kilometara). [Korisniku se za najam vozila izdaje račun, a on predstavlja zaradu za lokaciju u kojoj je vozilo unajmljeno. Račune izdaje središnji sustav tako da svakom računu dodjeli redni broj. Kod računa ne smije se koristiti objektna veza prema lokaciji \(ili organizacijskoj jedinici\), tako da se bilježi samo id lokacije najma i vraćanja. Kod lokacije \(ili organizacijskoj jedinici\) ne smije se koristiti objektna veza prema računima koji se odnose na lokaciju najma ili vraćanja vozila.](#)

Svaka aktivnost (najam, vraćanje, provjera,...) polazi od virtualnog vremena. Virtualno vrijeme jedne aktivnosti mora obavezno biti veće od virtualnog vremena prethodne aktivnosti.

[U priloženoj datoteci DZ_2_konfiguracija_1.txt nalaze se podaci za interaktivni rad programa.](#)

[U priloženoj datoteci DZ_2_konfiguracija_2.txt nalaze se podaci za skupni rad programa.](#)

U priloženoj datoteci DZ_2_vozila.txt nalazi se popis vrsta vozila i njihove osobine.

U priloženoj datoteci DZ_2_lokacije.txt nalazi se popis lokacija i njihove osobine.

U priloženoj datoteci DZ_2_lokacije_kapaciteti.txt nalazi se broj mjesta za pojedine vrste vozila i broj raspoloživih vrsta vozila po lokaciji.

[U priloženoj datoteci DZ_2_cjenik.txt nalazi se iznos najma, iznos po satu najma i po prijeđenom km po pojedinoj vrsti vozila.](#)

U priloženoj datoteci DZ_2_osobe.txt nalazi se popis osoba s njihovim osobinama.

U priloženoj datoteci DZ_2_aktivnosti.txt nalazi se aktivnosti za skupni način rada. Te aktivnosti mogu se koristiti i kod interaktivnog načina rada.

[U priloženoj datoteci DZ_2_tvrtka.txt nalazi se organizacijska struktura tvrtke.](#)

Priložene datoteke su samo jedan primjer sa svojim sadržajem. Nastavnik će svoje testiranje provoditi i na drugim datotekama i sadržajima. Njih će se koristiti na prezentacijama zadaća.

Predlaže se da studenti/ce pripreme svoje dodatne datoteke s kojima će testirati svoje programe. To se može uraditi tako da se priloženim datotekama dodaju novi zapisi, promijene podaci postojećim zapisima i/ili obrišu pojedini zapise.

Potrebno je provjeriti ispravnost podataka u datotekama i svaki zapis koji nije ispravan preskače se uz ispis informacije o tome.

Opis problema.

Na početku potrebno je inicijalizirati sustav iznajmljivanja električnih vozila tako da se učitaju datoteke: [tvrtke](#), lokacija, vozila, kapaciteta lokacija, cjenika i osoba. Ako pojedini redak u datoteci nije ispravan potrebno ga je ispisati i zašto je neispravan.

Inicijalno sva vozila imaju punu bateriju i ukupan broj kilometara kod svakog vozila je 0. Slijedi inicijalizacija virtualnog vremena na temelju opcije početnog vremena (-t). Izvršavanje programa može

se obavljati na [dva načina koja se mogu izmjenjivati](#): interaktivni i skupni. Osnovi način rada je interaktivni, a skupni način provodi se kada je upisana opcija -s. Kod interaktivnog rada korisnik upisuje pojedinu aktivnost nakon čega mu se prikazuje rezultat te aktivnosti. [Ako pojedina aktivnost nije ispravna potrebno je ispisati zašto je neispravna](#). Zatim može izvršiti sljedeću aktivnosti itd. Kod skupnog načina datoteka uz opciju -s sadrži aktivnosti koje se izvršavaju jedna iza druge tako da se ispiše aktivnost a zatim se prikazuje rezultat te aktivnosti. Ako pojedini redak u datoteci nije ispravan potrebno ga je ispisati i zašto je neispravan. [Ako u datoteci za skupni način ne postoji aktivnost za kraj rada tada program prelazi u interaktivni način](#).

