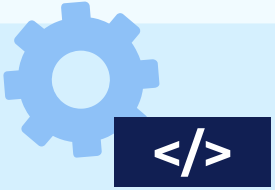
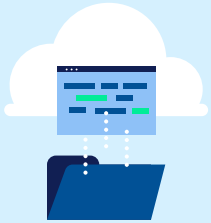
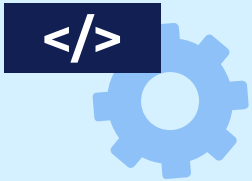
UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI
FAKULTET



TESTIRANJE BAZIRANO NA KONTROLI TOKA

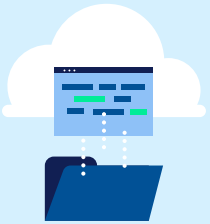


TESTIRANJE SOFTVERA



01

KONTROLA TOKA



TESTIRANJE BAZIRANO NA KONTROLI TOKA



- Kod tehnika crne kutije nije poznat način na koji program radi, već isključivo izlaz koji program proizvodi za određeni skup ulaznih podataka.
- Testiranje bazirano na kontroli toka je tehnika kojom se ispituje način na koji se program izvršava (tok izvršavanja programa).
- Ova tehnika se koristi kako bi se pokrio određeni deo programskog koda na osnovu odgovarajućeg **kriterijuma pokrivenosti**.
- Ona omogućava i da se otkriju logičke greške, mrtve grane i nedostižan kod u izvornom kodu programa.
- Može da se koristi i za procenu kvaliteta drugih napisanih testova.
- Pri projektovanju testova se koristi **graf toka kontrole**.



GRAF TOKA KONTROLE

- **Graf toka kontrole** (engl. *control flow graph*, CFG) je model kojim se vizuelno predstavlja struktura programskog koda i način na koji se on može izvršavati.

- Elementi grafa toka kontrole:



Čvor naredbe (bazični blok)



Čvor grananja



Čvor spajanja



Početni / završni čvor

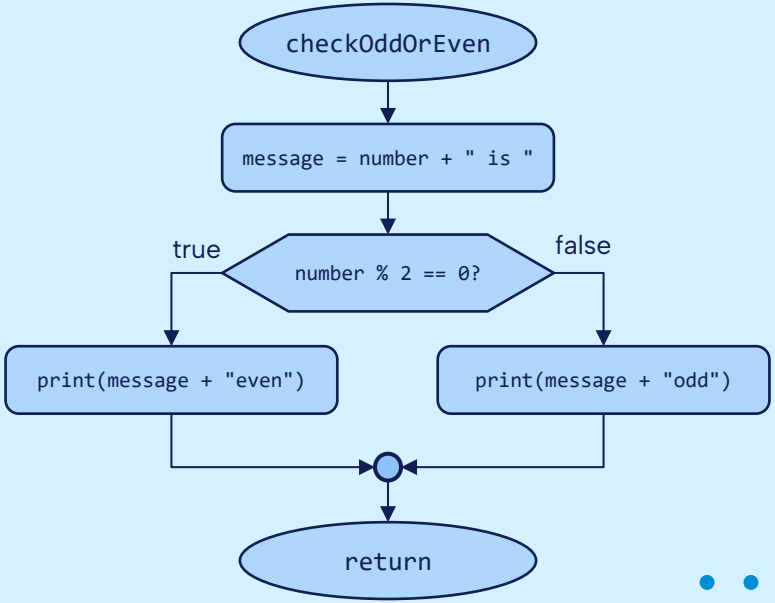
GRAF TOKA KONTROLE

PRIMER

- Primer grafa toka kontrole za funkciju koja određuje da li je broj paran ili neparan:

```
public static void checkOddOrEven(int number) {
    String message = number + " is ";

    if (number % 2 == 0) {
        System.out.println(message + "even");
    } else {
        System.out.println(message + "odd");
    }
}
```



