

PROGRAMSKI JEZICI 1 (4.7.2016.)

- ❶ (25 bodova) Definirati klasu **ModNum10** koja predstavlja broj u sistemu sa modulom 10 (u sistemu gdje su moguće vrijednosti broja u intervalu [0, 9], a sve operacije po modulu 10). Potrebno je implementirati:

- podrazumijevani konstruktor koji setuje vrijednost na 0
- eksplicitno kastovanje iz tipa `int` u ovaj tip koje baca izuzetak u obliku poruke „Impossible value“ u slučaju da je vrijednost veća ili jednaka 10. Izuzetak se obrađuje van klase, u kodu koji je pozvao kastovanje.
- **operator +** koji sabira dva objekta tipa **ModNum10** po modulu 10.
- **operator** za ispis na standardni izlaz.
- Izvršiti specijalizaciju šablona `std::less` za poređenje dva objekta tipa **ModNum10**, ako je zaglavlje **std::less** sljedeće:

```
struct less<Klasa>
{
    bool operator()(const Klasa&, const Klasa&) const
    {
        return poredjenje();
    }
};
```

- ❷ (25 bodova) Definirati generičku klasu **Value<T>** koja treba da enkapsulira objekat tipa **T** kako bi se nad njim mogle obavljati operacije. Klasa treba da ima podatak član tipa **T** i podrazumijevani konstruktor koji inicijalizuje taj podatak njegovim podrazumijevanim konstruktorom. Ova klasa namijenjena je za dalje izvođenje, pa implementirati metode, konstruktore i destruktore u skladu s tim. Pored toga, klasa treba da ima:

- konstruktor koji za parametar prihvata podatak tipa **T**
- **operator ()** koji vraća referencu na enkapsulirani objekat
- **operator <** koji poredi dva objekta tipa **Value<T>** korištenjem **operatora ()** iz klase **std::less<T>**
- metodu **print(ostream&)** koja štampa podatak na izlazni tok i **operator** za ispis na standardni izlaz koji omogućava štampanje potomaka na izlaz korištenjem metode **print(ostream&)**.

- ❸ (25 bodova) Definirati generičku klasu **CalculatedValue<T>** izvedenu iz klase **Value<T>**. Ova klasa treba da predstavlja vrijednost koja se dobija izračunavanjem neke binarne operacije nad dva operanda. Ova klasa treba da čuva lijevi i desni operand u obliku **Value<T>** i operaciju u obliku **std::function<T(T&, T&)>&**. Pored navedenog, klasa treba da ima:

- konstruktor koji prihvata operaciju i lijevi i desni operand, te inicijalizuje podatke članove (i reference članice)
- preklapljen **operator ()** iz natklase koji u ovoj klasi treba da vraća rezultat operacije nad operandima. Onemogućiti dalje preklapanje ovog operatora u klasama potomcima.

- ❹ (25 bodova) U glavnom programu treba napraviti dva objekta tipa **Value<ModNum10>** eksplicitnim kastovanjem iz 3 i 5.

Potom:

- Napraviti objekat tipa **CalculatedValue<ModNum10>** i proslijediti mu operaciju sabiranja i prethodno napravljena dva objekta. Operaciju konstruisati lambda operatorom koji poziva sabiranje iz klase **ModNum10**.
- U izlaznu datoteku pod imenom „output.txt“ upisati rezultat operacije konstruisanog objekta **CalculatedValue<ModNum10>** pozivom odgovarajućeg operatora.
- Pokušati kastovanje broja 10 u **Value<ModNum10>** i uhvatiti odgovarajući izuzetak, te poruku ispisati na standardni izlaz.

Napomene:

Razdvojiti interfejs i implementaciju.

Izbjeći dupliranje koda.

Izbjeći pisanje „magičnih brojeva“ definisanjem odgovarajućih konstanti.

U zaglavljima klasa koristiti punu specifikaciju imena objekata i metoda iz standardne biblioteke.

Koristiti standardnu biblioteku.

Sve metode treba da u zaglavlju sadrže odgovarajuću specifikaciju za signaliranje da li metoda baca izuzetak ili ne.

Deklaracije i definicije svih klasa treba da budu u prostoru imena **Ispit**.

Izuzetke bacati u obliku poruke tipa **char***.