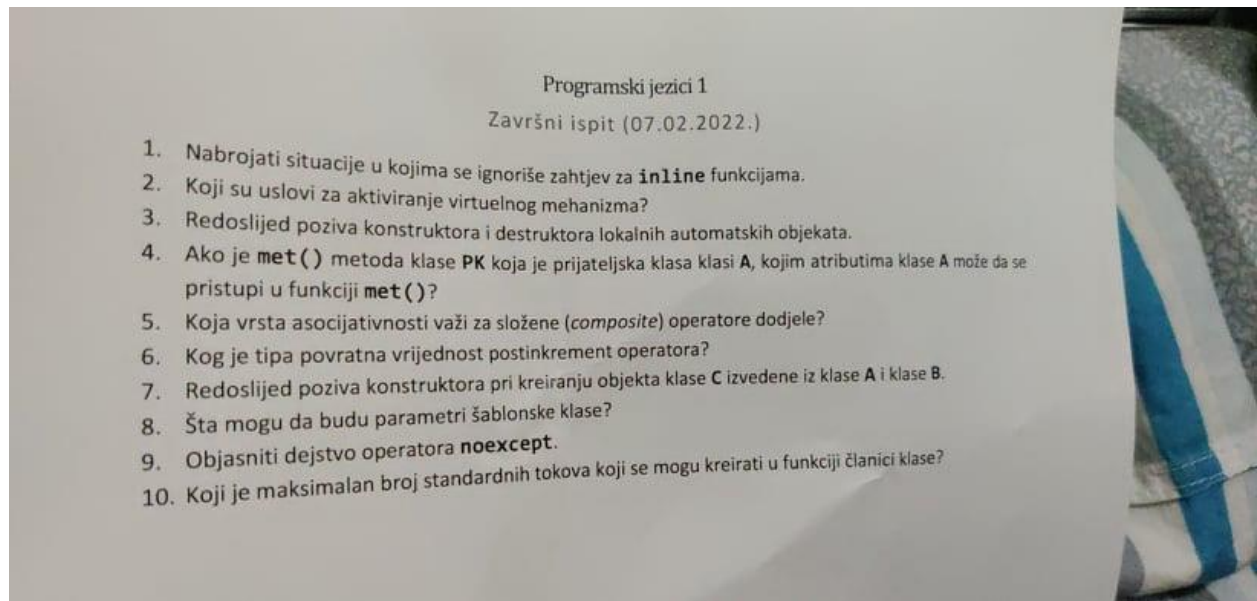


## PJ1 USMENI DRUGI DIO

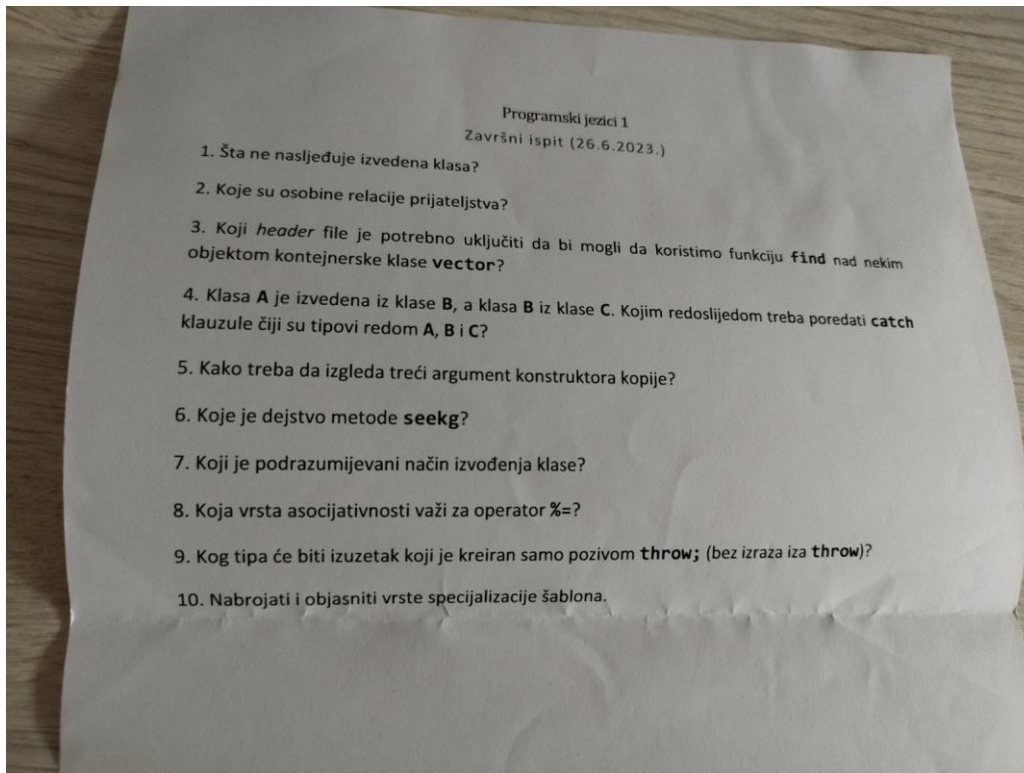


1. Postoje više situacija u kojima se ignoriše taj zahtjev:
  - Ukoliko funkcija sadrži neku **petlju** (for, while, do-while)
  - Ukoliko funkcija **nije** tipa **void** i nije napisana naredba za vraćanje vrijednosti (**return**)
  - Ako je **tijelo** funkcije **suviše dugačko**
  - Ukoliko u funkciji postoji **switch** ili **goto**
  - Ako je funkcija **rekurzivna**
  - Ako funkcija sardrži neke **statičke promjenljive**
2. Objektu se pristupa preko pokazivaca ili reference.
3. Redoslijed poziva konstruktora i destruktora za automatske lokalne objekte je određen njihovim životnim vijekom, odnosno opsegom u kojem su definisani. Kada se automatski lokalni objekat deklarise unutar nekog opsega, konstruktor će se pozvati pri ulasku u taj opseg, dok će destructor biti pozvan pri izlasku iz opsega.
4. Može da pristupi privatnim i zasticenim članovima.
5. RTL (right-to-left)
6. Postdekrement operator treba da sačuva **kopiju** svog operanda, izmjeni trenutni operand, a zatim vrati kopiju tog operanda.
7. Prvo se izvršava konstruktor od klase A pa od B pa od C.
8. Parametri šablonske klase mogu da budu:
  - Tipovi – **class**, **typename**, **lista\_parametara**
  - Vrijednosti – ne mogu da budu floating point, ni korisnički definisanog tipa. Moraju biti globalni, kako bi bili poznati u trenutku kompajliranja. Ti vrijednosni parametri su:
    - Cjelobrojni tip
    - Enumeratori
    - Reference
    - Pokazivač na funkciju
    - Pokazivač na objekat

- Pokazivač na članove
- Drugi šabloni, tj. šabloni mogu da budu parametri šablona klase

9. On ima operat (noexcept(izraz)), rezultat je logickog tipa (tačno ovako ne može da baci izuzetak) i provjerava se da li bi moglo doći do izuzetka u izrazu. Izraz je proizvoljnog tipa čak i void, ne izračunava se samo se provjerava u fazi prevodjenja.

10. postoje 4 standardna toka (cin, cout, cerr i clog) i ova 4 standardna toka se mogu koristiti neograničen broj puta



