

Ime i prezime: _____

Matični broj: _____

Ispitna pitanja

1. [4 bod.] Napišite predložak funkcije koja za predano joj polje elemenata proizvoljne duljine provjerava jesu li svi njegovi elementi jedinstveni (ne sadrži duplikate).
2. [2 bod.] Ako je dana klasa YTVideo prema izlistanju kôda:

```
1  class YoutubeVideo {  
2      string mId;  
3      string mTitle;  
4  };
```

napišite sve potrebno kako bi funkcija iz zadatka 1 ispravno radila s klasom YTVideo.

3. [3 bod.] Ako je dana klasa SolarPanel prema izlistanju kôda, napišite sve potrebno kako bi se funkcija run() mogla ispravno izvesti.

```
1  class SolarPanel {  
2  protected:  
3      double mWidth, mHeight; // [m], nuzno > 0  
4      double mEfficiency; // u rasponu 0.0–1.0  
5  };  
6  void run() {  
7      SolarPanel smallPanel, largePanel(2, 2, 0.6);  
8      cout << smallPanel << ", " << largePanel << endl;  
9  }
```

4. [2 bod.] Za klasu SolarPanel iz zadatka 3 napišite virtualnu metodu koja računa godišnju proizvodnju energije u kWh prema izrazu $E = A \cdot r \cdot 1300$, gdje je A površina panela u m^2 , a r predstavlja mEfficiency.
5. [3 bod.] Za klasu SolarPanel iz zadatka 3 napišite funkciju koja za predani panel računa kolika je godišnja zarada, ako je cijena kWh zadana argumentom funkcije. Napišite preopterećenu inačicu ove funkcije koja računa zaradu za predano joj polje panela.
6. [4 bod.] Napišite funkciju koja će između dva predana joj solarna panela izabrati onaj koji će proizvesti više energije. Funkcija treba količinu energije kao i informacije o odabranom solarnom panelu zapisati u datoteku čije se ime zadaje kao argument funkcije. Testirati funkciju, a kao ime datoteke odabrati vlastito ime.
7. [2 bod.] Iz klase SolarPanel izvedite klasu koja predstavlja solarni panel s podacima o faktoru starenja u rasponu 0.0-0.1 i godinama starosti. Napišite parametarski konstruktor te napišite metodu za inkrementiranje godina starosti.
8. [3 bod.] Za klasu iz zadatka 7 prepisite metodu za izračun proizvedene energije tako da se za svaku godinu starosti koeficijent efikasnosti umanjuje za iznos faktora starenja. Ako je u konačnici efikasnost niža od 0.02, podići kao iznimku objekt vlastite klase iznimke izvedene iz klase runtime.error s informacijom o konačnom faktoru efikasnosti. Omogućiti pristup toj vrijednosti preko objekta iznimke.

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 120 minuta.

9. [4 bod.] Napišite kôd u kojem se polje od n elemenata (gdje se n zadaje s tipkovnice) tipa osnovne klase do polovine popunjava nasumično stvorenim objektima osnovne klase, a od polovine nasumično stvorenim objektima izvedene klase. Zatim se računa i na ekran ispisuje koliko je energije proizveo onaj panel koji proizvodi najviše. U slučaju iznimke ne uzeti panel koji je do nje doveo u obzir kod izračuna, već samo ispisati vrijednost faktora efikasnosti preko objekta iznimke.

10. [2 bod.] [C#] Ako je dana klasa Student prema izlistanju:

```
1  class Student {
2      public double ExamPercent { get; private set; } // 0.0–1.0
3      public double AttendancePercent { get; private set; } // 0.0–1.0
4      public Student(double examPercent, double attendancePercent) {
5          if (examPercent > 1.0 || attendancePercent > 1.0) throw new ArgumentOutOfRangeException();
6          if (examPercent < 0.0 || attendancePercent < 0.0) throw new ArgumentOutOfRangeException();
7          ExamPercent = examPercent; AttendancePercent = attendancePercent;
8      }
9  }
```

Napišite sučelje *IGrader* koje će za klase koje ga implementiraju omogućiti:

- Za predanu listu studenata kreiranje nove liste u kojoj svaki element označava je li student u originalnoj listi prošao ili pao
- Za predanu listu studenata izračun prosječnog rezultata na ispitu.

11. [4 bod.] [C#] Za klasu *FeritGrader* danu izlistanjem

```
1  class FeritGrader {
2      public double MinScore { get; private set; }
3      public double MinAttendance { get; private set; }
4      public FeritGrader(double minScore, double minAttendance) {
5          // Kontrola granicnih vrijednosti je obavljena, ali izostavljena radi prostora
6          MinScore = minScore; MinAttendance = minAttendance;
7      }
8  }
```

napišite implementaciju sučelja iz zadatka 10. Za prolaz je nužno imati više od minimuma po oba kriterija. Prosječan rezultat studenata na ispitu računa se samo za one studente koji su položili.

12. [2 bod.] [C#] Napišite funkciju koja za predanu joj listu objekata klase *Student* pronalazi i iz liste uklanja one studente koji niti jednom nisu bili na nastavi. Moguće rabiti metodu liste čiji je prototip `void RemoveAt(int index)`. Ako je dana popunjena lista studenata prema izlistanju

```
1  List<Student> students = RetrieveStudentsFromISVU(); // Pretpostaviti da je dohvacanje studenata implementirano
```

izračunajte i na ekran ispišite je li studenata kojih nije nikada bilo više od pola.