

Ime i prezime: _____

Matični broj: _____

Ispitna pitanja

1. [3 bod.] Napišite predložak funkcije koja za predano joj polje proizvoljne veličine računa koliki postotak elemenata polja nije strogo unutar granica zadanih dodatnim argumentima funkcije.
2. [3 bod.] Proširite klasu Money danu izlistanjem kôda:

```
1  class Money {  
2      int mAmount;  
3  public:  
4      Money(int amount) : mAmount(amount) {}  
5  };
```

tako da se funkcija opisana u zadatku 1 može rabiti s poljima objekata klase Money. Testirajte funkciju poljem Money objekata.

3. [3 bod.] Ako je dana klasa Enemy prema izlistanju kôda:

```
1  class Enemy {  
2      std::string mName;  
3      int mHealth; // U rasponu [0-100]  
4      int mAttackDamage; // U rasponu [0-100]  
5  public:  
6      int getAttackDamage() const { return mAttackDamage; }  
7      int getHealth() const { return mHealth; }  
8      virtual void heal(int amount);  
9      virtual void takeDamage(int amount) = 0;  
10 };
```

napišite parametarski konstruktor koji stvara potpuno zdravog neprijatelja te implementaciju metode heal() tako da uvećava razinu zdravlja za predani iznos, ali liječenje nije moguće ako je zdravlje 0.

4. [2 bod.] Iz klase Enemy dane u zadatku 3 izvedite klasu koja predstavlja oklopljenog neprijatelja s podacima za vjerojatnost pogotka (između 0.0 i 1.0) i razinu oklopa (između 0 i 100). Napišite parametarski konstruktor za danu klasu.
5. [2 bod.] Za klasu iz zadatka 4 napišite preopterećenu inačicu metode takeDamage(). U slučaju da je nasumično generirana vrijednost $0.0 \leq r \leq 1.0$ manja od vjerojatnosti pogotka, nema štete. U suprotnom, najprije se umanjuje razina oklopa, a tek onda zdravlje.
6. [2 bod.] Napišite klasu koja predstavlja iznimku IllegalEnemyException izvedenu iz runtime_error klase. Preko objekta iznimke nužno je omogućiti pristup vrijednostima imena, zdravlja i snage napada neprijatelja koji je uzrokovao iznimku.
7. [4 bod.] Napišite funkciju createEnemies() koja na temelju predanog joj broja n vraća pokazivač na dinamički alocirano polje tipa Enemy popunjeno s n oklopljenih neprijatelja s nasumično postavljenim vrijednostima atributa (imena postaviti u obliku "Enemy1", "Enemy2", ... do n). U slučaju pokušaja stvaranja neprijatelja s napadom manjim od 50 podići iznimku, preko objekta iznimke ispisati informacije o neprijatelju te nastaviti stvarati neprijatelje. Bez obzira na iznimke, konačni broj neprijatelja mora biti jednak n. (za pretvaranje iz int u string moguće je rabiti string to_string (int val))

Koristiti programski jezik C++, osim gdje je naznačeno. U potpunosti rukovati resursima. Poštovati pravila enkapsulacije. Rabiti inicijalizacijske liste konstruktora. Izbjeći bespotrebno kopiranje objekata. Osigurati rad s konstantnim objektima gdje je moguće. Potpisati i predati sve ispitne listove. Ispit se piše 90 minuta.

8. [3 bod.] Napišite funkciju `runBattle()` koja kao argumente prima dva polja neprijatelja iz zadatka 3 jednake duljine. Svaki neprijatelj u prvom polju napada neprijatelja na nasumičnom indeksu u drugom polju nanoseći mu štetu u vrijednosti svog `mAttackDamage` atributa, a zatim svaki iz drugog na isti način napada nasumičnog neprijatelja u prvom. Funkcija vraća 1 ako je više preživjelih u prvom, 2 ako je više u drugom te 0 ako je broj preživjelih jednak. Testirati funkciju dvama poljima od 5 elemenata. Pretpostaviti postojanje funkcije `createEnemies()` iz zadatka 7.

9. [2 bod.] [C#] Kreirajte sučelje (engl. *interface*) naziva `ITaxable` koje će za klase koje ga implementiraju omogućiti dohvaćanje postotka poreza (vrijednost između 1 i 100), izračun ukupnog iznosa s porezom u kunama te izračun samo iznosa poreza u kunama.

10. [3 bod.] [C#] Za klasu danu izlistanjem kôda:

```
1  class Flour : ITaxable {  
2      public int TaxPercentage { get; private set; }  
3      public int QuantityKg { get; private set; }  
4      public int PricePerKg { get; private set; }  
5  
6      public Flour(int taxPercentage, int quantityKg, int pricePerKg) {  
7          TaxPercentage = taxPercentage; QuantityKg = quantityKg; PricePerKg = pricePerKg;  
8      }  
9  }
```

napišite implementaciju sučelja `ITaxable`.

11. [3 bod.] [C#] Napišite metodu koja za predanu joj listu tipa sučelja danog u zadatku 9 računa ukupnu cijenu s porezom za 10% najskupljih elemenata. Testirati metodu listom 100 nasumično inicijaliziranih objekata (podatkovne članove postaviti na nasumične vrijednosti u razumnim rasponima).