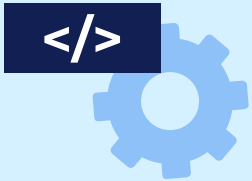
KRITERIJUMI POKRIVENOSTI

- Kriterijumi pokrivenosti programskog koda definišu koji deo programskog koda treba da bude izvršen prilikom testiranja:
  - **Pokrivenost iskaza** (engl. *statement coverage*)
    - Svaki iskaz u programu mora da bude izvršen u bar jednom test primeru.
  - **Pokrivenost odluka** (pokrivenost grana, engl. *decision coverage*)
    - Svaki ishod svake odluke mora da bude izvršen u bar jednom test primeru.
  - **Pokrivenost elementarnih uslova** (engl. *condition coverage*)
    - Svaki elementarni uslov svake odluke mora da bude ispunjen u bar jednom test primeru i neispunjen u bar jednom test primeru.
    - Elementarni uslov je najmanji deo uslova koji se ne može dalje rastaviti na jednostavnije uslove.

KRITERIJUMI POKRIVENOSTI

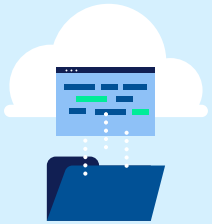
- Kriterijumi pokrivenosti programskog koda definišu koji deo programskog koda treba da bude izvršen prilikom testiranja:
  - **Pokrivenost višestrukih uslova** (engl. *multiple condition coverage*)
    - Svaka kombinacija vrednosti elementarnih uslova u svakoj odluci mora da se nađe u bar jednom test primeru.
  - **Modifikovana pokrivenost uslova i odluka** (engl. *modified condition/decision coverage,* MC/DC)
    - Obuhvata pokrivenost elementarnih uslova i odluka, uz ograničenje da za svaki elementarni uslov mora da postoji par test primera koji pokazuje da se izlaz testa menja ako se promeni vrednost isključivo posmatranog uslova.
  - **Pokrivenost putanja** (engl. *path coverage*)
    - Svaka putanja u programu mora da se nađe u bar jednom test primeru.





02

ZADACI



ZADATAK 1 – OSIGURANJE

POSTAVKA (1)

- Data je implementacija funkcije **`calculateInsurancePremium`** u programskom jeziku Java koja određuje cenu osiguranja na osnovu podataka o osiguraniku. Funkcija prihvata uzrast (ceo broj), pol (karakter ‘M’ ili ‘F’) i bračni status (logička vrednost) osiguranika, a ispisuje cenu osiguranja izraženu u dolarima.

```
public static void calculateInsurancePremium(int age, char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```



ZADATAK 1 – OSIGURANJE

POSTAVKA (2)

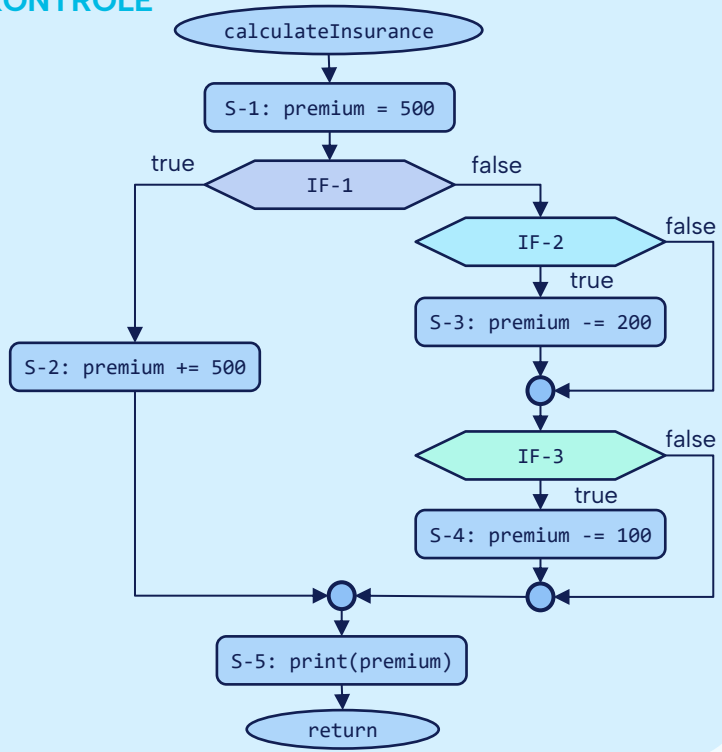
1. Konstruisati graf toka kontrole za datu funkciju.
2. Napisati minimalni skup test primera kojim će biti pokriveni svi iskazi u funkciji. Za svaki test primer jasno naznačiti ulazne podatke, očekivani izlaz i pokrivene iskaze.
3. Napisati minimalni skup test primera kojim će biti pokrivene sve odluke u funkciji.
4. Napisati minimalni skup test primera kojim će biti pokriveni svi elementarni uslovi u funkciji.
5. Napisati minimalni skup test primera kojim će biti pokriveni svi višestruki uslovi u funkciji.
6. Napisati minimalni skup test primera kojim će biti pokriveni svi uslovi i odluke u funkciji korišćenjem MC/DC kriterijuma.



ZADATAK 1 – OSIGURANJE

REŠENJE – GRAF TOKA KONTROLE

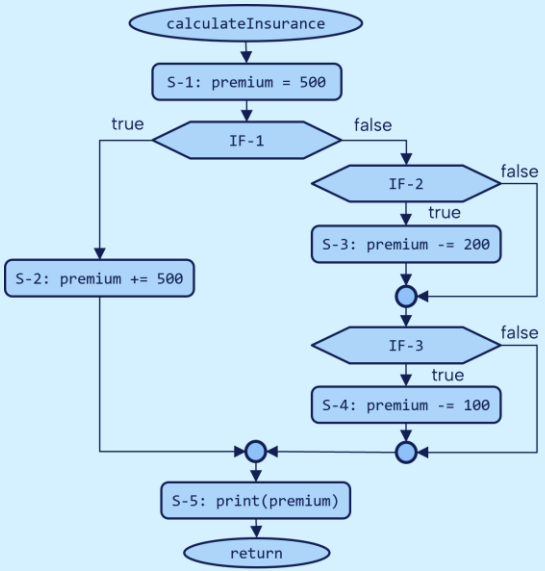
```
public static void calculateInsurancePremium(int age,
char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ISKAZA (1)

