Во всех заданиях реализовать вывод на экран, методы получения значений полей и методы установки значений полей, а также необходимые конструкторы (если это не указано в задании явно). Конструкторы и методы обязательно должны проверять параметры на допустимость; в случае неправильных данных — выводить сообщение об ошибке и заканчивать работу. Преобразование в строку реализовать в виде функции преобразования string.

Для демонстрации работы с объектами нового типа во всех заданиях требуется написать главную функцию. В программе должны присутствовать различные способы создания объектов и массивов объектов. Программа должна демонстрировать использование всех функций и методов.

**Открытое наследование**

Во всех заданиях реализовать функцию, получающую и возвращающую объекты базового класса. Продемонстрировать принцип подстановки.

1. Создать базовый класс Саг (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.
2. Создать класс Pair (пара чисел); определить методы изменения полей и срав­нения пар: пара pi больше пары р2, если (first.pl > first.р2) или (first.pl = first.р2) и (second.pl > second.р2). Определить класс-наследник Fraction с по­лями: целая часть числа и дробная часть числа. Определить полный набор ме­тодов сравнения.
3. Создать класс Liquid (жидкость), имеющий поля названия и плотности. Определить методы переназначения и изменения плотности. Создать произ­водный класс А1 cohol (спирт), имеющий крепость. Определить методы пере­назначения и изменения крепости.
4. Создать класс Pair (пара чисел); определить методы изменения полей и вы­числения произведения чисел. Определить производный класс Rectangle (прямоугольник) с полями-сторонами. Определить методы вычисления пе­риметра и площади прямоугольника.
5. Создать класс Man (человек), с полями: имя, возраст, пол и вес. Определить методы переназначения имени, изменения возраста и изменения веса. Соз­дать производный класс Student, имеющий поле года обучения. Определить методы переназначения и увеличения года обучения.
6. Создать класс Triad (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle с поля­ми-сторонами. Определить методы вычисления углов и площади треугольника.
7. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс Equilateral (равносторонний), имеющий поле площади. Определить метод вычисления площади.
8. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс RightAngled (прямоугольный), имеющий поле площади. Определить метод вычисления площади.
9. Создать класс Pair (пара чисел); определить методы изменения полей и вы­числения произведения чисел. Определить производный класс RightAngled с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.
10. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Date с полями: год, месяц и день. Определить полный набор методов сравнения дат.
11. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Time с полями: час, минута и секунда. Определить полный набор методов сравнения моментов времени.
12. Реализовать класс-оболочку Number для числового типа float. Реализовать ме­тоды сложения и деления. Создать производный класс Real, в котором реали­зовать метод возведения в произвольную степень, и метод для вычисления логарифма числа.
13. Создать класс Triad (тройка чисел); определить методы увеличения полей на 1. Определить производный класс Date с полями: год, месяц и день. Переопреде­лить методы увеличения полей на 1 и определить метод увеличения даты на п дней.
14. Реализовать класс-оболочку Number для числового типа double. Реализовать ме­тоды умножения и вычитания. Создать производный класс Real, в котором реализовать метод, вычисляющий корень произвольной степени, и метод для вычисления числа п в данной степени.
15. Создать класс Triad (тройка чисел); определить методы увеличения полей на
16. Определить класс-наследник Time с полями: час, минута, секунда. Пере­определить методы увеличения полей на 1 и определить методы увеличения на п секунд и минут.
17. Создать базовый класс Pair (пара целых чисел) с операциями проверки на ра­венство и перемножения полей. Реализовать операцию вычитания пар по формуле (a,b) - (с, d) = (а - b, с - d). Создать производный класс Rational; определить новые операции сложения (а, b) + (с, d) = (ad + bc, bd) и деления (а, b) / (с, d) = (ad, bc); переопределить операцию вычитания (а, b) - (с, d) = = (ad - bc, bd).
18. Создать класс Pair (пара чисел); определить метод перемножения полей и операцию сложения пар (а, b) + (с, d) = (а + b, с + d). Определить производ­ный класс Complex с полями: действительная и мнимая части числа. Опреде­лить методы умножения (a, b) x(c,d) = (ас - bd, ad + be) и вычитания (а, b) - - (с, d) = (а - b, с - d).
19. Создать класс Pair (пара целых чисел); определить методы изменения полей и операцию сложения пар (а, b) + (с, d) = (а + b, с + d). Определить класс-на­следник Long с полями: старшая часть числа и младшая часть числа. Пере­определить операцию сложения и определить методы умножения и вычитания.
20. Создать базовый класс Triad (тройка чисел) с операциями сложения с чис­лом, умножения на число, проверки на равенство. Создать производный класс vector3D, задаваемый тройкой координат. Должны быть реализованы: опера­ция сложения векторов, скалярное произведение векторов.
21. Создать класс Pair (пара целых чисел); определить метод умножения на чис­ло и операцию сложения пар (а, b) + (с, d) = (а + Ь, с + d). Определить класс- наследник Money с полями: рубли и копейки. Переопределить операцию сло­жения и определить методы вычитания и деления денежных сумм.

**Массивы и наследование**

Создать базовый класс Array, в котором определите поле-массив подходящего типа и поле для хранения количества элементов у текущего объекта-массива. Максимально возможный размер массива задается статической константой. Реа­лизуйте конструктор инициализации, задающий количество элементов и началь­ное значение (по умолчанию 0). Реализуйте методы доступа к отдельному эле­менту, перегрузив операцию индексирования []. При этом должна выполняться проверка индекса на допустимость

Реализовать один из классов заданий главы 2 как производный класс от класса Array; использовать открытое наследование. Во всех заданиях необходимо реали­зовать конструкторы инициализации и конструктор без аргументов. Указанные в задании операции реализуются посредством перегрузки подходящих операций. Во всех заданиях должны быть поддержаны соответствующие операции с при­сваиванием, ввод с клавиатуры, вывод на экран.

1. Создать класс Polinom (см. задание 2.5).
2. Создать класс BitString (см. задание 2.1).
3. Создать класс Decimal (см. задание 2.2).
4. Создать класс Money для работы с денежными суммами (см. задание 2.4).
5. Создать класс Decimal (см. задание 2.2) для работы со знаковыми целыми. Знак представить отдельным полем sign.
6. Создать класс Hex (см. задание 2.3).
7. Создать класс Octal (см. задание 2.19).
8. Создать класс PriceList (см. задание 2.21).
9. Создать класс Fraction для работы с дробными десятичными числами. Коли­чество цифр в дробной части должно задаваться в отдельном поле и инициа­лизироваться конструктором. Знак представить отдельным полем sign.
10. Создать класс Dictionary (см. задание 2.10).
11. Создать класс TestContent (см. задание 2.11).
12. Создать класс ListPerson (см. задание 2.12).
13. Создать класс Goods (см. задание 2.13).
14. Создать класс Subscriber (см. задание 2.14).
15. Создать класс Directory (см. задание 2.15).
16. Создать класс ListPaier (см. задание 2.16).
17. Создать класс WorkTeacher (см. задание 2.18).
18. Создать класс String (см. задание 2.20).
19. Создать класс PlanEducation (см. задание 2.17).
20. Создатькласс DailySchedule (см. задание 2.21).