

PROGRAMMERS

Zadaća 5

Jašarević Ahmed
Kos Harun

Zenica, juli 2022. godine

1 Zadatak 1

Za upravljanje nekim robotskim manipulatorom koristi se skup upravljačkih funkcija čiji su prototipovi sljedeći

```
void AktivirajRobota();
void DeaktivirajRobota();
void Rotiraj(Pravci&orijentacija, Smjernakojustranu);
void Idi(int&x, int&y, Pravci&orijentacija, intkorak);
void IspisiPoziciju(intx, inty, Pravci&orijentacija);
void PrijaviGresku(KodoviGresakakodgreske);
void IzvrsiKomandu(Komandekomanda, intparametar, int&x, int&y, Pravci&orijentacija);
```

Funkcije “**AktivirajRobota**” i “**DeaktivirajRobota**” aktiviraju odnosno deaktiviraju robota. Robot izvršava komande samo kada je aktivan. U suprotnom, dok je deaktiviran, robot ignorira svaku komandu koja mu se uputi, odnosno prihvata komandu, ali je ne izvršava.

Funkcija “**Rotiraj**” mijenja pravac u kojem robot gleda, odnosno obrće ga nalijevo ili nadesno za 45° ovisno od vrijednosti parametra “**na koju stranu**”. Ovaj parametar može imati samo dvije vrijednosti “**Nalijevo**” ili “**Nadesno**”. Njegov tip je “**Smjer**”, što predstavlja pobrojani tip koji je deklariran (na globalnom nivou) kao **enum class Smjer Nalijevo, Nadesno**;

Parametar “**orijentacija**” predstavlja pravac u kojem robot trenutno gleda. Tip ovog parametra je “**Pravci**”, što je pobrojani tip koji je (takoder na globalnom nivou) deklariran kao

```
{enum class Pravci {Sjever, Sjeveroistok, Istok, Jugoistok, Jug,
    Jugozapad, Zapad, Sjeverozapad}}
```

Značenja pojedinih pobrojanih konstanti u ovom tipu su očigledna. Funkcija “**Rotiraj**” utiče na vrijednost parametra “**orijentacija**”, tako da će on po završetku funkcije sadržavati novu orijentaciju robota nakon obavljene rotacije. U slučaju da je robot deaktiviran ova funkcija ne radi ništa (isto vrijedi za sve ostale funkcije, ukoliko nije rečeno drugačije).

Funkcija “**Idi**” pomjera robota za broj koraka zadan parametrom “**korak**” u smjeru koji je određen parametrom “**orijentacija**”, pri čemu se u slučaju “**mješovitih**” pravaca kao što je npr. jugoistok jedan korak na jugoistok tretira kao da je ekvivalentan jednom koraku na jug i jednom koraku na istok (slično vrijedi i za sve ostale “**mješovite**” pravce). Trenutna pozicija robota određena je vrijednostima parametara “**x**” i “**y**” na ulazu, a po završetku funkcije isti parametri trebaju sadržavati ažuriranu poziciju. Ukoliko je parametar “**korak**” negativan, kretanje se vrši u smjeru suprotnom od tekuće orijentacije za iznos jednak apsolutnoj vrijednosti parametra.

Funkcija “**IspisiPoziciju**” ispisuje na ekran informacije o poziciju robota u sljedećem formatu: **Robot** je status, **nalazi se na poziciji** (x, y) i **gleda na** orijentacija. “**status**” može biti “aktivan” ili “neaktivan”, ovisno da li je robot trenutno aktivan ili ne. “**x**” i “**y**” su pozicija robota, koja je određena istoimenim parametrima, dok “**orijentacija**” može biti “sjever”, “sjeveroistok”, “istok”, “jugoistok”, “jug”, “jugozapad”, “zapad” ili “sjeverozapad”, u zavisnosti od vrijednosti istoimenog parametra. Funkcija “**PrijaviGresku**” ima parametar “**kod greske**” tipa “**KodoviGresaka**” koji predstavlja pobrojani tip definiran kao

```
enum class KodoviGresaka {PogresnaKomanda, NedostajeParametar, SuvisanParametar, NeispravanParametar};
```

