

Programmazione ad Oggetti

Mod. 2

30/6/2023

Studente _____ Matricola _____

1. Si prenda in considerazione la seguente classe Java 8+, parametrica su due generics X e Y , le cui istanze rappresentano sequenze *iterabili* di coppie di coordinate cartesiane $X \times Y$ che rappresentano i punti individuati da una funzione $X \rightarrow Y$ in un determinato intervallo del dominio X .

```
public class FunSeq<X extends Number & Comparable<? super X>, Y extends Number>
    implements Iterable<Pair<X, Y>> {

    private final X a, b;
    private final Function<? super X, ? extends Y> f;
    private final Function<X, X> inc;

    public FunSeq(X a, X b, Function<? super X, ? extends Y> f, Function<X, X> inc) {
        this.a = a;
        this.b = b;
        this.f = f;
        this.inc = inc;
    }

    /* da finire di implementare */
}
```

Al costruttore vengono passati l'intervallo di dominio $[a, b]$, la funzione f e la funzione inc per incrementare l'ascissa. La funzione f è la funzione principale: per ogni ascissa x di tipo X compresa nell'intervallo $[a, b]$, l'applicazione $f(x)$ ha tipo Y . La sequenza iterabile consiste in coppie $(x, f(x))$ di tipo $\text{Pair}<X, Y>$ che rappresentano l'ascissa e l'ordinata di ogni punto. Essendo impossibile rappresentare funzioni continue, le ascisse procedono in maniera discreta secondo un valore di incremento: la funzione inc è necessaria proprio per calcolare la prossima ascissa nella sequenza.

- (a) 1 punti Si definisca una classe `Pair` parametrica su 2 generics A e B che rappresenta coppie *immutabili*.
- (b) 8 punti Si finisca di implementare la classe `FunSeq` opportunamente. Per realizzare il confronto delle ascisse, si noti che il type parameter X è vincolato non solo ad essere un `Number` ma anche a implementare `Comparable<X>`.
- (c) 5 punti Si implementi uno snippet di codice che utilizzi la classe `FunSeq` per rappresentare la parabola $x^2 + 2x - 1$ nel piano $\mathbb{R} \times \mathbb{R}$ nell'intervallo di dominio $[-2, 2)$ con incremento di ascissa 0.1. Ciò significa che le ascisse partono da -2 e procedono a step di 0.1 fino a 2 (escluso), producendo $((2 - 0.1) - (-2))/0.1 = 39$ punti. Si definisca la parabola con una opportuna lambda.
- (d) 2 punti (bonus) Si implementi uno snippet simile a quello del punto precedente ma che rappresenti la parabola data nel piano $\mathbb{R} \times \mathbb{Z}$, ovvero con ordinate di tipo `int`.
2. 16 punti Si implementi in C++ una classe templatizzata `fun_seq` analoga a quella dell'esercizio precedente, adottando però i pattern, gli stili e le convenzioni di C++ e di STL. Alcuni suggerimenti:

- si usi il tipo `std::pair` definito da STL per rappresentare le coppie;
- l'iterabilità in C++ è rappresentata dal *concept* denominato **Container** in STL;
- la funzione di incremento passata al costruttore in Java non è necessaria in C++ e può essere sostituita dagli opportuni operatori;
- il vincolo **Comparable** sul type parameter **X** in Java non ha equivalente in C++, è sufficiente assumere che il relativo template argument supporti un operatore di confronto.

Si scriva anche uno snippet analogo a quello del punto 1.(c).

Question:	1	2	Total
Points:	14	16	30
Bonus Points:	2	0	2
Score:			