- Svaki iskaz (naredba ili provera uslova) mora da se nađe u jednom test primeru.
- Da li je moguće pokriti sve iskaze jednim test primerom?
  - Ne, jer je izvršavanje iskaza S-2 međusobno isključivo sa izvršavanjem iskaza S-3 i S-4.
  - Koliko je test primera potrebno?

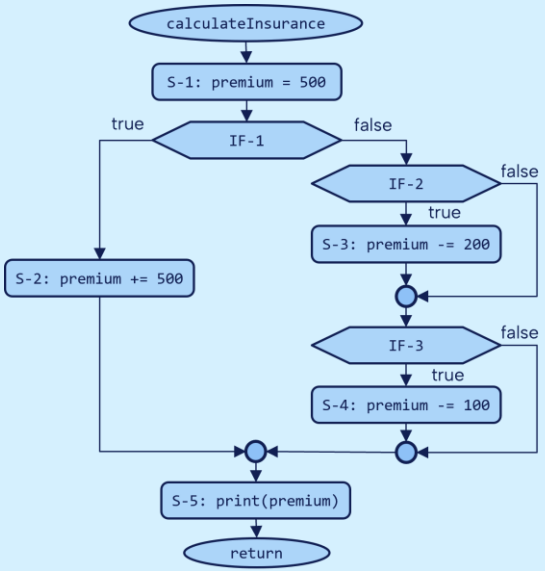


ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ISKAZA (2)

- Test primeri:

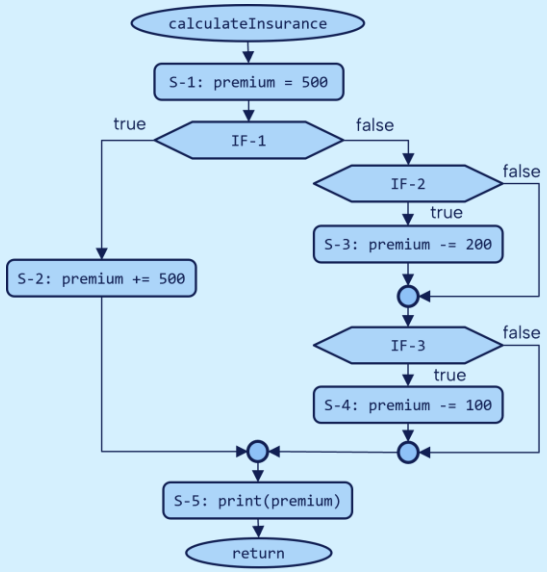
TP	Ulazni podaci			Očekivani izlaz	Pokriveni iskazi
	Uzrast	Pol	Bračni status	Cena osiguranja	
1	21	M	False	1000	S–1, IF–1, S–2, S–5
2	53	F	True	200	S–1, IF–1, IF–2, S–3, IF–3, S–4, S–5