#### ODGOVORI:

1. Izvedena klasa ne nasljeđuje konstruktor i destruktor osnovne klase i ne nasljeđuje operator dodojela `'='` osnovne klase.
2. Osobine relacije prijateljstva su da se ne nasljeđuje, nije simetrična ni tranzitivna.
3. Potrebno je uključiti header file `<algorithm>`.
4. Što znači, generalno se preporučuje da se `catch` klauzule poredaju od najspecifičnijeg do najopštijeg tipa izuzetka. U ovom konkretnom slučaju, **redoslijed** `catch` klauzula trebao bi biti **A, B, C**. Razlog za ovakav redoslijed je da `catch` blokovi obrade izuzetaka trebaju biti poredani tako da najopštiji tip izuzetka bude na kraju (tip za osnovnu/baznu klasu), jer će se prvo provjeriti klauzule koje se odnose na specifičnije tipove izuzetaka (izvedene klase). Ako bi redoslijed bio suprotan, tj. `C, B, A`, `catch` klauzula za izuzetak tipa `C` bi uhvatila sve izuzetke, uključujući one koji pripadaju tipovima `B` i `A`, te `catch` klauzule za tipove `B` i `A` ne bi bile **nikada dosegnute**.
5. Konstruktor kopije može da ima jedan argument (konstantna referenca na drugi objekat istog tipa ili referenca na konstantni objekat istog tipa) ili 2 i više argumenata (u tom slučaju 2. ili  $n$ -ti argument **mora imati podrazumijevanu vrijednost**).

6. Metoda **seekg**(long pozicija) se koristi za postavljanje pozicije čitanja u ulaznu datoteku za broj određen argumentom 'pozicija'. Odnosno, čitanje će krenuti od onog bajta u datoteci određen promjenljivom 'pozicija'. Drugim riječima, na tu poziciju se postavlja vrijednost pokazivača koja će biti pročitana sljedeća.

7. Podrazumijevani način izvođenja klase je private.

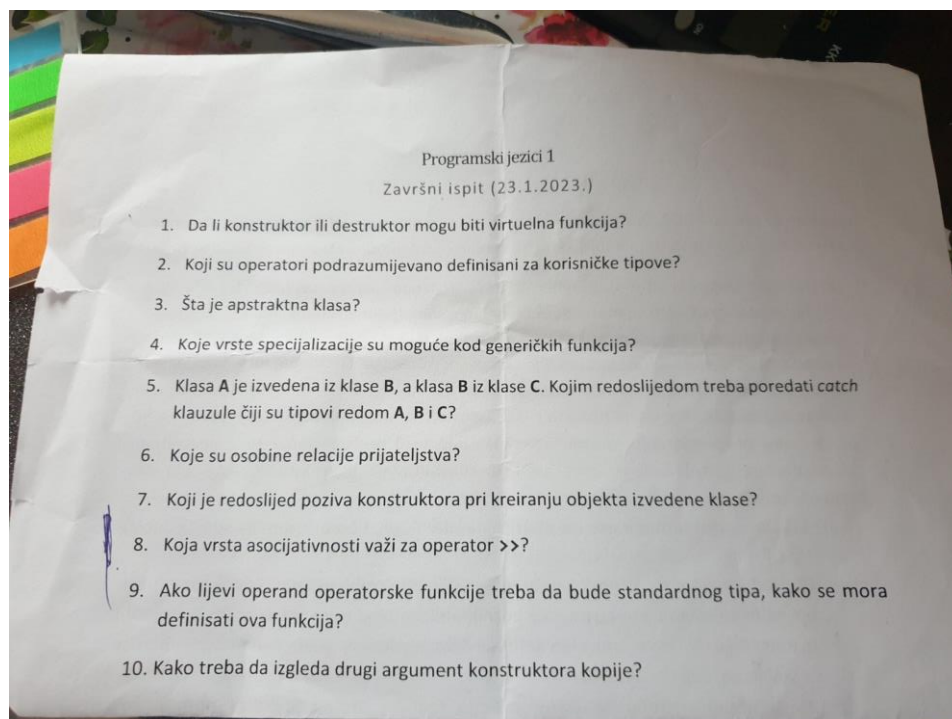
8. RTL (right-to-left).

9. Kada se koristi samo **throw**; bez izraza iza njega, tada će se trenutni izuzetak ponovno baciti dalje bez promjene njegovog tipa. Drugim riječima, tip izuzetka će ostati isti kao i tip izuzetka koji je bio trenutno aktiviran prilikom tog **throw**; izraza (odnosno, biće propagiran dalje sa istim tipom)

10. Postoje dvije vrste specijalizacije šablona:

- **Djelimična specijalizacija** – šablon ima bar jedan parametar, pa se može generisati više klasa (instanci) iz specijalizovanog šablona.

