

Uzorci dizajna

Moja naslovnica / Moji e-kolegiji / Diplomski studij / FOI diplomski studij / UzDiz
/ Uzorci iz knjige POSA 1 / Domaća zadata 3.

Domaća zadata 3.

Naziv: ToF 3. dio (Things of FOI - 3. dio)

Naziv projekta: {LDAP_korisničko_ime}_zadaca_3

Ishodišni direktorij projekta: {LDAP_korisničko_ime}_zadaca_3

Naziv rješenja: {LDAP_korisničko_ime}_zadaca_3.zip

Prije predavanja projekta potrebno je napraviti Clean na projektu. Zatim cijeli projekt sažeti u .zip (NE .rar) format s nazivom {LDAP_korisničko_ime}_zadaca_3.zip i predati u Moodle. Uključiti izvorni kod, primjere datoteke podataka, popunjenu datoteku dokumentacije i popunjen obrazac za zadata (obje datoteke u ishodišnom direktoriju projekta).

Uvod.

Zadata se nastavlja na opis 2. zadata uz određene promjene i nove elemente. ToF sastoji se od uređaja (senzora, aktuatora i sl) koji primaju kontrolne signale (poruke), obavljaju mjerenja i druge aktivnosti (pomicanje mehanizma, npr. otvaranje ili zatvaranje, paljenje ili gašenje i sl), vraćaju informaciju o mjerenim elementima i statusu. U FOI postoji više prostorija, dvorana, područja i sl. (sve njih zovemo mjesta) na kojima se mogu postaviti uređaji. Podaci o mjestima, senzorima, aktuatorima i njihovom rasporedu nalaze se u priloženim datotekama. Datoteke su u formatu CSV (točka zarez) tako da prvi redak sadrži opis strukture datoteke a ostali retci su podaci. U datoteci rasporeda prva 3 retka su opis strukture.

U priloženoj datoteci DZ_3_mjesta.txt nalazi se popis mjesta s njihovim osobinama.

U priloženoj datoteci DZ_3_senzori.txt nalazi se popis raspoloživih modela senzora s njihovim osobinama.

U priloženoj datoteci DZ_3_aktuatori.txt nalazi se popis raspoloživih modela aktuatora s njihovim osobinama.

U priloženoj datoteci DZ_3_raspored.txt nalazi se popis stvarnih senzora i aktuatora koji su pridruženi pojedinom mjestu, te popis senzora koji su pridruženi pojedinom aktuatoru.

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 zadata.

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

Opis problema.

U sustavu postoji samo jedan generator slučajnog broja koja se inicijalizira zadanim sjemenom prilikom pokretanja programa. Generator ima metode public int dajSlucajniBroj(int odBroja, int doBroja) i public float dajSlucajniBroj(float odBroja, float doBroja).

Program kod pokretanja provjerava upisane argumente, priprema ekran za rad tako da dijeli ekran u dva logička dijela (prozora) od kojih je gornji za prikaz podataka (broj redaka odgovara ukupnom broju redaka (opcija -br) umanjeno za broj redaka za unos komandi (opcija -brk)), a donji je za unos komandi (broj redaka odgovara opciji -brk).

Za prikaz podataka u pojedinom prozoru treba koristiti **direktno pozicioniranje na temelju ANSI/VT100 kontrolnih ESC**

kodova (<http://www.termsys.demon.co.uk/vtansi.htm>). Odnosno, sve radnje vezane uz ekran treba obavljati na temelju **ANSI/VT100 kontrolnih ESC kodova**. Za rad s korisničkim sučeljem treba se koristiti MVC.

Početna pozicija za ispis je gornja lijeva koordinata prozora. Svaki novi ispis prikazuje se u retku ispod prethodnog retka dok ne popuni cijeli gornji dio. U tom slučaju u donjem dijelu se ispiše da se za nastavak treba pritisnuti tipka n/N. Kada se n/N pritisne, obriše se gornji prozor te se nastavlja s ispisom kao što je prethodno opisano.

Sve akcije oko prikazivanja podataka trebaju biti realizirane vlastitim rješenjem. Nije dozvoljeno koristiti biblioteke za prikaz podataka u VT100 terminalu.

Na početku se učitavaju mjesta te je potrebno osigurati da tijekom izvršavanja programa postoji samo jedna instanca pojedinog mjesta na bazi jednoznačnog broja (identifikatora). Slijedi učitavanje modela senzora te je potrebno osigurati da tijekom izvršavanja programa postoji samo jedna instanca pojedinog modela senzora na

bazi jednoznačnog broja (identifikatora). Slijedi učitavanje modela aktuatora te je potrebno osigurati da tijekom izvršavanja programa postoji samo jedna instanca pojedinog modela aktuatora na bazi jednoznačnog broja (identifikatora).