Ova funkcija, u zavisnosti od vrijednosti parametra koji joj je proslijeđen, na ekran ispisuje neki od tekstova koji se mogu javiti kao greška u radu sa robotom, u skladu sa sljedećom tabelom (objašnjenje će uslijediti poslije):

Kod Greske	Tekst koji treba ispisati
PogresnaKomanda	Nerazumljiva komanda!
NedostajeParametar	Nerazumljiva komanda!
NeispravanParametar	Nerazumljiva komanda!
SuvisanParametar	Nerazumljiva komanda!

Funkcija “**IzvršiKomandu**” ima parametar “**komanda**” tipa “Komande” koji predstavlja pobrojani tip definiran kao

```
enum class Komande {Aktiviraj, Deaktiviraj, Nalijevo, Nadesno, Idi, Kraj};
```

Ova funkcija, u zavisnosti od vrijednosti parametra “**komanda**”, izvršava zadanu komandu, pozivom odgovarajuće funkcije za izvršenje komande (osim ukoliko je komanda “**Kraj**”; tada funkcija ne radi ništa). Parametar “**parametar**” koristi se samo ukoliko je komanda “**Idi**”, i on tada predstavlja vrijednost koja se prosljeđuje istoimenoj funkciji (tj. funkciji “**Idi**”). U svim ostalim slučajevima, vrijednost parametra “**parametar**” se ignorira. Preostali parametri funkcije “**IzvršiKomandu**” predstavljaju informaciju o poziciji i orijentaciji robota. Komunikacija između korisnika i robota vrši se posredstvom funkcije “**UnosKomande**” koja ima sljedeći prototip:

```
bool UnosKomande(Komande &komanda, int &parametar, KodoviGresaka &kod greske);
```

Ova funkcija očekuje od korisnika da zada unos komande putem tastature. Legalne komande su sljedeće (u komandama “A+” i “A-” znakovi “+” i “-” su dio komande a ne parametar): Razmaci ispred i iza komande su dozvoljeni, ali bilo kakav neočekivani znak (osim razmaka) nakon komande tretira se kao suvišan parametar. Komanda “I” je procjena cijelim brojem koji predstavlja broj koraka koji će robot napraviti. Razmaci između komande i broja su dozvoljeni (ali ne i obavezni), pri čemu se bilo šta što ne predstavlja ispravan cijeli broj (uključujući suvišne znakove iza broja) tretira kao neispravan parametar. U slučaju

Komanda	Značenje
A+	Aktivira robota
A-	Deaktivira robota
L	Rotira robota nalijevo
D	Rotira robota nadesno
I korak	Pomjera robota za navedeni broj koraka
K	Završetak rada programa

da se prepozna ispravna komanda, funkcija smješta njen kod u parametar “**komanda**”, eventualni parametar komande smješta se u parametar “**parametar**” (u slučaju da komanda nema parametra, vrijednost parametra “parametar” je nedefinirana), a funkcija