- **Potpuna specijalizacija** – šablon nema ni jedan parametar, ali se može generisati samo JEDNA instanca šablona (samo jedna klasa npr). Funkcije mogu imati samo ovu specijalizaciju!



#### ODGOVORI:

1. Konstruktor i destructor mogu biti virtualne funkcije.

2. '=', '&', ','.

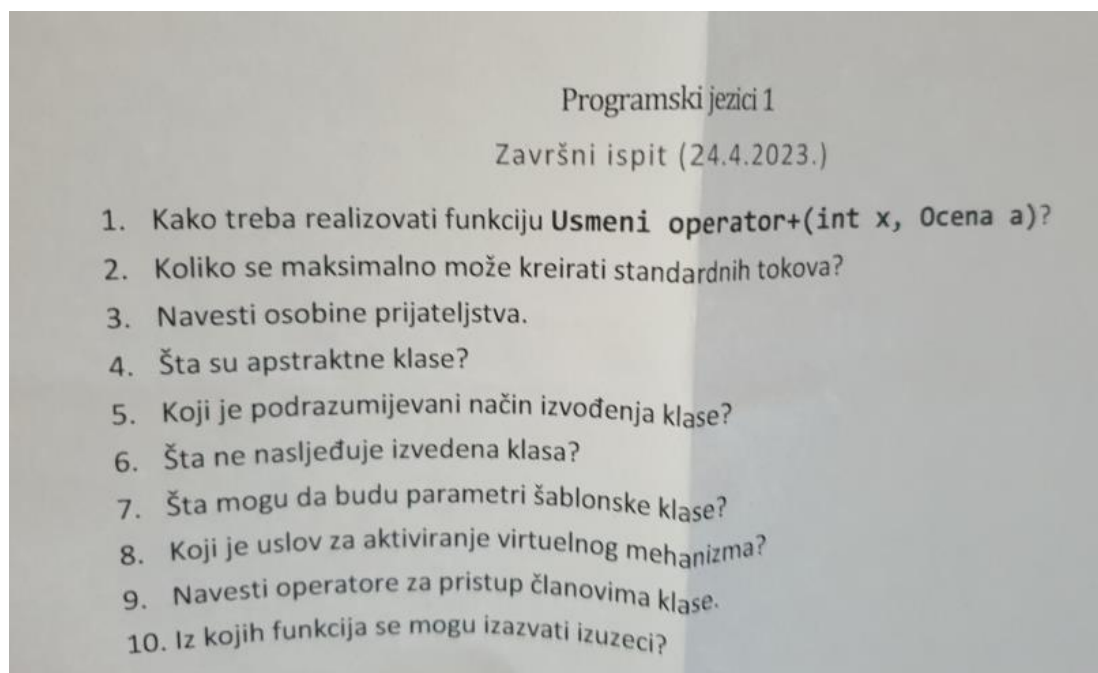
3. Apstraktna klasa je klasa koja ima bar jednu čistu virtualnu metodu. Čisto virtualna metoda je funkcija članica koja je deklarirana sa = 0 na kraju svog potpisa, nema tijelo.

4. Generičke funkcije se mogu samo **potpuno specijalizovati**.

5. Što znači, generalno se preporučuje da se catch klauzule poredaju od najspecifičnijeg do najopštijeg tipa izuzetka. U ovom konkretnom slučaju, **redoslijed** catch klauzula trebao bi biti **A, B, C**. Razlog za ovakav redoslijed je da catch blokovi obrade izuzetaka trebaju biti poredani tako da najopštiji tip izuzetka bude na kraju (tip za osnovnu/baznu klasu), jer će se prvo provjeriti klauzule koje se odnose na

specifičnije tipove izuzetaka (izvedene klase). Ako bi redoslijed bio suprotan, tj. C, B, A, catch klauzula za izuzetak tipa C bi uhvatila sve izuzetke, uključujući one koji pripadaju tipovima B i A, te catch klauzule za tipove B i A ne bi bile **nikada dosegnute**.