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA (1)

- Svaka grana svake odluke mora da se nađe u barem jednom test primeru.
- U datoj funkciji postoje tri provere uslova, odnosno tri odluke.
- Da li je neophodno šest test primera u tom slučaju?
  - Ne, jer više provera uslova može da se pokrije jednim test primerom.
  - Koliko je test primera potrebno?

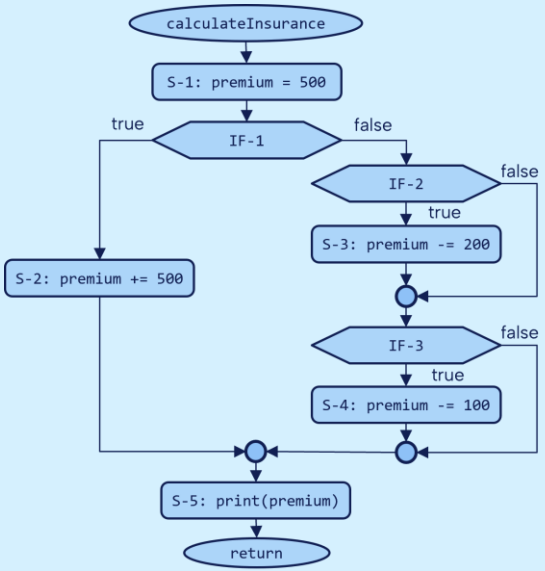


ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA (2)

- Test primeri:

TP	Ulazni podaci			Očekivani izlaz	Pokrivene odluke
	Uzrast	Pol	Bračni status	Cena osiguranja	
1	21	M	False	1000	IF-1 True
2	23	F	False	300	IF-1 False, IF-2 True, IF-3 False
3	50	M	False	400	IF-1 False, IF-2 False, IF-3 True



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ELEMENTARNIH USLOVA (1)

- Svaki elementarni uslov mora da bude ispunjen u bar jednom i neispunjen u bar jednom test primeru.
- Koji su to elementarni uslovi u funkciji?

(IF-1, C-1)	`age < 25`
(IF-1, C-2)	`gender == 'M'`
(IF-1, C-3)	`!married`
(IF-2, C-4)	`married`
(IF-2, C-5)	`gender == 'F'`
(IF-3, C-6)	`age > 45`
(IF-3, C-7)	`age < 65`

```
public static void calculateInsurancePremium(int
age, char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ELEMENTARNIH USLOVA (2)

- Dakle, potrebno je pokriti sledeće vrednosti ulaznih parametara:

(C-1 True) `age < 25`	**(C-1 False)** `age >= 25`
(C-2 True) `gender == 'M'`	**(C-2 False)** `gender == 'F'`
(C-3 True) `married == false`	**(C-3 False)** `married == true`
(C-4 True) `married == true`	**(C-4 False)** `married == false`
(C-5 True) `gender == 'F'`	**(C-5 False)** `gender == 'M'`
(C-6 True) `age > 45`	**(C-6 False)** `age <= 45`
(C-7 True) `age < 65`	**(C-7 False)** `age >= 65`

```
public static void calculateInsurancePremium(int
age, char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ELEMENTARNIH USLOVA (3)

- Test primeri:

TP	Ulazni podaci			Očekivani izlaz	Pokriveni uslovi
	Uzrast	Pol	Bračni status	Cena osiguranja	
1	23	F	False	300	C-1 True, C-3 True, C-5 True, C-7 True, C-2 False, C-4 False, C-6 False
2	30	M	True	300	C-2 True, C-4 True, C-7 True, C-1 False, C-3 False, C-5 False, C-6 False
3	70	F	False	300	C-3 True, C-5 True, C-6 True, C-1 False, C-2 False, C-4 False, C-7 False

