

## Задание №5 Классы, интерфейсы `Iterator<T>/Iterable<T>`, `Comparable<T>`

- 1) Создать класс (классы), указанный в задании. По возможности использовать `assert` и исключения для обработки ошибочных ситуаций.
- 2) В отдельном файле разработать тестовое приложение, использующее класс (классы), указанный в задании. Провести тестирование всех методов и конструкторов с выводом данных и результатов

1) Определить класс `Point` для описания координат точки в двумерном пространстве. Создать абстрактный базовый класс `Figure` с виртуальными или абстрактными методами вычисления площади и периметра. Создать производный класс `Rectangle` (прямоугольник).

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения
- интерфейс `Iterator` - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить `toString()`)
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

2) Определить класс `Point` для описания координат точки в двумерном пространстве. Создать абстрактный базовый класс `Figure` с виртуальными или абстрактными методами вычисления площади и периметра. Создать производный класс `Trapezium` (трапеция).

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения
- интерфейс `Iterator` - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить `toString()`)
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

3) Создать интерфейс `Body` с методами вычисления площади поверхности и объёма. Создать класс `Parallelepiped` (параллелепипед) реализующий интерфейс `Body`.

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения

- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

4) Создать интерфейс Body с методами вычисления площади поверхности и объёма. Создать класс Ball (шар) реализующий интерфейс Body.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

5) Создать интерфейс Function с методом вычисления значения функции  $y=f(x)$ . Создать классы Ellipse (эллипс) и Hyperbola (гипербола) реализующие интерфейс Function (по каноническому уравнению).

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

6) Создать абстрактный базовый класс Triangle для представления треугольника с виртуальными или абстрактными функциями вычисления площади и периметра. Поля данных должны включать две стороны и угол между ними. Определить класс наследник - равнобедренный треугольник, который должен реализовать функции вычисления площади и периметра.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

7) Создать абстрактный базовый класс Triangle для представления треугольника с абстрактными функциями вычисления площади и периметра. Поля данных должны включать две стороны и угол между ними. Определить класс наследник - равнобедренный треугольник, который должен реализовать функции вычисления площади и периметра.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

8) Рациональная (несократимая) дробь представляется парой целых чисел (a,b), где a - числитель, b - знаменатель. Создать класс Rational для работы с рациональными дробями. Необходимо реализовать 4 арифметические операции: +, -, \*, /.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

9) Нечёткие числа представляются тройками чисел  $(x - e1, x, x + e2)$ . Реализовать класс `FuzzyNumber` для работы с нечёткими числами. Необходимо реализовать 4 арифметические операции:  $+-*/$  и метод вычисления обратного нечёткого числа.

Правила для чисел  $A = (A - al, A, A + ar)$  и  $B = (B - bl, B, B + br)$ :

$$A + B = (A + B - al - bl, A + B, A + B + ar + br)$$

$$A - B = (A - B - al - br, A - B, A - B + ar + bl)$$

$$A * B = (A * B - B * al - A * bl + al * bl, A * B, A * B + B * ar + A * br + ar * br)$$

$$A / B = ((A - al)/(B + br), A / B, (A + ar)/(B - bl)), B > 0$$

$$\text{обратное } A = (1 / (A + ar), 1 / A, 1/(A - al)), A > 0$$

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения
- интерфейс `Iterator` - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить `toString()`)
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

10) Разработать класс `Angle` для работы с углами на плоскости, задаваемыми величиной в градусах и минутах.

Необходимо реализовать 4 арифметические операции:  $+-*/$ , методы: перевод в радианы, приведение к диапазону 0-360.

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения
- интерфейс `Iterator` - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить `toString()`)
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

11) Разработать абстрактный базовый класс `Series` (Прогрессия) и производный класс `Linear` - арифметическая прогрессия, который должен реализовать вычисления  $J$ -го члена прогрессии, метод вычисления суммы прогрессии.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

12) Разработать абстрактный базовый класс Series (Прогрессия) и производный класс Exponential - геометрическая прогрессия, который должен реализовать вычисления J-го члена прогрессии, метод вычисления суммы прогрессии.

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

13) Разработать класс Account, представляющий собой банковский счёт. В классе должны быть следующие поля:

- ФИО владельца
- номер счёта
- наименование валюты счёта
- годовой процент
- сумма

Необходимо реализовать операции со счётом: снять указанную сумму, добавить указанную сумму, начислить проценты за указанный период времени

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

14) Разработать класс `Bankomat`, моделирующий работу банкомата. Класс должен содержать поля для хранения ID-номера банкомата, информации о текущей сумме остатка, минимальной и максимальной сумме, которые позволяет снять клиенту в один день. Сумма денег представляется полями номиналами: 10тр - 100тр.

Реализовать методы:

- загрузка купюр
- снятие суммы (проверять корректность суммы и возможность выдачи купюрами, которые есть в наличии)

Класс должен реализовать:

- интерфейс `Comparable` с возможностью выбора одного из полей для сравнения
- интерфейс `Iterator` - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить `toString()`)
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.

15) Разработать класс `Paiment` (зарплата). В классе должны быть следующие поля:

- ФИО
- год поступления на работу
- оклад
- кол-во рабочих дней в месяце
- кол-во отработанных дней в месяце
- процент надбавки
- процент налогов и отчислений
- начисленная сумма
- удержанная сумма

Реализовать методы:

- вычисления начисленной суммы (2)
- вычисления удержанной суммы (1)
- вычисления суммы выдаваемой на руки (0)

Класс должен реализовать:

- интерфейс Comparable с возможностью выбора одного из полей для сравнения
- интерфейс Iterator - индексатор по всем полям объекта
- метод для сохранения значений всех полей в строке текста (переопределить toString())
- конструктор или метод для инициализации объекта из строки текста

Создать консольное приложение, демонстрирующее использование класса. Создать небольшой массив объектов и напечатать отсортированными по выбранному полю.