

Ime i prezime: _____

Matični broj: _____

Ispitna pitanja

1. [3 b.] Ako je, prema izlistanju kôda, dana klasa koja predstavlja knjigu, definirajte sve potrebno kako bi se metoda Run() mogla u potpunosti ispravno izvesti uz prikazani izlaz. Trebate podržati isključivo žanrove Fantasy, Thriller, Biography.

```

1 class Book {
2     Genre genre;
3     string title;
4     string author;
5 }
6 public static void Run() { // Member of the Program class
7     Book enigma = new Book("Alan Turing: The enigma", "Andrew Hodges", Genre.Biography);
8     Book another = new Book(enigma);
9     another.Genre = Genre.Thriller;
10    Console.WriteLine($"Book info: {another.Title}, {another.Author}, {another.Genre}");
11 }

```

```
Book info: Alan Turing: The enigma, Andrew Hodges, Thriller
```

2. [3 b.] Ako je dana klasa koja predstavlja audioknjigu izvedena iz klase Book iz zadatka 1, definirajte odgovarajući parametarski konstruktor i metodu koja za predan datum željenog završetka slušanja knjige računa koliko minuta po danu je potrebno slušati knjigu kako bi se do tada dovršila. Od koristi može biti svojstvo double TotalDays { get; } strukture TimeSpan.

```

1 class AudioBook : Book {
2     public DateTime StartedAt { get; private set; }
3     public int DurationMinutes { get; private set; }
4     public int MinutesListened { get; private set; }
5 }

```

3. [3 b.] Za klasu BookReporter napišite metodu koja za predanu listu audioknjiga vraća objekt iz kojega je moguće dobiti informaciju o broju ukupno odslušanih i neodslušanih minuta zadanoj žanru. Primjer korištenja ove metode za popunjenu listu imena audiobooks dan je izlistanjem kôda.

```

1 ListeningReport report = BookReporter.Analyze(audiobooks, Genre.Fantasy);
2 Console.WriteLine($"Report for {report.Genre}: {report.Listened} min. done, {report.Remaining} min. left.");

```

4. [3 b.] U danom izlistanju kôda pronadîte pogreške i dajte prijedlog ispravka

```

1 public class ReadingTracker {
2     public string id;
3     public int booksLiked;
4     public int booksRead;
5     public ReadingTracker(string id) {
6         id = id; booksLiked = 0; booksRead = 0;
7     }
8     public int SetBooksRead() { booksRead = value; }
9     public static string operator > (ReadingTracker left, ReadingTracker right) {
10         if (left.booksRead > right.booksRead)
11             return left.id;
12         return right.id;
13     }
14     public void GetLikedToReadRatio() { return (double) booksLiked / booksRead; }
15 }

```

▼▼▼

5. [3 b.] Ako je, prema izlistanju kôda, dana klasa koja predstavlja izvješće, definirajte metodu koja na temelju predanog polja stringova koji predstavljaju ključne riječi stvara novo izvješće. Stvoreno izvješće ima isti naslov i sadržaj kao i originalno, ali su u njegovom sadržaju svaka pojedina ključna riječ zamijenjene je s onoliko znakova '*' koliko slova sadrži. Od koristi može biti metoda klase string prototipa string Replace(string oldValue, string newValue), kao i konstruktor klase string String(char c, int count).

```

1 abstract class Report {
2     public string Title { get; protected set; }
3     public string Contents { get; protected set; }
4     public Report(string title, string contents) {
5         Title = title; Contents = contents;
6     }
7 }
8 public static void Run() {
9     Report report = ReportGenerator.Create("Messages", "Josipa wrote to Ivan."); // Assume report generator exists.
10    Report masked = report.Mask(new string[] { "wrote" });
11 }

```

6. [3 b.] Od svih budućih inaćica izvješća očekuje se ponašanje koje omogućuje pronađazak i izdvajanje novčanih iznosa koji se spominju u sadržaju izvješća. S obzirom da nije unaprijed poznato kakve će sve konkretnе inaćice izvješća postojati, nije poznato niti u kakvom će obliku biti rezultati izdvajanja novčanih iznosa. Jedino što je nužno je da svaki konkretni tip takvog rezultata ima mogućnost saznavanja koliki je ukupni iznos i omogući pristup pojedinom iznosu preko indeksa. Osigurajte ovakvo ponašanje za sve buduće konkretnе inaćice izvješća i rezultata izdvajanja imena proširivanjem klase `Report` i dodavanjem eventualnih klasa za postizanje ovog cilja.
7. [3 b.] Definirajte klasu koja predstavlja arhiv i omogućuje pohranu više različitih izvješća (objekti tipa `Report`) pri čemu je potrebno osigurati da je naslov svakog pohranjenog izvješća jedinstven. Omogućite pohranu izvješća i dohvaćanje izvješća preko naslova. Definirajte metodu koja pronađazak i kao rezultat u obliku stringa vraća sva imena koja se spominju u izvješćima. Ime je svaka riječ u sadržaju izvješća koja počinje velikim slovom. U slučaju da ne postoji niti jedno takvo ime, podići iznimku vlastitog tipa `MissingReportException`. Testirajte napisanu metodu, ako je dana popunjena arhiva imena porukice, uz ispravno rukovanje iznimkama.

▼▼▼

8. [3 b.] Prema izlistanju kôda dani su klasa `Task` i sučelje `ITaskProcessor`. Napišite implementaciju tog sučelja imena `FeritProcessor`. Važni zadaci koji se prebrojavaju su oni koji imaju ekstremne periode. Ekstremni period je onaj koji se za barem 20% razlikuje od prosjeka. Kreira se lista čiji elementi predstavljaju vremena kada će završiti zadaci s ekstremnim periodima, ake se počinju izvršavati u trenutku poziva metode.

```

1 class Task {
2     public string Name { get; set; }
3     public int Duration { get; set; } // in seconds
4     public int Period { get; set; } // task repeats every period seconds
5     public Task(string name, int duration, int period) {
6         Name = name; Duration = duration; Period = period;
7     }
8 }
9 interface ITaskProcessor {
10     int CountImportant(List<Task> tasks);
11     List<DateTime> GetNotificationTimestamps(List<Task> tasks);
12 }

```

9. [3 b.] Definirajte klasu čije stanje čine dvije kolekcije zadataka, jedna koja predstavlja osobne i druga koja predstavlja poslovne te alat za obradu zadataka. Mora biti moguće rabiti bilo kakav alat za obradu zadataka te ga naknadno promijeniti. Definirajte metodu koja oslanjači se na vlastite atribute i polimorfizam računa koliki je omjer važnih osobnih i važnih poslovnih zadataka.
10. [3 b.] Za statičku klasu `Profiler` definirajte metodu koja na temelju predane joj liste zadataka (`Task` objekata) i delegata odgovarajućeg tipa omogućuje izdvajanje onih zadataka koji zadovoljavaju uvjet predstavljen delegatom. Testirajte metodu uporabom lambda izraza tako da izdvojite one zadatake kojima je period paran broj.