6. Osobine relacije prijateljstva su da se ne nasljedjuje, nije simetricna ni tranzitivna.
7. Prvo se poziva konstruktor osnovne klase, onda se inicijalizuju podaci clanovi pozivanjem njihovih konstruktora nakraju se izvrsava tijelo konstruktora izvedene klase.
8. LTR (left-to-right).
9. Ako lijevi operand operatorske funkcije treba da bude standardnog tipa (npr. **int**, **double**, **char**, itd), operatorska funkcija se mora definisati kao **globalna funkcija** ili kao **funkcija članica koja je deklarirana kao prijateljska funkcija** u klasi.
10. Drugi argument konstruktora kopije je podrazumijevana vrijednost.

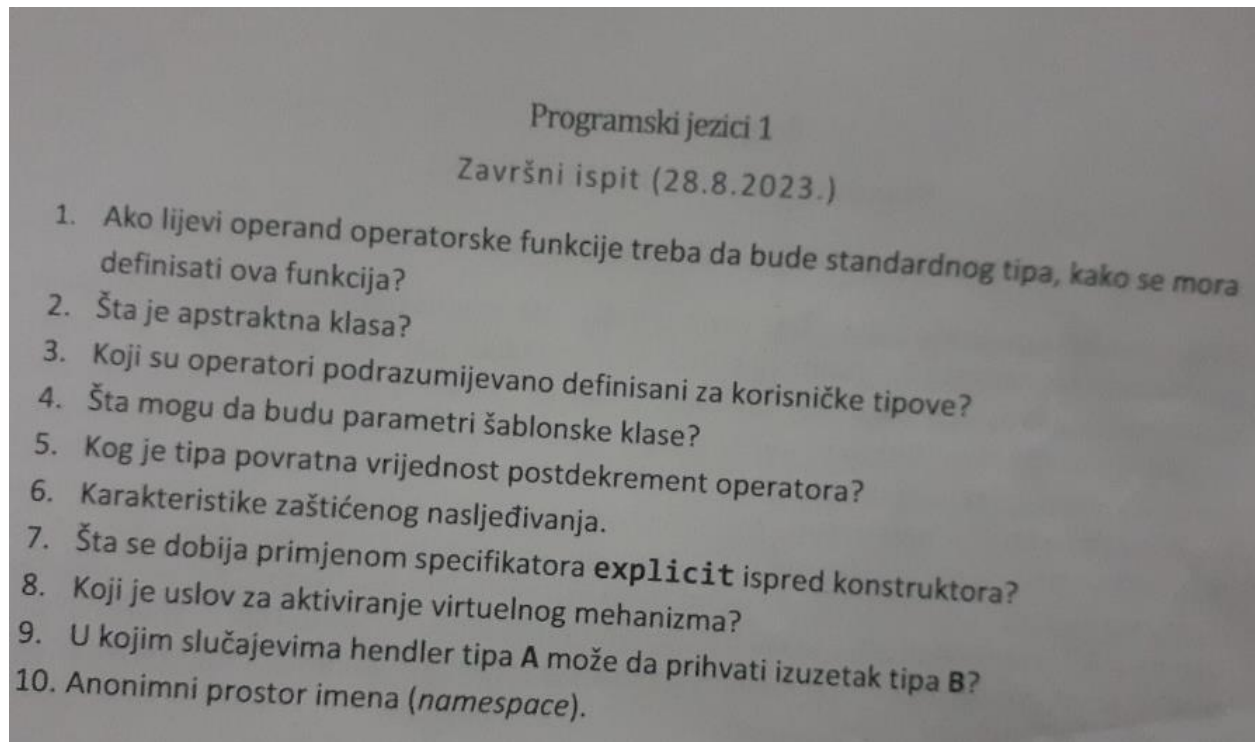


#### ODGOVORI:

1. Funkcija Usmeni nije dobro potpisana može se realizovati na dva načina:  
kao globalna: `Usmeni operator+(const Usmeni& lhs, const Usmeni& rhs);`  
kao funkcija članica: `Usmeni operator+(const Usmeni& rhs);`
2. postoje 4 standardna toka (`cin`, `cout`, `cerr` i `clog`) i ova 4 standardna toka se mogu koristiti neograničen broj puta
3. Ne nasljedjuje se, nije simetricna ni tranzitivna.
4. Apstrakte klase su one klase koje sadrže bar jednu čisto virtuelnu funkciju članicu koja je deklarirana sa `=0` na kraju svog potpisa i nema tijelo.
5. Podrazumijevani način izvođenja klase je `private`.
6. Izvedena klasa ne nasljedjuje konstruktor i destructor osnovne klase i operator dodjele `'='` osnovne klase.
7. Parametri šablonske klase mogu da budu:
  - Tipovi – **class**, **typename**, **lista\_parametara**
  - Vrijednosti – ne mogu da budu floating point, ni korisnički definisanog tipa. Moraju biti

globalni, kako bi bili poznati u trenutku kompajliranja. Ti vrijednosni parametri su:

- Cjelobrojni tip
  - Enumeratori
  - Reference
  - Pokazivač na funkciju
  - Pokazivač na objekat
  - Pokazivač na članove
  - Drugi šabloni, tj. šabloni mogu da budu parametri šablona klase
8. Objektu se pristupa preko pokazivaca ili reference.
9. . i ->, tacka i strelica.
10. Članice klase, operatorske funkcije, konstruktori i destruktori.



#### ODGOVORI:

1. Treba se definisati kao globalna funkcija ili kao funkcija članica deklarirana kao prijateljska.