Korisniku se daje mogućnost da izvrši sljedeće aktivnosti:

- Pregled raspoloživih vozila odabrane vrste na odabranoj lokaciji
 - Sintaksa:
 - id_aktivnosti (1); vrijeme; id_korisnika; id_lokacije; id_vrste_vozila
 - Primjer:
 - 1; „2020-10-16 08:01:00“; 1; 6; 4
 - Opis:
 - U „2020-10-16 08:01:00“ korisnik Pero traži na lokaciji Kod placa broj raspoloživih automobila.
- Najam odabrane vrste vozila na odabranoj lokaciji
 - Sintaksa:
 - id_aktivnosti (2); vrijeme; id_korisnika; id_lokacije; id_vrste_vozila
 - Primjer:
 - 2; „2020-10-16 08:01:21“; 1; 6; 4
 - Opis:
 - U „2020-10-16 08:01:21“ korisnik Pero traži na lokaciji Kod placa najam automobila.
- Pregled raspoloživih mjesta odabrane vrste vozila za odabranu lokaciju
 - Sintaksa:
 - id_aktivnosti (3); vrijeme; id_korisnika; id_lokacije; id_vrste_vozila
 - Primjer:
 - 3; „2020-10-16 10:33:57“; 1; 2; 4
 - Opis:
 - U „2020-10-16 10:33:57“ korisnik Pero traži na lokaciji Bivši Mercator broj raspoloživih mjesta za automobile.
- Vraćanje vozila na odabranu lokaciju uz unos ukupnog broj kilometara te ispis računa
 - Sintaksa:
 - id_aktivnosti (4); vrijeme; id_korisnika; id_lokacije; id_vrste_vozila; broj_km
 - Primjer:
 - 4; „2020-10-16 10:38:24“; 1; 2; 4; 27
 - Opis:
 - U „2020-10-16 10:38:24“ korisnik Pero na lokaciji Bivši Mercator vraća unajmljeni automobil koji ima ukupno 27 km. Stavke računa su: 1 najam automobila – 30 kn, najma je bio 3 sata – $3 * 5 \text{ kn} = 15 \text{ kn}$, prethodno stanje bilo je 0 km znači da je prošao 27 km – $27 * 5 \text{ kn} = 135 \text{ kn}$. Račun ukupno iznosi $30 \text{ kn} + 15 \text{ kn} + 135 \text{ kn} = 180 \text{ kn}$.
- [Vraćanje neispravnog vozila na odabranu lokaciju uz unos ukupnog broj kilometara te ispis računa](#)
 - Sintaksa:

- id_aktivnosti (4); vrijeme; id_korisnika; id_lokacije; id_vrste_vozila; broj_km; opis_problema
- Primjer:
 - 4; „2020-10-16 11:22:10“; 1; 2; 4; 29; Prednje svjetlo je neispravno
- Opis:
 - U „2020-10-16 11:22:10“ korisnik Pero na lokaciji Bivši Mercator vraća unajmljeni automobil koji ima ukupno 29 km te prijavljuje da vozilo ima problem 'Prednje svjetlo je neispravno'. Stavke računa su: 1 najam automobila – 30 kn, najma je bio 3 sata – $3 * 5 \text{ kn} = 15 \text{ kn}$, prethodno stanje bilo je 27 km znači da je prošao 2 km – $2 * 5 \text{ kn} = 10 \text{ kn}$. Račun ukupno iznosi $30 \text{ kn} + 15 \text{ kn} + 10 \text{ kn} = 55 \text{ kn}$.
- Prijelaz iz interaktivnog u skupni načina
 - Sintaksa:
 - id_aktivnosti (5); datoteka
 - Primjer:
 - 5; DZ_2_aktivnosti_1.txt
 - Opis:
 - Potrebno je učitati datoteku DZ_2_aktivnosti_1.txt i izvršiti ju u skupnom načinu.
- Ispis podataka o stanju
 - Sintaksa:
 - id_aktivnosti (6); komanda {komanda ...} {id}
 - Ako je bez id tada se odnosi na cijelu tvrtku. id je id organizacijske jedinice.
 - Podaci se ispisuju u obliku tablice s poravnatim podacima u stupcima. Brojčani podaci nižih razina kumuliraju se u više razine.
 - Komande mogu biti:
 - struktura – struktura organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz naziv organizacijske jedinice ili lokacije.
 - stanje – stanje organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz broj mjesta, broj raspoloživih vozila i broj neispravnih vozila po vrstama vozila.
 - Primjer 1:
 - 6; struktura
 - Opis:
 - Potrebno je ispisati strukturu cijele tvrtke.
 - Primjer 2:
 - 6; struktura stanje 2
 - Opis:
 - Potrebno je ispisati strukturu i stanje organizacijske jedinice 2 (Varaždinska županija) i njenih dijelova.
- Ispis podataka o najmu i zaradi
 - Sintaksa:
 - id_aktivnosti (7); komanda {komanda ...} datum_1 datum_2 {id}
 - Datumi su u formatu dd.mm.gggg
 - Ako je bez id tada se odnosi na cijelu tvrtku. id je id organizacijske jedinice.

