Zadatak 2: Upoznavanje sa osnovama C++-a

# Zadatak 2.1

Cilj ovog zadatka je realizacija klase koja se koristi za rad sa stringovima promenljive duzine.

U okviru namespace-a **my\_*string*** kreirati klasu ***MyString*** koja ima sledeće metode:

* Default konstruktor, konstruktor kopije i konstruktor koji kao parametar prima *char\**
* Destruktor
* Operator + koji omogucava konkatenaciju stringova
* Operator >> koji omogucava ucitavanje stringa sa standardnog ulaza
* Operator << koji omogucava ispis stringa na standardni izlazMetodu length() koja vraca duzinu stringa
* Operator = (operator dodele)
* Operator == koji omogucava komparaciju stringova
* Metodu cat(????) koja konkatenira string na postojeci string
* Metodu substring(int pos, int length) koja vraca podstring od indeksa *pos*, duzine *length*
* Metodu isSubstring(????) koja proverava da li se zadati argument sadrzi u stringu
* Metodu append(int pos, ????) koja na indeksu *pos* umece zadati string iz drugog argumenta

Takodje, potrebno je kreirati klase koje ce omoguciti rukovanje izuzecima. Klase implementirati u okviru ***my\_string*** namespace-a. Analizirati prethodno navedene metode i dodati kod za izuzetke ukoliko je potrebno.

Napisati ***main*** funkciju koja demonstrira rad svih prethodno implementiranih funkcija. Takodje, dodati ***try-catch*** blok/ove i demonstrirati rukovanje izuzecima.

U toku izrade zadatka fokusirati se na:

* Implementaciju klase; klasa treba da sadrzi sve potrebne specijalne metode (konstruktore, destruktore). Voditi racuna o pravima pristupa za atribute i metode. Voditi racuna da sve sekcije (***public***, ***protected*** i ***private***) budu lepo odvojene.
* Obratiti paznju na tip argumenata i na tip povratne vrednosti funkcija
* Obratiti paznju na povratnu vrednost funkcija (kada treba da se vrati rezultat kao kopija, a kada kao referenca)
* Za alociranje i dealociranje memorije koristiti ***new***i ***delete*** operatore; istraziti kako ovi operatori rade (Note: ovo je mala razlika u odnosu na C programski jezik u kome ovi operatori ne postoje)
* Ukoliko je moguce, iskoristit i ***inline***funkcije (za neke manje, pomocne funkcije, samo ukoliko ima smisla)
* Ne duplirati kod. Ukoliko je potrebno, moguce je dodati i pomocne funkcije koje nisu gore definisane. Sakriti od korisnika sve pomocne funkcije

Napomena: C++ standardna biblioteka već ima podršku za rad sa stringovima u okviru klase std::string. Nije dozvoljeno korišćenje ove klase. Potrebno je implementirati spostvenu klasu koju realizuje identičnu funkcionalnost.