vraca kao rezultat logicku vrijednost **“true”** kao signal da je komanda prepoznata. Parametar **“kod greske”** tada je nedefiniran. U suprotnom, ukoliko nije prepoznata ispravna komanda, kod greške se smješta u parametar **“kod greske”**, parametri **“komanda”** i **“parametar”** su nedefinirani, a funkcija vraca kao rezultat logicku vrijednost **“false”** kao signal da nije prepoznata ispravna komanda. Eventualni razmaci ispred komande kao i nakon komande su dozvoljeni i ne trebaju uzrokovati prikavu greške. Potrebno je predvidjeti sve što bi korisnik eventualno mogao unijeti (funkcija kao i program koji je koristi ne smije da “crkne” šta god korisnik unio). Na početku rada, robot je aktivan, nalazi se na poziciji (0, 0) i gleda na sjever. Napisane funkcije treba demonstrirati u glavnom programu u kojem ce se u petlji zahtijevati unos komande pozivom funkcije **“UnosKomande”**, nakon čega ce se odgovarajuca komanda izvršiti (pozivom funkcije **“IzvršiKomandu”**) ili ce se prijaviti greška (pozivom funkcije **“PrijaviGresku”**). Petlja se prekida nakon što se zada komanda **“Kraj”**. Tada program ispisuje poruku **“Dovidjenja!”** i završava sa radom. Slijedi primjer kako može izgledati dijalog između korisnika i programa:

```
Robot je aktivan, nalazi se na poziciji(0,0) i gleda na sjever.
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(0,0) i gleda na sjeveroistok.
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(0,0) i gleda na istok.
Unesikomandu:I
Komanda trazi parametar koji nije naveden!
Unesikomandu:I5
Robot je aktivan, nalazi se na poziciji(5,0) i gleda na istok..
Unesikomandu:A-
Robot je neaktivan, nalazi se na poziciji(5,0) i gleda na istok.
Unesikomandu:I3
Robot je neaktivan, nalazi se na poziciji(5,0) i gleda na istok.
Unesikomandu:D
Robot je neaktivan, nalazi se na poziciji(5,0) i gleda na istok.
Unesikomandu:A+
Robot je aktivan, nalazi se na poziciji(5,0) i gleda na istok.
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(5,0) i gleda na jugoistok.
Unesikomandu:I4
Robot je aktivan, nalazi se na poziciji(9,-4) i gleda na jugoistok.
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(9,-4) i gleda na jug.
Unesikomandu:S
Nerazumljiva komanda!
Unesikomandu:IXY2
Parametar komande nije ispravan!
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(9,-4) i gleda na jugozapad.
Unesikomandu:D
Robot je aktivan, nalazi se na poziciji(9,-4) i gleda na zapad.
Unesikomandu:I2XY
Parametar komande nije ispravan!
Unesikomandu:I0
Robot je aktivan, nalazi se na poziciji(9,-4) i gleda na zapad.
Unesikomandu: I 1
Robot je aktivan, nalazi se na poziciji(10,-4) i gleda na zapad.
Unesikomandu:I3
Robot je aktivan, nalazi se na poziciji(7,-4) i gleda na zapad.
Unesikomandu:L2
```

```

Zadan je suvisan parametar nakon komande!
Unesikomandu:L
Robot je aktivan, nalazi se na poziciji(3,-4) i gleda na jugozapad.
Unesikomandu:KK
Zadan je suvisan parametar nakon komande!
Unesikomandu:K
Dovidjenja!

```

U slučaju da se prilikom kretanja robota premači opseg tipa “int”, odgovarajuća koordinata robota koja bi izašla izvan opsega treba da se postavi tačno na granicu opsega, kao da se oko polja po kojem se kreće robot nalazi “zid” koji ne da robotu da ide dalje iza zida. Na primjer, ukoliko pretpostavimo da je tekuća x koordinata robota 9990, a da je maksimalna vrijednost koja može stati u tip “int” 10000 (nije tolika, ali to je samo primjer), nakon pokušaja kretanja na istok u iznosu od recimo 50 koraka (uglavnom, viš se od 10), nova y koordinata robota treba biti 10000. Pri tome, ne smijete unaprijed pretpostaviti koliki je maksimalni opseg za tip “int” (npr. osloniti se na to da je na kompajleru koji koristimo taj opseg od - 2147483648 do 2147483647), nego te informacije trebate saznati programskim putem.