Svako mjesto ima zadani broj senzora i aktuatora. Iz datoteke rasporeda preuzimaju se senzori i aktuatori koji su pridruženi određenom mjestu uz provjeru odgovaraju li tipu mjesta temeljeg svog modela i potrebnom broju za mjesto. Također se iz datoteke rasporeda preuzimaju koji su senzori pridruženi pojedinom aktuatoru.

U datoteci rasporeda podaci mogu biti različito zapisani tako da se prvo može nalaziti npr. redak s podacima o pridruženim sensorima za aktuator, a aktuator još nije definiran ili senzori nisu definirani i sl. Redoslijed pojavljivanja podataka nije bitan ali je važno da se provjere da li postoje svi potrebni podaci i da li su ispravni. Prilikom učitavanja datoteke rasporeda ne smije se provoditi sortiranje podataka iz datoteke i/ili bilo koji drugi oblik obrade podataka iz datoteke rasporeda. Potrebno je ispisati sve neispravne podatke i objašnjenje.

Slijedi faza inicijalizacije sustava koja prolazi po mjestima pa se svakom uređaju pošalje inicijalizacijska poruka. Uređaj vraća svoj status (1 - u redu, 0 - pogreška), temeljem generatora slučajnog broja uz prosječnu ispravnost (opcija -pi odnosno komanda PI n). Ako je pogreška uređaj se ne koristi u nastavku ali i dalje je prisutan u svim pregledima/ispisima.

Potrebno je dodati vlastitu funkcionalnost projektu tako da se koristi GOF uzorak Chain of Responsibility. To znači da je u dokumentaciju projekta potrebno dodati opis funkcionalnosti koja nije zadana u opisu zadaće i koja će se realizirati zadanim GOF uzorkom i izvršavati kada se pozove upisanom komandom (VF).

U nastavku se ispisuje da se očekuje unos komande:

M x - ispis podataka mjesta x

S x - ispis podataka senzora x

A x - ispis podataka aktuatora x

S - ispis statistike

SP - spremi podatke (mjesta, uređaja)

VP - vrati spremljene podatke (mjesta, uređaja)

C n - izvršavanje n ciklusa dretve (1-100)

VF - izvršavanje vlastite funkcionalnosti

PI n - prosječni % ispravnosti uređaja (0-100)

I - izlaz.

Komande se tako dugo unose dok se ne unese I nakon čega se prekida program.

Kod svih ispisa treba se koristiti tablični prikaz s poravnatim vrijednostima u stupcima (tekstualni lijevo poravnati, brožčani desno poravnati).

Kod spremanja podataka ne smije se koristiti datoteka i postoji samo jedno "spremište" za podatke.

Izvršavanje dretve u zadanom broju ciklusa provjerava sva mjesta. Provjera mjesta polazi od utvrđivanja statusa njegovih uređaja. Uređaj koji 3 puta u nizu (3 slijedna ciklusa dretve za isto mjesto) vrati pogrešku, označava se kao neispravan u radu te ga je potrebno odmah zamijeniti novim uređajem istog modela i inicijalizirati, s time da je njegov id veći za 1 od najvećeg broja uređaja koji postoje. Neispravan uređaj i dalje je prisutan u svim pregledima. Potrebno je ispisati informacije u obavljanim poslovima. Nakon provjere (utvrđivanje stanja) od senzora se traži očitavanje vrijednosti i ispiše se informacija o tome. Kod aktuatora se, samo u slučaju kada je bila promjena vrijednosti kod barem jednog senzora koji je pridružen tom aktuatoru, prvo traži očitavanje vrijednosti a zatim izvršavanje radnje. U jednom ciklusu provjere mjesta aktuator može samo jednom obaviti svoju radnju. Kod onih koji ima binarnu vrstu obavlja se suprotna radnja trenutnom stanju. Kod ostalih se generira slučajni broj u intervalu pojedinog aktuatora. Taj broj se dodaje u smjeru prethodne operacije. Prvo se počinje s povećavanjem vrijednosti. Kada se dođe do maksimalne vrijednosti aktuatora tada se mijenja smjer i počinje se sa smanjivanjem vrijednosti. I tako se provodi šetnja od jedne (min) do druge strane vrijednosti (max) i obratno. Ispisuju se informacije o tome.

Program se mora izvršavati unutar programa koji omogućava emulaciju VT100 terminala. Nastavnik koristi ConEmu (<http://www.fosshub.com/ConEmu.html>) i zadaće će se izvršavati u tom programu pa je preporuka da ga koristite i studenti.