2. Apstraktna klasa je ona klasa koja sadrži bar jednu čisto virtuelnu metodu, čisto virtuelna metoda je onda metoda koja je deklarirana sa =0 na kraju svog potpisa i nema tijelo.
3. '=', '&', ' '.
4. Parametri šablonske klase mogu da budu:
  - Tipovi – **class**, **typename**, **lista\_parametara**
  - Vrijednosti – ne mogu da budu floating point, ni korisnički definisanog tipa. Moraju biti globalni, kako bi bili poznati u trenutku kompajliranja. Ti vrijednosni parametri su:
    - Cjelobrojni tip
    - Enumeratori
    - Reference
    - Pokazivač na funkciju
    - Pokazivač na objekat

- Pokazivač na članove

- Drugi šabloni, tj. šabloni mogu da budu parametri šablona klase

5. Postdekrement operator treba da sačuva **kopiju** svog operanda, izmjeni trenutni operand, a zatim vrati kopiju tog operanda

6. Ukoliko klasa naslijedi neku drugu klasu navodeći **protected** kao modifikator nasljeđivanja, svi **public** članovi bazne klase će biti „promovisani“ u **protected**, zaštićeni članovi će ostati zaštićeni, dok privatni članovi bazne klase će ostati privatni članovi (biće naslijeđeni, ali neće biti omogućen direktan pristup). Ako su neka polja/funkcije deklarirani kao **protected** u roditeljskoj klasi, jedino klase nasljednice će imati direktan pristup tim poljima/funkcijama

7. Dobija se to da se ne može vršiti implicitna konverzija.

8. Objektu se pristupa preko pokazivaca ili reference.

9. **Handler** tipa **A** će da prihvati **izuzetak** tipa **B** ako:

- Su A i B istog tipa

- A je javna osnovna klasa za izvedenu klasu B

- A i B su pokazivački tipovi i B može da se standardnom konverzijom konvertuje u tip A

10. Globalni identifikator sa unutrašnjim povezivanjem

Identifikatori iz anonimnog prostora imena : -imaju doseg na nivou datoteke

- moraju se razlikovati od globalnih identifikatora

- moraju se koristiti bez navodjenja rezolucionog operatora ::

Isti naziv identifikatora iz anonimnog prostora imena u dve datoteke označava dva različita entiteta

Poruka koristi anonimni prostor imena umjesto proglašavanja globalnih imena za static.