Виртуальные функции

Создать базовый класс Array с полями: массив типа unsigned char и поле для хра­нения количества элементов у текущего объекта-массива. Максимально возмож­ный размер массива задается статической константой. Реализовать конструктор инициализации, задающий количество элементов и начальное значение (по умолчанию 0). Реализовать метод доступа к элементу, перегрузив операцию ин­дексирования []. При этом должна выполняться проверка индекса на допусти­мость (задание 1.8).

Реализовать в классе Array виртуальную функцию поэлементного сложения мас­сивов. Реализовать два класса, переопределив виртуальную функцию сложения. Вызывающая программа должна продемонстрировать все варианты вызова вир­туальных функций.

1. Создать класс Fraction (задание 29) и класс BitString (задание 2.1).
2. Создать класс BitString (задание 2.1) и класс Hex (задание 2.3).
3. Создать класс Decimal (задание 2.2) и класс BitString (задание 2.1).
4. Создать класс Money (задание 2.4) и класс BitString (задание 2.1).
5. Создать класс Decimal (задание 25) и класс BitString (задание 2.1).
6. Создать класс Hex (задание 2.3) и класс BitString (задание 2.1).
7. Создать класс Octal (задание 2.19) и класс BitString (задание 2.1).
8. Создать класс Hex (задание 2.3) и класс Octal (задание 2.19).
9. Создать класс Hex (задание 2.3) и класЬ Money (задание 2.4).
10. Создать класс Hex (задание 2.3) и класс Fraction (задание 29).
11. Создать класс Decimal (задание 25) и класс Hex (задание 2.3).
12. Создать класс Octal (задание 2.19) и класс Hex (задание 2.3).
13. Создать класс Decimal (задание 25) и класс Octal (задание 2.19).
14. Создать класс Decimal (задание 25) и класс String (задание 2.20).
15. Создать класс BitString (задание 2.1) и класс String (задание 2.20).
16. Создать класс Fraction (задание 29) и класс String (задание 2.20).
17. Создать класс Hex (задание 2.3) и класс String (задание 2.20).
18. Создать класс Octal (задание 2.19) и класс String (задание 2.20).
19. Создать класс Money (задание 2.4) и класс String (задание 2.20).
20. Создать класс Money (задание 2.4) и класс Hex (задание 2.3).

Абстрактные классы

В следующих заданиях требуется реализовать абстрактный базовый класс, опре­делив в нем чистые виртуальные функции. Эти функции определяются в произ­водных классах. В базовых классах должны быть объявлены чистые виртуаль­ные функции ввода/вывода, которые реализуются в производных классах.

Вызывающая программа должна продемонстрировать все варианты вызова вир­туальных функций с помощью указателей на базовый класс. Написать функцию вывода, получающую параметры базового класса по ссылке и демонстрирующую виртуальный вызов

.

1. Создать абстрактный базовый класс Figure с виртуальными методами вычис­ления площади и периметра. Создать производные классы: Rectangl е (прямо­угольник), Circle (круг), Trapezium (трапеция) со своими функциями площади и периметра. Самостоятельно определить, какие поля необходимы, какие из них можно задать в базовом классе, а какие — в производных. Площадь тра­пеции: S = (а + Ь) х h /
2. Создать абстрактный базовый класс Number с виртуальными методами — ариф­метическими операциями. Создать производные классы Integer (целое) и Real (действительное).
3. Создать абстрактный базовый класс Body (тело) с виртуальными функциями вычисления площади поверхности и объема. Создать производные классы: Parallelepiped (параллелепипед) и Ball (шар) со своими функциями площади поверхности и объема.
4. Создать абстрактный класс Currency (валюта) для работы с денежными сум­мами. Определить виртуальные функции перевода в рубли и вывода на эк­ран. Реализовать производные классы Do! 1 аг (доллар) и Euro (евро) со своими функциями перевода и вывода на экран.
5. Создать абстрактный базовый класс Triangle для представления треугольника с виртуальными функциями вычисления площади и периметра. Поля данных должны включать две стороны и угол между ними. Определить классы-на­следники: прямоугольный треугольник, равнобедренный треугольник, равно­сторонний треугольник со своими функциями вычисления площади и пери­метра.
6. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Создать производные классы FazzyNumber (задание 1.31) и Complex (задание 1.21).
7. Создать абстрактный базовый класс Root (корень) с виртуальными методами вычисления корней и вывода результата на экран. Определить производные классы Linear (линейное уравнение) и Square (квадратное уравнение) с собст­венными методами вычисления корней и вывода на экран. '
8. Создать абстрактный базовый класс Function (функция) с виртуальными ме­тодами вычисления значения функции у = f(r) в заданной точке х и вывода результата на экран. Определить производные классы Ellipse (эллипс), Hyperbola (гипербола) с собственными функциями вычисления у в зависимости