Kod izvršavanja programa upisuju se opcije, s time je njihov redoslijed proizvoljan, a mogu biti sljedeće:

- br broj redaka na ekranu (24-40). Ako nije upisana opcija, uzima se 24.
- bs broj stupaca na ekranu (80-160). Ako nije upisana opcija, uzima se 80.
- brk broj redaka na ekranu za unos komandi (2-5). Ako nije upisana opcija, uzima se 2.
- pi prosječni % ispravnosti uređaja (0-100). Ako nije upisana opcija, uzima se 50.
- g sjeme za generator slučajnog broja (u intervalu 100 - 65535). Ako nije upisana opcija, uzima se broj milisekundi u trenutnom vremenu na bazi njegovog broja sekundi i broja milisekundi.
- m naziv datoteke mjesta
- s naziv datoteke senzora

-a naziv datoteke aktuatora

-r naziv datoteke rasporeda

-tcd trajanje ciklusa dretve u sek. Ako nije upisana opcija, uzima se slučajni broj u intervalu 1 - 17.

--help pomoć za korištenje opcija u programu.

```
> dkermek_zadaca_3 -br 30 -bs 100 -brk 3 -g 717 -m DZ_3_mjesta.txt -s  
DZ_3_senzori.txt -a DZ_3_aktuatori.txt -r DZ_3_raspored.txt -alg  
org.foi.uzdiz.dkermek.AlgoritamAbecedno -tcd 5
```

> dkermek_zadaca_3 --help

Program se može izvršavati sa svojeg direktorija putem naziva (npr. dkermek_zadaca_3) kao što je prikazano u gornjoj liniji no potrebno je računati da će se sigurno izvršavati s nekog drugom direktorija (npr.: D:\UzDiz\DZ_3\) tako da će biti potrebno za izvršavanje upisati apsolutnu (npr. D:\UzDiz\DZ_3\dkermek_zadaca_3\dkermek_zadaca_3) ili relativnu adresu (npr. ../..\\dkermek_zadaca_3\dkermek_zadaca_3).

Nazivi datoteka mogu biti relativni u odnosu na poziciju izvršne datoteke (npr. DZ_3_mjesta.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_3\DZ_3_mjesta.txt) ili relativnom adresom (nr. ../..\\DZ_3\DZ_3_mjesta.txt).

Potrebno je napraviti program tj. aplikaciju za **komandni/linijski mod** u operacijskom sustavu putem kojeg će se izvršiti opisane akcije. Aplikacija NE smije biti s grafičkim korisničkim sučeljem.

U ishodišnom direktoriju projekta treba priložiti datoteku dokumentacije {LDAP_korisničko_ime}_zadaca_3.{doc|pdf} u kojoj se na 1. stranici nalazi objašnjenje razloga odabira pojedinog uzorka dizajna (max 1 str A4, font min 10), na 2. stranici opis vlastite funkcionalnosti i promjene u odnosu na rješenje 2. zadatke (max 1 str A4, font min 10), a na 3. stranici dijagram rješenja (max 1 str A4, font min 10). U dijagramu treba jasno prikazati koje klase su sastavni dio pojedinog uzorka dizajna. Ne stavljaju se nazivi atributa i metoda.





U ishodišnom direktoriju projekta treba priložiti datoteku {LDAP_korisničko_ime}_obrazac_za_zadacu_3.{doc|pdf} u kojoj se nalazi popunjen obrazac za zadataku.

Treba pronaći prikladne GOF i POSA1 uzorke dizajna za opisane probleme. Za rad s korisničkim sučeljem treba se koristiti MVC.


Napomena: koristiti samo GOF i POSA1 uzorke dizajna. **PODRAZUMIJEVA SE DA SE SMIJU KORISTITI SAMO UZORCI DIZAJNA KOJI SU OBRAĐENI NA NASTAVI DO OBJAVE ZADAĆE (04.12.2017. zadnji je bio MVC).** Ne smiju se koristiti ugrađene osobine odabranog programskog jezika za realizaciju funkcionalnosti pojedenih uzoraka dizajna.

Jezici implementacije: C#, Java.

Razvojni alati: NetBeans (8.2), Visual Studio 2015

 DZ_3_aktuatori.txt
 DZ_3_mjesta.txt
 DZ_3_raspored.txt
 DZ_3_senzori.txt

Stanje predane zadaće

Stanje predane zadaće	Zadaca nije predana
Ocjena	Nije ocijenjeno
Rok predaje	Subota, 30 Prosinac 2017, 23:55
Preostalo vrijeme	16 dana
Posljednja izmjena	-
Napomene uz zadacu	 Komentari (0)

Predajte zadacu

Izmjenite svoju predanu zadacu

Prijavljeni ste kao **Karlo Grlić** ([Odjava](#))

[UzDiz](#)

[Hrvatski \(hr\)](#)

[English \(en\)](#)

[Hrvatski \(hr\)](#)