- Podaci se ispisuju u obliku tablice s poravnatim podacima u stupcima. Brojčani podaci nižih razina kumuliraju se u više razine.
- Komande mogu biti:
 - struktura – struktura organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz naziv organizacijske jedinice ili lokacije.
 - najam – broj najмова vozila organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz broj najma i trajanje najma po vrstama vozila. Podaci se odnose na razdoblje između datum_1 i datum_2 (oba uključena).
 - zarada – zarada organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz zaradu od najma po vrstama vozila. Podaci se odnose na razdoblje između datum_1 i datum_2 (oba uključena).
- Primjer 1:
 - 7; struktura 01.01.2020 31.11.2020
- Opis:
 - Potrebno je ispisati strukturu cijele tvrtke. Datumi nemaju posebnu ulogu.
- Primjer 2:
 - 7; struktura najam 01.01.2020 31.11.2020 2
- Opis:
 - Potrebno je ispisati strukturu podacima o najmu i zarad za organizacijsku jedinicu 2 (Varaždinska županija) i njenih dijelova u razdoblju 01.01.2020 - 31.11.2020
- Primjer 3:
 - 7; struktura najam zarada 01.01.2020 31.11.2020 2
- Opis:
 - Potrebno je ispisati strukturu podacima o najmu i zaradi za organizacijsku jedinicu 2 (Varaždinska županija) i njenih dijelova u razdoblju 01.01.2020 - 31.11.2020
- Ispis podataka o računima
 - Sintaksa:
 - id_aktivnosti (8); komanda {komanda} datum_1 datum_2 {id}
 - Datumi su u formatu dd.mm.gggg
 - Ako je bez id tada se odnosi na cijelu tvrtku. id je id organizacijske jedinice.
 - Podaci se ispisuju u obliku tablice s poravnatim podacima u stupcima.
 - Komande mogu biti:
 - struktura – struktura organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz naziv organizacijske jedinice ili lokacije.
 - računi – pojedinačni računi organizacijske jedinice i njenih dijelova do lokacija. Radi se o podacima vezano uz račune od najma po vrstama vozila. Za račun se ispisuje redni broj, vrijeme izdavanja, osoba kojoj je izdan, id i naziv lokacija najma, id i naziv lokacija vraćanja vozila i struktura računa. Podaci se odnose na razdoblje između datum_1 i datum_2 (oba uključena).
 - Primjer 1:

- 8; struktura 01.01.2020 31.11.2020
- Opis:
 - Potrebno je ispisati strukturu cijele tvrtke. Datumi nemaju posebnu ulogu.
- Primjer 2:
 - 8; struktura računi 01.01.2020 31.11.2020 2
- Opis:
 - Potrebno je ispisati strukturu s podacima o računima za organizacijsku jedinicu 2 (Varaždinska županija) i njenih dijelova u razdoblju 01.01.2020 - 31.11.2020
- Kraj programa
 - Sintaksa:
 - id_aktivnosti (0); vrijeme
 - Primjer:
 - 0; „2020-10-16 12:00:00“
 - Opis:
 - U „2020-10-16 12:00:00“ program završava s radom.

