Задание №5 Классы, интерфейсы Iterator<T>/Iterable<T>, Comparable<T>/Compartor<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). Кроме геометрических характеристик класс Rectangle хранит ещё цвет контура и цвет заливки.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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). Кроме геометрических характеристик класс Trapezium хранит ещё цвет контура и цвет заливки.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 Parallepiped (параллепипед) реализующий интерфейс Body. Кроме геометрических характеристик класс Parallepiped хранит ещё цвет поверхности.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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. Кроме геометрических характеристик класс Ball хранит ещё цвет поверхности и номер шара.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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). Кроме геометрических характеристик классы Ellipse и Hyperbola хранят ещё цвет контура и координаны центра.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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. Кроме геометрических характеристик класс наследник хранит ещё цвет контура и цвет заливки.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 - равносторонний треугольник, который должен реализовать функции вычисления площади и периметра. Кроме геометрических характеристик класс наследник хранит ещё цвет контура и цвет заливки.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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:

-интерфейсы Comparable и Comparator 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 в виде статических методов: 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

обратное A = (1 / (A + ar), 1 / A, 1/(A - al)), A > 0

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 (в виде статических методов): +-*/, методы: перевод в радианы, приведение к диапазону 0-360.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 класс Profile (Профиль местности) – последовательность высот вычисленных через равные промежутки по-горизонтали. Необходимо реализовать методы вычисления наибольшей, наименьшей и средней высот.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 Linear - арифметическая прогрессия, который должен реализовать вычисления J-го члена прогрессии, метод вычисления суммы прогрессии.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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 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:

-интерфейсы Comparable и Comparator 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 Man (человек), с полями: имя, возраст, пол и вес). Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс Student, имеющий поля факультет, курс, группа. Определить методы изменения возраста, веса, перехода на следующий курс, перевода в другую группу.

Класс Student должен реализовать:

-интерфейсы Comparable и Comparator 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 Pair (пара целых чисел); определить метод умножения на число и операцию сложения пар (a, b) + (c, d) = (a + b, c + d). Определить класс-наследник Money с полями: рубли и копейки. Переопределить операцию сложения и определить методы вычитания и деления денежных сумм.

Класс Money должен реализовать:

-интерфейсы Comparable и Comparator 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.

16) Разработать базовый класс Car (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс Lorry (грузовик), характеризуемый также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.

Класс Lorry должен реализовать:

-интерфейсы Comparable и Comparator 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.

17) Разработать класс Polynom для работы с многочленами. Коэффициенты должны быть представлены массивом элементов-коэффициентов. Младшая степень имеет меньший индекс (нулевая степень — нулевой индекс). Размер массива задается как аргумент конструктора. Реализовать арифметические операции и операции сравнения, вычисление значения полинома для заданного значения x.

Класс должен реализовать:

-интерфейсы Comparable и Comparator 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.

18) Разработать суперкласс Computer и на его основе подклассы: Desktop и Noutebook. В базовом классе определите общие свойства, а в производных - уникальные свойства.

Производные классы должен реализовать:

-интерфейсы Comparable и Comparator 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.

19) 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:

-интерфейсы Comparable и Comparator 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.

20) 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 позволяется 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:

-интерфейсы Comparable и Comparator 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.

21) 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:

-интерфейсы Comparable и Comparator 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.