VAZNA NAPOMENA: Informacije o poziciji i orijentaciji robota ne smiju se čuvati u globalnim promjenljivim, nego se te informacije moraju razmjenjivati između funkcija putem prenosa parametara. Također, u programu nije dozvoljeno koristiti unos sa tastature u promjenljive tipa string ili niz znakova, nego sva procesiranja ulaza treba vršiti direktnim čitanjima iz spremnika ulaznog toka. Nepoštovanje ovih ograničenja biće kažnjeno davanjem 0 poena na čitav zadatak!

Rješenje

```

#include <iostream>
#include <string>
#include <string.h>

using namespace std;

int x=0, y=0;
bool negativanBroj=false;
bool greska = false;
bool greska1= false;

bool aktivan = 123 ;

enum class Smjer {Nalijevo=0, Nadesno};

enum class Pravci {Sjever=0, Sjeveroistok, Istok, Jugoistok, Jug,
    Jugozapad, Zapad, Sjeverozapad};

enum class KodoviGresaka {PogresnaKomanda, NedostajeParametar,
    SuvisanParametar, NeispravanParametar};

enum class Komande {Aktiviraj, Deaktiviraj, Nalijevo, Nadesno, Idi,
    Kraj};

bool UnosKomande (Komande &komanda, int &parametar, KodoviGresaka &
    kod_greske)

```

```

{

}

void AktivirajRobota()
{
    aktivan=1;
}

void DeaktivirajRobota()
{
    aktivan=0;
}

void Rotiraj(Pravci &orijentacija, Smjer na_koju_stranu)
{

}

string status(bool aktivan){
    if(aktivan == 1)
        return "aktivan";
    else{
        return "neaktivan";
    }
}

void IspisiPoziciju (int x, int y, Pravci orijentacija)
{
    if(orijentacija==Pravci::Sjever)
        cout<<"Robot je " << status(aktivan) << ", nalazi se na
poziciji ("<<x<<","<<y<<")"<<" i gleda na sjever" << endl;
    if(orijentacija==Pravci::Sjeveroistok)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na sjeveroistok" <<endl;
    if(orijentacija==Pravci::Istok)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na istok"<<endl;
    if(orijentacija==Pravci::Jugoistok)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na jugoistok"<<endl;
    if(orijentacija==Pravci::Jug)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na jug"<<endl;
    if(orijentacija==Pravci::Jugozapad)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na jugozapad"<<endl;
    if(orijentacija==Pravci::Zapad)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na zapad"<<endl;
    if(orijentacija==Pravci::Sjeverozapad)
        cout<<"Robot je " << status(aktivan)<<", nalazi se na poziciji
("<<x<<","<<y<<")"<<" i gleda na sjeverozapad"<<endl;
}

```

```

void Idi(int &x, int &y, Pravci orijentacija, int integerbroj)
{
    int n;
    if(aktivan == 0){
        IspisiPoziciju(x,y,Pravci::Sjever);
    }
    else{
        if(orijentacija==Pravci::Sjever && integerbroj >= 0) {
            y+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Sjever);
        }
        if(orijentacija==Pravci::Sjever && integerbroj < 0) {
            y+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Sjever);
        }
        if(orijentacija==Pravci::Sjeveroistok && integerbroj >= 0) {
            x+=integerbroj;
            y+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Sjeveroistok);
        }
        if(orijentacija==Pravci::Sjeveroistok && integerbroj <0) {
            x+=integerbroj;
            y+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Sjeveroistok);
        }
        if(orijentacija==Pravci::Istok && integerbroj >= 0) {
            x+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Istok);
        }
        if(orijentacija==Pravci::Istok && integerbroj <0) {
            x+=integerbroj;
            IspisiPoziciju(x,y,Pravci::Istok);
        }
        if(orijentacija==Pravci::Jugoistok && integerbroj >= 0) {
            x+=integerbroj;
            y-=integerbroj;
            IspisiPoziciju(x,y,Pravci::Jugoistok);
        }
        if(orijentacija==Pravci::Jugoistok && integerbroj < 0) {
            x+=integerbroj;
            y-=integerbroj;
            IspisiPoziciju(x,y,Pravci::Jugoistok);
        }
        if(orijentacija==Pravci::Jug && integerbroj >= 0) {
            y-=integerbroj;
            IspisiPoziciju(x,y,Pravci::Jug);
        }
        if(orijentacija==Pravci::Jug && integerbroj < 0) {
            y-=integerbroj;
            IspisiPoziciju(x,y,Pravci::Jug);
        }
        if(orijentacija==Pravci::Jugozapad && integerbroj >= 0) {
            x-=integerbroj;
            y-=integerbroj;
        }
    }
}

