

Вариант 1

1 Создать абстрактный класс **Figure** с функциями вычисления площади и периметра, а также функцией, выводящей информацию о фигуре на экран.

2 В абстрактном классе **Figure** реализовать метод **CompareTo** так, чтобы можно было отсортировать объекты по их площадям.

3 Создать производные классы: **Rectangle** (прямоугольник), **Circle** (круг), **Triangle** (треугольник).

4 В методе **Main()** создать массив *n* фигур и вывести полную информацию о фигурах на экран, отсортировав объекты по их площадям, а также организовать поиск фигур, площадь которых попадает в заданный диапазон.

Вариант 2

1 Создать абстрактный класс **Function** с функциями вычисления значения по формуле $y=f(x)$ в заданной точке, а также функцией, выводящей информацию о виде функции на экран.

2 В абстрактном классе **Function** реализовать метод **CompareTo** так, чтобы можно было отсортировать функции по коэффициенту *a*.

3 Создать производные классы: **Line** ($y=ax+b$), **Kub** ($y=ax^3+bx+c$), **Hyperbola** ($y=a/x$).

4 В методе **Main()** создать массив *n* функций и вывести полную информацию о значении данных функций в точке *x*, отсортировав функции по коэффициенту *a*.

Вариант 3

1 Создать абстрактный класс **Edition** с функциями, позволяющими вывести на экран информацию об издании, а также определить, является ли данное издание искомым.

2 В абстрактном классе **Edition** реализовать метод **CompareTo** так, чтобы можно было отсортировать каталог изданий по фамилии автора.

3 Создать производные классы: **Book** (название, фамилия автора, год издания, издательство), **Article** (название, фамилия автора, название журнала, его номер и год издания), **OnlineResource** (название, фамилия автора, ссылка, аннотация).

4 В методе **Main()** создать массив из *n* изданий, вывести полную информацию из каталога, отсортировав каталог изданий по фамилии автора, а также организовать поиск изданий по фамилии автора.

Вариант 4

1 Создать абстрактный класс **Transport** с функциями, позволяющими вывести на экран информацию о транспортном средстве, а также определить грузоподъемность транспортного средства.

- 2 В абстрактном классе `Transport` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о машинах по их грузоподъемности.
- 3 Создать производные классы: `Car` (марка, номер, скорость, грузоподъемность), `Motorbike` (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна 0), `Truck` (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в 2 раза).
- 4 В методе `Main()` создать массив из n машин, вывести полную информацию из базы на экран, отсортировав массив данных о машинах по их грузоподъемности, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

Вариант 5

- 1 Создать абстрактный класс `Persona` с функциями, позволяющими вывести на экран информацию о персоне, а также определить её возраст (на момент текущей даты).
- 2 В абстрактном классе `Persona` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о персонах по дате рождения.
- 3 Создать производные классы: `Enrollee` (фамилия, дата рождения, факультет), `Student` (фамилия, дата рождения, факультет, курс), `Teacher` (фамилия, дата рождения, факультет, должность, стаж).
- 4 В методе `Main()` создать массив из n персон, вывести полную информацию из базы на экран, отсортировав массив данных о персонах по дате рождения, а также организовать поиск персон, чей возраст попадает в заданный диапазон.

Вариант 6

- 1 Создать абстрактный класс `Goods` с функциями, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли он сроку годности на текущую дату.
- 2 В абстрактном классе `Goods` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о товарах по их цене.
- 3 Создать производные классы: `Product` (название, цена, дата производства, срок годности), `Party` (название, цена, количество штук, дата производства, срок годности), `Kit` (название, цена, перечень продуктов).
- 4 В методе `Main()` создать массив из n товаров, вывести полную информацию из базы на экран, отсортировав массив данных о товарах

по их цене, а также организовать поиск просроченного товара (на момент текущей даты).

Вариант 7

- 1 Создать абстрактный класс `Goods` с функциями, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли она искомому типу.
- 2 В абстрактном классе `Goods` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о товарах по возрасту детей, на которых он рассчитан.
- 3 Создать производные классы: `Toy` (название, цена, производитель, материал, возраст, на который рассчитана), `Book` (название, автор, цена, издательство, возраст, на который рассчитана), `SportsEquipment` (название, цена, производитель, возраст, на который рассчитан).
- 4 В методе `Main()` создать массив из n товаров, вывести полную информацию из базы на экран, отсортировав массив данных о товарах по возрасту детей, на которых он рассчитан, а также организовать поиск товаров определенного типа.

Вариант 8

- 1 Создать абстрактный класс `TelephoneDirectory` с функциями, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска.
- 2 В абстрактном классе `TelephoneDirectory` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных справочника по номеру телефона.
- 3 Создать производные классы: `Persona` (фамилия, адрес, номер телефона), `Organization` (название, адрес, телефон, факс, контактное лицо), `Friend` (фамилия, адрес, номер телефона, дата рождения).
- 4 В методе `Main()` создать массив из n записей, вывести полную информацию из базы на экран, отсортировав массив данных справочника по номеру телефона, а также организовать поиск в базе по фамилии.

Вариант 9

- 1 Создать абстрактный класс `Client` с функциями, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска.
- 2 В абстрактном классе `Client` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о клиентах банка по дате открытия их счета.

3 Создать производные классы: **Depositor** (фамилия, дата открытия вклада, размер вклада, процент по вкладу), **Credited** (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), **Organization** (название, дата открытия счета, номер счета, сумма на счету).

4 В методе **Main()** создать массив из *n* клиентов, вывести полную информацию из базы на экран, отсортировав массив данных о клиентах банка по дате открытия их счета, а также организовать поиск клиентов, начавших сотрудничать с банком с заданной даты.

Вариант 10

1 Создать абстрактный класс **Software** с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).

2 В абстрактном классе **Software** реализовать метод **CompareTo** так, чтобы можно было отсортировать массив данных по названию ПО.

3 Создать производные классы: **FreeSoftware** (название, производитель), **SharewareSoftware** (название, производитель, дата установки, срок бесплатного использования), **ProprietarySoftware** (название, производитель, цена, дата установки, срок использования).

4 В методе **Main()** создать массив из *n* видов программного обеспечения, вывести полную информацию из массива на экран, отсортировав массив данных по названию ПО, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.