Kod ispisa strukture prvo se ispisuju podaci najviše razina koja je određene komandom (tvrtka tj. ishodišna organizacijska jedinica ako nije upisan id ili organizacijska jedinica za koju je upisan id). Slijedi prijelaz u nižu razinu. Uzima se prvi sastavni dio (dijete) te organizacijske jedinice i ispisuju se njegovi podaci, itd. Kada završi ispis svih dijelova pojedine organizacijske jedinice potrebno je ispisati kumulativne podatke njenih dijelova. Zatim se vraća na prethodnu razinu i nastavlja s ispisom ostalih dijelova (braće) itd.

Kada se podaci ispisuju u obliku tablice tada se za stupce s tekstualnim podacima primjenjuje lijevo poravnanje, a za stupce s bročanim podacima desno poravnanje. Broj znakova za formatiranje tekstualnih podataka određen je opcijom -dt, broj znakova za formatiranje cijelih brojeva (i cijelog dijela realnih brojeva) određen je opcijom -dc, a broj decimalnih mjesta za formatiranje realnih brojeve određen je opcijom -dd. Ako opcije nisu upisane tada se pretpostavlja da je -dt 30, -dc 5, -dd 2.

Potrebno je napraviti program tj. aplikaciju za komandni/linijski mod u operacijskom sustavu putem kojeg će se izvršiti opisane akcije. Program se NEĆE izvršavati putem razvojnog alata (IDE). Aplikacija NE smije biti s grafičkim korisničkim sučeljem. Kod izvršavanja programa NE smiju se ispisivati nepotrebni podaci (ostaci od testiranja i sl).

Izvršavanje programa može se obaviti na dva načina:

1. Putem opcija i naziva datoteka
2. Putem konfiguracijske datoteke.

Kod izvršavanja programa **putem opcija i naziva datoteka** upisuju se opcije i nazivi datoteka s podacima. Npr:

- interaktivni način rada

```
> {JVM} dkermek_zadaca_2[.jar | .exe] -v DZ_2_vozila.txt -l DZ_2_lokacije.txt -c DZ_2_cjenik.txt -k DZ_2_lokacije_kapaciteti.txt -o DZ_2_osobe.txt -t „2020-10-16 08:00:00“ -os DZ_2_tvrtka.txt
```

- skupni način rada

```
> {JVM} dkermek_zadaca_2[.jar | .exe] -v DZ_2_vozila.txt -l DZ_2_lokacije.txt -c DZ_2_cjenik.txt -k DZ_2_lokacije_kapaciteti.txt -o DZ_2_osobe.txt -t „2020-10-16 08:00:00“ -s DZ_2_aktivnosti.txt -os DZ_2_tvrtka.txt
```

Redoslijed opcija je proizvoljan kod unosa komandne linije tako da treba voditi brigu kod obrade primljenih podataka.

Program se može izvršavati sa svojeg direktorija putem naziva izvršne verzije programa (npr. dkermek_zadaca_1) kao što je prikazano u gornjoj liniji no potrebno je računati da će se sigurno izvršavati s nekog drugog direktorija (npr.: D:\UzDiz\DZ_2\) tako da će biti potrebno za izvršavanje upisati i apsolutnu (npr. D:\UzDiz\DZ_2\dkermek_zadaca_1\podaci) ili relativnu adresu/putanju (npr. ../../dkermek_zadaca_1\podaci) datoteka.

```
> {JVM} D:\UzDiz\DZ_2\dkermek_zadaca_1[.jar | .exe]
-v D:\UzDiz\DZ_2\DZ_2_vozila.txt
-l D:\UzDiz\DZ_2\DZ_2_lokacije.txt
-c D:\UzDiz\DZ_2\DZ_2_cjenik.txt
-k D:\UzDiz\DZ_2\DZ_2_lokacije_kapaciteti.txt
-o D:\UzDiz\DZ_2\DZ_2_osobe.txt
-t „2020-10-16 08:00:00“
-os D:\UzDiz\DZ_2\DZ_2_tvrtka.txt
-dt 30 -dc 5 -dd 2
```

Naziv datoteke može biti relativni u odnosu na poziciju izvršne datoteke (npr. DZ_2_vozila.txt) kao što je prikazano u gornjoj liniji no potrebno je računati da će sigurno datoteke biti na određenom direktoriju izvan projekta zbog čega je potrebno postaviti da naziv datoteke može biti na bilo kojem direktoriju s apsolutnom (npr. D:\UzDiz\DZ_2\DZ_2_vozila.txt) ili relativnom adresom (npr. ../../DZ_2/DZ_2_vozila.txt). Nazivi datoteka mogu biti drugačiji nego što su ovdje prikazani.