```

```

        IspisiPoziciju(x,y,Pravci::Jugozapad);
    }
    if(orientacija==Pravci::Jugozapad && integerbroj < 0) {
        y-=integerbroj;
        x-=integerbroj;
        IspisiPoziciju(x,y,Pravci::Jugozapad);
    }
    if(orientacija==Pravci::Zapad && integerbroj >= 0) {
        x-=integerbroj;
        IspisiPoziciju(x,y,Pravci::Zapad);
    }
    if(orientacija==Pravci::Zapad && integerbroj <0) {
        x-=integerbroj;
        IspisiPoziciju(x,y,Pravci::Zapad);
    }
    if(orientacija==Pravci::Sjeverozapad && integerbroj >= 0) {
        y+=integerbroj;
        x-=integerbroj;
        IspisiPoziciju(x,y,Pravci::Sjeverozapad);
    }
    if(orientacija==Pravci::Sjeverozapad && integerbroj < 0) {
        y+=integerbroj;
        x-=integerbroj;
        IspisiPoziciju(x,y,Pravci::Sjeverozapad);
    }

    n++;
}
}

void PrijaviGresku(KodoviGresaka kod_greske)
{
    if (kod_greske==KodoviGresaka::PogresnaKomanda)
        cout<<"Nerazumljiva komanda! "<<endl;

    if (kod_greske==KodoviGresaka::NeispravanParametar)
        cout<<"Parametar komande nije ispravan! "<<endl;
        if (kod_greske==KodoviGresaka::SuvisanParametar)
            cout<<"Zadan je suvisan parametar nakon komande!"<<endl;
    if (kod_greske==KodoviGresaka::NedostajeParametar)
        cout<<"Komanda trazi parametar koji nije naveden!"<<endl;
}

void IzvrsiKomandu(Komande komanda, int parametar, int &x, int &y,
    Pravci &orientacija)
{
}

string ocistiKomandu(string &komanda){
    string ociscenaKomanda="";
    for(int i=0; i<komanda.length(); i++)
    {

```



```

        if(komanda[i]==' ')
            continue;
        else
            ociscenaKomanda=ociscenaKomanda+komanda[i];
    }
    komanda = ociscenaKomanda;
return komanda;
}

bool validirajUnos(string komanda)
{
    if(komanda.length()==1 )
        return false;
    return true;
}

string uzmiKomandu(string komanda)
{
    char str[100],num[100], I[100];
    int i, j=0, a=0, velicina = 0;
    string sve;

    ocistiKomandu(komanda);

    for(int i = 0; komanda[i] != '\0'; i++) {
        velicina++;
        if(komanda[i] == 'I' && komanda[i+1] == 45) {
            negativanBroj = true;
        }

    }

    for(i=0; i<velicina; i++) {
        if (komanda[0] == 'I' && komanda[1] > 47 && komanda [1] < 58 ) {
            greska = false;

        }

        if (komanda[0] == 'I'          && komanda[i] > 64 && komanda [i] <
123) {
            greska = true;

        }

        if(komanda[i]>='0' && komanda[i]<='9') {
            num[j] = komanda[i];
            j++;
        } else {
            if(komanda[i]!='I' ) {
                continue;
            } else {
                sve=sve+komanda[i];
            }
        }
    }
}