(C-1 True) `age < 25`
(C-2 True) `gender == 'M'`
(C-3 True) `married == false`
(C-4 True) `married == true`
(C-5 True) `gender == 'F'`
(C-6 True) `age > 45`
(C-7 True) `age < 65`
(C-1 False) `age >= 25`
(C-2 False) `gender == 'F'`
(C-3 False) `married == true`
(C-4 False) `married == false`
(C-5 False) `gender == 'M'`
(C-6 False) `age <= 45`
(C-7 False) `age >= 65`

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (1)

- Svaka kombinacija vrednosti elementarnih uslova u svakoj odluci mora da se nađe u bar jednom test primeru.

- Koji su to uslovi za koje se traže kombinacije vrednosti?

  (IF-1) `age < 25 && gender == 'M' && !married`

  (IF-2) `married || gender == 'F'`

  (IF-3) `age > 45 && age < 65`

```
public static void calculateInsurancePremium(int
age, char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (2)

- Kombinacije uslova i odgovarajuće vrednosti parametara na osnovu njih za prvu odluku:

age < 25	gender == 'M'	!married
True	True	True
True	True	False
True	False	True
True	False	False
False	True	True
False	True	False
False	False	True
False	False	False

age	gender	married
< 25	M	False
< 25	M	True
< 25	F	False
< 25	F	True
>= 25	M	False
>= 25	M	True
>= 25	F	False
>= 25	F	True



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (3)

- Kombinacije uslova i odgovarajuće vrednosti parametara na osnovu njih za drugu odluku:

married	gender == 'F'
True	True
True	False
False	True
False	False

age	married	gender
*	True	F
*	True	M
*	False	F
*	False	M



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (4)

- Kombinacije uslova i odgovarajuće vrednosti parametara na osnovu njih za treću odluku:

age > 45	age < 65
True	True
True	False
False	True
False	False

age	married	gender
age > 45, age < 65	*	*
age > 45, age >= 65	*	*
age <= 45, age < 65	*	*
age <= 45, age >= 65	*	*



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (5)

- Objedinjene kombinacije vrednosti ulaznih parametara za sve odluke u funkciji:

C	age	gender	married
1	< 25	M	False
2	< 25	M	True
3	< 25	F	False
4	< 25	F	True
5	>= 25	M	False
6	>= 25	M	True
7	>= 25	F	False
8	>= 25	F	True

C	age	gender	married
9	*	F	True
10	*	M	True
11	*	F	False
12	*	M	False
13	age > 45, age < 65	*	*
14	age > 45, age >= 65	*	*
15	age <= 45, age < 65	*	*
16	age <= 45, age >= 65	*	*

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (6)

- Neka ograničenja je moguće pojednostaviti:

C	age	gender	married
...			
13	45 < age < 65	*	*
14	age >= 65	*	*
15	age <= 45	*	*
16	age <= 45, age >= 65	*	*

- Da li postoji problem sa poslednjom kombinacijom vrednosti?



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (7)

- Test primeri:

TP	Ulazni podaci			Očekivani izlaz	Pokriveni višestruki uslovi
	Uzrast	Pol	Bračni status	Cena osiguranja	
1	23	M	False	1000	C-1, C-12
2	23	M	True	300	C-2, C-10
3	23	F	False	300	C-3, C-11
4	23	F	True	300	C-4, C-9
5	30	M	False	500	C-5, C-15
6	70	M	True	300	C-6, C-14
7	50	F	False	200	C-7, C-13
8	30	F	True	200	C-8

C	age	gender	married
1	< 25	M	False
2	< 25	M	True
3	< 25	F	False
4	< 25	F	True
5	>= 25	M	False
6	>= 25	M	True
7	>= 25	F	False
8	>= 25	F	True
9	*	F	True
10	*	M	True
11	*	F	False
12	*	M	False
13	45 < age < 65	*	*
14	age >= 65	*	*
15	age <= 45	*	*
16	age <= 45, age >= 65	*	*

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (1)

- Svaka odluka i elementarni uslov moraju da budu pokriveni uz ograničenje da se za svaki elementarni uslov pronađe par test primera koji pokazuje da izmena ispunjenosti tog elementarnog uslova direktno uzrokuje promenu ishoda, bez promene ostalih elementarnih uslova.

- Funkcija sadrži tri odluke, tako da svaki elementarni uslov u okviru njih treba pronaći odgovarajući skup test primera koji ovo svojstvo demonstrira.