от входного параметра х. Уравнение эллипса х2 / а2 + у2 / b2 = 1; гиперболы: х2 / а2-у2 /b2= 1.

1. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Реализовать производные классы Complex (задание 1.21) и Rational (задание 1.28).
2. Создать абстрактный базовый класс Tri ad с виртуальными методами увеличе­ния на 1. Создать производные классы Date (задание 1.29) и Time (зада­ние 1.30).
3. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Создать производные классы Money (задание 1.24) и Fraction (за­дание 1.35).
4. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Реализовать производные классы Money (задание 1.24) и Money (задание 1.33).
5. Создать абстрактный базовый класс Integer (целое) с виртуальными ариф­метическими операциями и функцией вывода на экран. Определить про­изводные классы Decimal (десятичное) и Binary (двоичное), реализующие соб­ственные арифметические операции и функцию вывода на экран. Число представляется массивом, каждый элемент которого — цифра.
6. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Создать производные классы Money (задание 1.24) и Complex (за­дание 1.21).
7. Создать абстрактный базовый класс Series (прогрессия) с виртуальными функциями вычисления у-го элемента прогрессии и суммы прогрессии. Опре­делить производные классы: Linear (арифметическая) и Exponential (геомет­рическая). (Арифметическая прогрессия = а<> + jd, у = 0, 1, 2,... Сумма ариф­метической прогрессии: s„ = (п + 1) (яо + я») / 2. Геометрическая прогрессия: dj = aor’J = 0, 1, 2,... Сумма геометрической прогрессии: sn = (ао - a„r) / (1 - г).)
8. Создать абстрактный класс Norm с виртуальной функцией вычисления нормы и модуля. Определить производные классы Complex, Vector3D с собственными функциями вычисления нормы и модуля. (Модуль для комплексного числа вычисляется как корень из суммы квадратов действительной и мнимой частей; норма для комплексных чисел вычисляется как модуль в квадрате. Модуль вектора вычисляется как корень квадратный из суммы квадратов ко­ординат; норма вектора вычисляется как максимальное из абсолютных значе­ний координат.)
9. Создать абстрактный базовый класс Pair с виртуальными арифметическими операциями. Создать производные классы FazzyNumber (задание 1.31) и Fraction (задание 1.35).
10. Создать абстрактный базовый класс Container с виртуальными методами sortO и поэлементной обработки контейнера foreachO. Разработать производные классы Bubble (пузырек) и Choice (выбор). В первом классе сортировка реали­зуется методом пузырька, а поэлементная обработка состоит в извлечении

квадратного корня. Во втором классе сортировка реализуется методом выбо­ра, а поэлементная обработка — вычисление логарифма.

1. Создать абстрактный базовый класс Array с виртуальными методами сложе­ния и поэлементной обработки массива foreachO. Разработать производные классы SortArray и ХогАггау. В первом классе операция сложения реализуется как объединение множеств, а поэлементная обработка — сортировка. Во вто­ром классе операция сложения реализуется как исключающее ИЛИ, а поэле­ментная обработка — вычисление корня.
2. Создать абстрактный базовый класс Array с виртуальными методами сложе­ния и поэлементной обработки массива foreachO. Разработать производные классы AndArray и ОгАггау (выбор). В первом классе операция сложения реали­зуется как пересечение множеств, а поэлементная обработка представляет собой извлечение квадратного корня. Во втором классе операция сложения реализуется как объединение, а поэлементная обработка — вычисление лога­рифма.