ZADAĆA 4

Objektno orijentirano programiranje

Upute:

Zadaću je potrebno predati do 27. ožujka u 08:00 na Teamsu. Diskutiranje zadaće u okviru "study group" je dozvoljen, ali predana zadaća mora biti samostalno riješena. Studenti koji predaju zadaću obavezni su prisustvovati vježbama, u suprotnome zadaća neće biti bodovana. Ako se kod ne prevodi (neovisno o tipu compiler-a), ili se događa greška prilikom izvršavanja koda, zadaća neće biti bodovana.

Zadatak 1. (70 bodova) Jednostruko povezana kružna lista

Jednostruka povezana kružna lista struktura je podataka koja se sastoji od niza čvorova, pri čemu svaki čvor sadrži informaciju o svom sljedbeniku, a sljedbenik "zadnjeg" čvora liste, je "prvi" element liste. Na osnovu sljedećih deklaracija klasa `Node` i `CSLL`, definirajte odgovarajuće članove.

Metoda `Node::swap` zamjenjuje vrijednost dvaju čvorova – čvora koji ju poziva i onog koji je proslijeđen po referenci. Ta metoda služi pri implementaciji neke od metoda za sortiranje u jednostruko povezanoj kružnoj listi. Metoda `CSLL::sort` sortira čvorove u rastućem poretku. Možete rabiti bilo koji od algoritama sortiranja (najjednostavnija je implementacija za Bubble Sort ili Insertion Sort). Nakon poziva metode `sort` nad nekom instancom klase `CSLL`, ta instanca treba sadržavati čvorove sortirane u rastućem poretku.

Dinamički alocirajte instancu klase `CSLL` te ju ispišite. Zatim redom popunite tu instancu s `double` vrijednostima: `59.9`, `13.7`, `10.0`, `98.44`, `16.7`, `20.269`, `1.5` te ju ponovno ispišite. Pomoću konstruktora kopiranja stvorite novu dinamički alociranu instancu, nad njom pozovite metodu sortiranja te ju ispišite.

```
struct Node {
	double value;
	Node* next;

	Node();
	Node(double value);
	Node(const Node& n);
	~Node();

	//swap premješta sadržaj između dva čvora
	void swap(Node& n);

	void print() const;
};
```

```
class CSLL {
protected:
	Node* head;
public:
	CSLL();
	CSLL(const CSLL& c);
	~CSLL();

	bool empty() const;

	void prepend(double value);
	void append(double value);

	double removeFromHead();

	void sort();

	void print() const;
};
```

Zadatak 2. (30 bodova) DynamicQueue

U ovom zadatku potrebno je po uzoru na klasu `Queue` implementiranu na trećim vježbama, implementirati klasu `DynamicQueue` – red koji enkapsulira `CSLL` iz zadatka 1 ove zadaće kao kontejner.

Deklaracija `DynamicQueue` klase je

```
class DynamicQueue {
protected:
    CSLL container;
public:
    DynamicQueue();
    DynamicQueue(const DynamicQueue& q);
    ~DynamicQueue();

    bool empty() const;

    void enqueue(double x);
    double dequeue();

    void print() const;
};
```

Dinamički alocirajte instancu klase `DynamicQueue` te ju ispišite. Zatim redom popunite tu instancu s `double` vrijednostima: `59.9`, `13.7`, `10.0`, `98.44`, `16.7`, `20.269`, `1.5` te ju ponovno ispišite. Pomoću konstruktora kopiranja stvorite novu dinamički alociranu instancu te ju ispišite.