```
public static void calculateInsurancePremium(int
age, char gender, boolean married) {
    int premium = 500;
    if (age < 25 && gender == 'M' && !married) {
        premium += 500;
    }
    else {
        if (married || gender == 'F')
            premium -= 200;
        if (age > 45 && age < 65)
            premium -= 100;
    }
    System.out.println(premium);
}
```

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (2)

- Prvi složeni uslov je: **`age < 25 && gender == 'M' && !married`**
- Pošto su elementarni uslovi povezani operatorom „i“, neispunjenost bilo kog od tih elementarnih uslova uzrokuje da i čitav složen uslov bude neispunjen.
- Osnovni slučaj može biti da su svi elementarni uslovi ispunjeni, kada je i čitav složeni uslov ispunjen.

`age < 25`	`gender == 'M'`	`!married`	**`!shod`**
True	True	True	True

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (3)

- Prva i druga kombinacija pokazuju da promena trećeg uslova utiče na promenu ishoda.
- Prva i treća kombinacija pokazuju da promena drugog uslova utiče na promenu ishoda.
- Prva i četvrta kombinacija pokazuju da promena prvog uslova utiče na promenu ishoda.

age < 25	gender == 'M'	!married	Ishod
True	True	True	True
True	True	**False**	**False**
True	**False**	True	**False**
False	True	True	**False**

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (4)

- Na osnovu dobijenih kombinacija ispunjenosti uslova u prvoj odluci, dobijaju se sledeće kombinacije vrednosti ulaznih parametara:

age < 25	gender == 'M'	!married
True	True	True
True	True	False
True	False	True
False	True	True

age	gender	married
age < 25	M	False
age < 25	M	True
age < 25	F	False
age >= 25	M	False



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (5)

- Drugi složeni uslov je: **`married || gender == 'F'`**
- Pošto su elementarni uslovi povezani operatorom „ili“, ispunjenost bilo kog od tih elementarnih uslova uzrokuje da i čitav složen uslov bude ispunjen.
- Osnovni slučaj može biti da su svi elementarni uslovi neispunjeni, kada je i čitav složeni uslov neispunjen.

`married`	`gender == 'F'`	Ishod
False	False	False
True	False	**True**
False	**True**	**True**

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (6)

- Na osnovu dobijenih kombinacija ispunjenosti uslova u drugoj odluci, dobijaju se sledeće kombinacije vrednosti ulaznih parametara:

married	gender == 'F'
False	False
True	False
False	True

age	gender	married
*	M	False
*	M	True
*	F	False



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (7)

- Treći složeni uslov je: **age > 45 && age < 65**
- Slično kao u prvom složenom uslovu, pošto su elementarni uslovi povezani operatorom „i“, neispunjenost bilo kog od tih elementarnih uslova uzrokuje da i čitav složen uslov bude neispunjen.

age > 45	age < 65	Ishod
True	True	True
True	**False**	**False**
False	True	**False**



ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (8)

- Na osnovu dobijenih kombinacija ispunjenosti uslova u trećoj odluci, dobijaju se sledeće kombinacije vrednosti ulaznih parametara:

age > 45	age < 65
True	True
True	False
False	True

age	gender	married
age > 45, age < 65	*	*
age > 45, age >= 65	*	*
age <= 45, age < 65	*	*

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (9)

- Objedinjene kombinacije vrednosti parametara za sve odluke u funkciji:

C	age	gender	married
1	age < 25	M	False
2	age < 25	M	True
3	age < 25	F	False
4	age >= 25	M	False
5	*	M	False
6	*	M	True
7	*	F	False
8	age > 45, age < 65	*	*
9	age > 45, age >= 65	*	*
10	age <= 45, age < 65	*	*

ZADATAK 1 – OSIGURANJE

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (10)

- Test primeri:

TP	Ulazni podaci			Očekivani izlaz	Pokrivene kombinacije uslova
	Uzrast	Pol	Bračni status	Cena osiguranja	
1	20	M	False	1000	C-1, C-5, C-10
2	23	M	True	300	C-2, C-6
3	19	F	False	300	C-3, C-7
4	57	M	False	400	C-4, C-8
5	72	F	True	300	C-9

C	age	gender	married
1	age < 25	M	False
2	age < 25	M	True
3	age < 25	F	False
4	age >= 25	M	False
5	*	M	False
6	*	M	True
7	*	F	False
8	45 < age < 65	*	*
9	age >= 65	*	*
10	age <= 45	*	*

- Da li su pokrivene sve odluke i elementarni uslovi?

ZADATAK 2 – ROLERKOSTER

POSTAVKA (1)

- Data je implementacija funkcije **canEnterRide** u programskom jeziku Java koja određuje da li posetilac luna parka sme da se vozi na najvećem rolerkosteru u parku. Funkcija prihvata uzrast posetioca (ceo broj) i informaciju da li posetilac ima dozvolu roditelja da se vozi (logička vrednost), a vraća logičku vrednost koja označava da li je posetiocu dozvoljeno da se vozi rolerkosterom ili ne.