```

```

    }

}

    }
    string brojevi;
    for(int i=0; i<j; i++) {
        brojevi=brojevi+ num[i];

    }
    string minusString;
    char minus='-';
    if(negativanBroj == 1) {
        minusString=minus + brojevi;
        return minusString;
    } else
        return brojevi;
}

Pravci ispisiOrijentaciju(int &n){
Pravci orijentacija;

        if(n==0)
            orijentacija=Pravci::Sjever;
        if(n==1)
            orijentacija=Pravci::Sjeveroistok;
        if(n==2)
            orijentacija=Pravci::Istok;
        if(n==3)
            orijentacija=Pravci::Jugoistok;
        if(n==4)
            orijentacija=Pravci::Jug;
        if(n==5)
            orijentacija=Pravci::Jugozapad;
        if(n==6)
            orijentacija=Pravci::Zapad;
        if(n==7)
            orijentacija=Pravci::Sjeverozapad;

return orijentacija;
}

int main()
{
    AktivirajRobota();
    string komanda;
    int n = 0, m = 0;

    Pravci orijentacija;
    cout<<"Robot je aktivan, nalazi se na poziciji (0,0) i gleda na
sjever" << endl;
    while(komanda != "K") {
        cout << "Unesi komandu: ";
        getline(cin,komanda);
        ocistiKomandu(komanda);
    }
}

```

```

        if(komanda[0]=='I' ) {
            if(aktivan == 0){
                IspisiPoziciju(x,y,ispisiOrijentaciju(n));
                continue;}
            if(validirajUnos(komanda)==false){
                KodoviGresaka kod_greske=KodoviGresaka::
NedostajeParametar;
                PrijaviGresku(kod_greske);
                continue;
            }
            ocistiKomandu(komanda);
            int integerbroj = stoi(uzmiKomandu(komanda));

            if (greska == true && negativanBroj == false) {
                KodoviGresaka kod_greske=KodoviGresaka::
NeispravanParametar;
                PrijaviGresku(kod_greske);
                continue;
            }

            Idi(x,y,ispisiOrijentaciju(n),integerbroj);
        } else {
            if (komanda == "A+"){
                AktivirajRobota();

                IspisiPoziciju(x,y,ispisiOrijentaciju(n));
                continue;
            }
            if (komanda == "A-"){
                DeaktivirajRobota();

                IspisiPoziciju(x,y,ispisiOrijentaciju(n));
continue;

            }
            if (komanda == "L") {
                if(aktivan == 0){
                    IspisiPoziciju(x,y,
ispisiOrijentaciju(n));
                    continue;}

                int *f;
                f=&n;
                if(n==0) {
                    n=7;

                } else {
                    n=*f-1;
                }

                IspisiPoziciju(x,y,ispisiOrijentaciju(n));
                continue;
            }

```

```

        }
        if (komanda == "D") {
            if(aktivan == 0){
                IspisiPoziciju(x,y,
ispisiOrijentaciju(n));
                continue;}
            Pravci orijentacija;

            int *f;
            f=&n;

            if(n==7) {
                n=0;

            } else {
                n=*f+1;

            }

            IspisiPoziciju(x,y, ispisiOrijentaciju(n));
            continue;
        }
    }
    if (komanda[0] != 'L' && komanda[0] != 'D' && komanda[0] !=
'I'&& komanda[0] != 'K' ) {
        KodoviGresaka kod_greske=KodoviGresaka::PogresnaKomanda;
        PrijaviGresku(kod_greske);

    }

    if (komanda[0] == 'L' || komanda[0] == 'D' || komanda[0] == 'K' &&
komanda[1] > 47 && komanda [1] < 123) {
        KodoviGresaka kod_greske=KodoviGresaka::SuvisanParametar;
        PrijaviGresku(kod_greske);
        if (komanda[0] == 'I' && komanda[1] == ' ') {
            KodoviGresaka kod_greske=KodoviGresaka::
NedostajeParametar;
            PrijaviGresku(kod_greske);

        }

    }

}

}
cout << "Dovidjenja!";
}

```