Kod izvršavanja programa putem konfiguracijske datoteke upisuju se naziv datoteke s konfiguracijskom podacima. Npr:

- interaktivni način rada

```
> {JVM} dkermek_zadaca_2[.jar | .exe] DZ_2_konfiguracija_1.txt
```

- skupni način rada

```
> {JVM} dkermek_zadaca_2[.jar | .exe] DZ_2_konfiguracija_2.txt
```

Program se može izvršavati sa svojeg direktorija putem naziva izvršne verzije programa (npr. dkermek_zadaca_2) kao što je prikazano u gornjoj liniji no potrebno je računati da će se sigurno izvršavati s nekog drugog direktorija (npr.: D:\UzDiz\DZ_2\) tako da će biti potrebno za izvršavanje upisati i apsolutnu (npr. D:\UzDiz\DZ_2\dkermek_zadaca_2\podaci) ili relativnu adresu/putanju (npr. ../../dkermek_zadaca_2\podaci) datoteka.

```
> {JVM} D:\UzDiz\DZ_2\dkermek_zadaca_2[.jar | .exe]  
D:\UzDiz\DZ_2\DZ_2_konfiguracija_2.txt
```

Naziv datoteke može biti relativni u odnosu na poziciju izvršne datoteke (npr. DZ_2_konfiguracija_2.txt) kao što je prikazano u gornjoj liniji no potrebno je računati da će sigurno datoteke biti na određenom direktoriju izvan projekta zbog čega je potrebno postaviti da naziv datoteke može biti na bilo kojem direktoriju s apsolutnom (npr. D:\UzDiz\DZ_2\DZ_2_konfiguracija_2.txt) ili relativnom adresom (nr. ../..../DZ_2 DZ_2_konfiguracija_2.txt). Nazivi datoteka mogu biti drugačiji nego što su ovdje prikazani.

U datoteci konfiguracije sustava (npr. DZ_2_konfiguracija_1.txt ili DZ_2_konfiguracija_2.txt) postoje zapisi koji se tretiraju kao oblik ključ=vrijednost. Ključevi su ekvivalentni opcijama, a vrijednosti su nazivi datoteka ili brojevi za formatiranje ispisa podataka. Npr. za opciju -v postoji ključ vozila, za opciju -l postoji ključ lokacije, za opciju -c postoji ključ cjenik, za opciju -k postoji ključ kapaciteti, za opciju -o postoji ključ osobe, za opciju -t postoji ključ vrijeme, za opciju -s postoji ključ aktivnosti, za opciju -os postoji ključ struktura, za opciju -dt postoji ključ tekst, za opciju -dc postoji ključ cijeli, za opciju -dd postoji ključ decimala. Za nazive datoteka koje se koriste unutar konfiguracijske datoteke vrijede ista pravila kao i kada se koriste opcije i nazivi datoteka (relativni ili apsolutni naziv).

U ishodišnom direktoriju projekta treba priložiti datoteku dokumentacije {LDAP_korisničko_ime}_zadaca_2.pdf kako je opisano u dokumentima „Preporuke u vezi zadaća“ i „Opći model ocjenjivanja zadaća“.

Napomena: SMIJU SE KORISTITI SAMO UZORCI DIZAJNA KOJI SU OBRAĐENI NA NASTAVI DO OBJAVE ZADAĆE (zadnji je bio **Template Method**). Preporučuje se da ponovno razmislite o svojem rješenju 1. zadaće i da ga refaktorirate u skladu s opisom 2. zadaće i uzorcima dizajna koji su obrađeni na nastavi. Ne smiju se koristiti ugrađene osobine odabranog programskog jezika za realizaciju funkcionalnosti pojedinih uzoraka dizajna. Ne smiju se koristite dodatne biblioteke/knjižnice klasa.