```
public static boolean canEnterRide(int age, boolean parentalConsent) {
    if (age >= 18 || (age < 18 && parentalConsent)) {
        return true;
    }
    else {
        return false;
    }
}
```



ZADATAK 2 – ROLERKOSTER

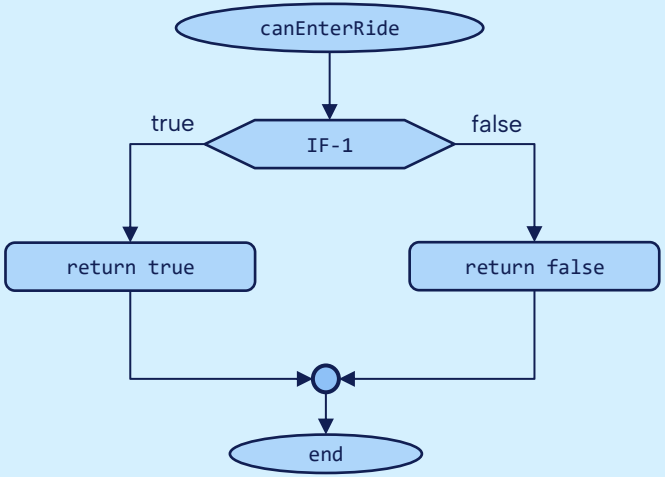
POSTAVKA (2)

1. Konstruisati graf toka kontrole za datu funkciju.
2. Napisati minimalni skup test primera kojim će biti pokrivene sve odluke u funkciji. Da li pokrivanje svih odluka garantuje pokrivenost svih elementarnih uslova u testiranom programu?
3. Napisati minimalni skup test primera kojim će biti pokriveni svi višestruki uslovi u funkciji. Ukoliko nije moguće u potpunosti zadovoljiti ovaj kriterijum pokrivenosti, obrazložiti zašto i odrediti procentualnu pokrivenost višestrukih uslova za projektovani skup test primera.
4. Odrediti procentualnu pokrivenost iskaza, odluka i elementarnih uslova za projektovani skup test primera u tački 3. Obrazložiti odgovor.
5. Napisati minimalni skup test primera kojim će biti pokriveni svi uslovi i odluke u funkciji korišćenjem MC/DC kriterijuma.

ZADATAK 2 – ROLERKOSTER

REŠENJE – GRAF TOKA KONTROLE

```
public static boolean canEnterRide(int age, boolean
parentalConsent) {
    if (age >= 18 || (age < 18 && parentalConsent)) {
        return true;
    }
    else {
        return false;
    }
}
```



ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST ODLUKA (1)

- Test primeri:

TP	Ulazni podaci		Očekivani izlaz	Pokrivene odluke
	Uzrast	Dozvola roditelja	Dozvola za vožnju	
1	24	False	True	IF-1 True
2	16	False	False	IF-1 False

- Da li pokrivenost odluka garantuje pokrivenost svih elementarnih uslova?
  - Ne, u napisanim test primerima ne postoji vrednost *True* za dozvolu roditelja.
  - Da li obrnuta relacija važi?



ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (1)

- Prvi i jedini složeni uslov u funkciji je: **`age >= 18 || (age < 18 && parentalConsent)`**
- Kombinacije uslova i odgovarajuće vrednosti parametara na osnovu njih su:

`age >= 18`	`age < 18`	`parentalConsent`
True	True	True
True	True	False
True	False	True
True	False	False
False	True	True
False	True	False
False	False	True
False	False	False

`age`	`parentalConsent`
age >= 18, age < 18	True
age >= 18, age < 18	False
age >= 18	True
age >= 18	False
age < 18	True
age < 18	False
age < 18, age >= 18	True
age < 18, age >= 18	False

ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (2)

- Da li je moguće pokriti sve dobijene kombinacije vrednosti?

C	age	parentalConsent
1	**age >= 18, age < 18**	True
2	**age >= 18, age < 18**	False
3	age >= 18	True
4	age >= 18	False
5	age < 18	True
6	age < 18	False
7	**age < 18, age >= 18**	True
8	**age < 18, age >= 18**	False



ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST VIŠESTRUKIH USLOVA (3)

- Test primeri:

TP	Ulazni podaci		Očekivani izlaz	Pokriveni višestruki uslovi
	Uzrast	Dozvola roditelja	Dozvola za vožnju	
1	24	True	True	C-3
2	24	False	True	C-4
3	16	True	True	C-5
4	16	False	False	C-6

- Procentualna pokrivenost **svih** višestrukih uslova je 4 / 8, odnosno 50%.
- Procentualna pokrivenost **ostvarivih** višestrukih uslova je 4 / 4, odnosno 100%.

ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST ISKAZA, ODLUKA I USLOVA (1)

TP	Ulazni podaci		Očekivani izlaz
	Uzrast	Dozvola roditelja	Dozvola za vožnju
1	24	True	True
2	24	False	True
3	16	True	True
4	16	False	False

```
public static boolean canEnterRide(int age, boolean
parentalConsent) {
    if (age >= 18 || (age < 18 && parentalConsent)) {
        return true;
    }
    else {
        return false;
    }
}
```

- Procentualna pokrivenost iskaza je 3 / 3, odnosno 100%.
- Procentualna pokrivenost odluka je 2 / 2, odnosno 100%.
- Procentualna pokrivenost elementarnih uslova je 6 / 6, odnosno 100%.
- Da li pokrivenost višestrukih uslova garantuje pokrivenost iskaza, odluka i elementarnih uslova?

ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (1)

- Prvi i jedini složeni uslov u funkciji je: **`age >= 18 || (age < 18 && parentalConsent)`**
- Prva i druga kombinacija pokazuju da promena prvog uslova utiče na promenu ishoda.
- Prva i četvrta kombinacija pokazuju da promena drugog uslova utiče na promenu ishoda.
- Treća i četvrta kombinacija pokazuju da promena trećeg uslova utiče na promenu ishoda.

`age >= 18`	`age < 18`	`parentalConsent`	Ishod
False	False	True	False
True	False	True	True
False	True	False	False
False	True	True	True

ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (2)

- Da li postoji problem sa dobijenim kombinacijama?
- Na osnovu dobijenih kombinacija ispunjenosti uslova, dobijaju se sledeće kombinacije vrednosti ulaznih parametara:

age >= 18	age < 18	parentalConsent
False	False	True
True	False	True
False	True	False
False	True	True

age	parentalConsent
age < 18, age >= 18	True
age >= 18	True
age < 18	False
age < 18	True

ZADATAK 2 – ROLERKOSTER

REŠENJE – POKRIVENOST ODLUKA I USLOVA MC/DC (3)

- Test primeri:

TP	Ulazni podaci		Očekivani izlaz	Pokrivene kombinacije uslova
	Uzrast	Dozvola roditelja	Dozvola za vožnju	
1	24	True	True	C-2
2	16	False	False	C-3
3	16	True	True	C-4

C	age	parentalConsent
1	age < 18, age >= 18	True
2	age >= 18	True
3	age < 18	False
4	age < 18	True

- MC/DC kriterijum pokrivenosti nije moguće zadovoljiti u potpunosti zbog međusobne zavisnosti elementarnih uslova **age >= 18** i **age < 18**.

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI
FAKULTET

PITANJA?

https://si3ts.etf.bg.ac.rs/

nikolas@etf.bg.ac.rs

