**Лабораторная работа № 1 (I часть)**

**Тема:** Разработка структур данных для программ, использующих классы

**Цель работы:** получить практические навыки разработки иерархии классов, методов, свойств, операций классов.

*Указания:*

Требуется описать базовый класс (возможно, абстрактный) и 1-3 наследованных класса. Каждый разрабатываемый класс должен находиться в отдельном модуле (файле) в составе проекта. Для добавления нового модуля в проект выберите пункт меню «Проект» → «Добавить класс» и укажите название класса.

Класс должен содержать: поля, конструкторы, свойства, методы (или перегруженные методы).

*Разработка структур данных должна производиться по следующему плану:*

1. Объявите базовый класс и опишите его поля. **Поля** должны быть скрыты (доступ private или protected).
2. Опишите конструкторы базового класса (их может быть несколько).
3. Опишите простые методы базового класса (если они есть). **Методы,** которые представляют интерфейсную часть класса, должны быть открыты (доступ public).
4. Продумайте, какие методы должны быть виртуальными и будут перегружены в дочерних классах. Опишите эти методы, пометив их словом virtual.
5. По условию задачи необходимо будет получать или задавать значения некоторых полей. Опишите **свойства**, связанные с этими полями, доступные для чтения (get) и (или) для установки (set).
6. Объявите дочерний класс и опишите его дополнительные поля. Дочерний класс наследует поля и методы базового класса.
7. Опишите конструкторы дочернего класса (т.к. у каждого класса должен быть свой конструктор).
8. Опишите дополнительные (новые) и перегруженные методы дочернего класса. Перегруженные методы пометьте словом override.

**Пример описания классов:**

namespace Monster

{

class Monstr // базовый класс Монстр

{

string name; // поля класса

int power;

int health;

public Monstr() // конструктор по умолчанию

{

this.name = "Noname";

this.power = 100;

this.health = 100;

}

public Monstr(string name, int power, int health) // конструктор

{

this.name = name;

this.power = power;

this.health = health;

}

virtual public void Show()// метод для вывода информации на экран

{

if (this is Demon)

Console.Write("\tДемон: {0}\t", name);

if (this is Monster)

Console.Write("\tМонстр: {0}\t", name);

Console.Write("Сила: {0}\tЗдоровье: {1}", power, health);

}

}

class Demon : Monstr // порождённый класс Демон

{

int brain; // демон имеет ум

public Demon(string name, int power, int health, int brain): base(name, power, health) // конструктор для Демона

{

this.brain = brain;

}

override public void Show() // перегруженный метод для вывода информации о Демоне

{

base.Show();

Console.Write("\tУм: {0}", brain);

}

public int Brain // метод для проверки ума

{

get

{

return brain;

}

set

{

brain = value;

}

}

}

class Program

{

static void Main()

{

…// здесь создание объектов и вызов методов

}

}

}

**ВАРИАНТЫ:**

**Вариант 1**

Создать абстрактный класс Transport (транспортное средство) и на его основе два производных класса Plain (самолёт) и Ship (корабль). Классы должны иметь возможность получать параметры (название, количество мест, скорость, год выпуска, для самолёта – высота, для корабля – порт приписки) средств передвижения, а также вывода на экран сведений о транспортном средстве. Организовать вывод на экран сведений из массива обо всех самолётах и кораблях, вместимость которых больше заданной.

**Вариант 2**

Создать абстрактный класс Point (точка), содержащий поля X, Y с координатами точки, и на его основе класс ColoredPoint (цветная точка) с дополнительным полем Color (цвет) и Line (линия) с дополнительными полями Length (длина линии) и Ugol (угол наклона линии к оси X). Описать методы для вывода на экран сведений об объекте, получения и установки координат, а также для изменения цвета и получения текущего цвета.

**Вариант 3**

Создать базовый класс Plain (самолёт), содержащий поля: тип самолёта, количество пассажиров. Создать производный класс Fight (рейс), содержащий дополнительные поля: номер рейса, название пункта назначения. Описать методы для вывода на экран сведений об объекте, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива обо всех самолётах, вылетающих в пункт назначения, введённый с клавиатуры (если таких самолётов нет, то вывести соответствующее сообщение).

**Вариант 4**

Создать базовый класс Woker (работник), содержащий поля: ФИО работника, должность, год поступления на работу. Создать производный класс Teacher (преподаватель) с дополнительными полями нагрузка (то есть количество часов на учебный год) и категория. Описать методы для вывода на экран сведений об объекте, получения и установки значений некоторых полей. Организовать расчёт заработной платы: для простых работников с учётом 8-ми часового рабочего дня и почасовой ставки для конкретной должности, для преподавателей – с учётом нагрузки и категории.

**Вариант 5**

Создать базовый класс Tovar (товар), содержащий поля: код товара, название, цена. Создать производный класс Product (продукт) c дополнительным полем срок реализации продукта (то есть сколько суток он годен). Описать методы для вывода на экран сведений об объекте, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива обо всех товарах, цена которых лежит в указанном диапазоне.

**Вариант 6**

Создать базовый класс Note (запись), содержащий поля: ФИО человека, номер телефона. Создать производный класс Friend (друг), содержащий дополнительные поля: электронная почта, дата рождения (массив из трёх чисел). Описать методы для вывода на экран записей, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива обо всех друзьях, родившихся в указанный месяц (если таких нет, то вывести соответствующее сообщение).

**Вариант 7**

Создать базовый класс Circle (круг), содержащий поле: радиус. Создать производные классы Cone (конус) и Cylinder (цилиндр), содержащие дополнительное поле: высота. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади поверхности и объёма фигур для производных классов. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

**Вариант 8**

Создать базовый класс Rectangle (прямоугольник), содержащий поля: длина, ширина. Создать производные классы Parallelogram (параллелограмм) и Trapeze (равнобедренная трапеция), содержащие дополнительное поле: угол наклона. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади фигур. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

**Вариант 9**

Создать базовый класс Point (точка), содержащий поля: x, y (координаты точки). Создать производный класс Circle (круг), содержащий дополнительное поле: радиус, а от класса Circle производный класс Ellipse (эллипс), содержащий ещё одно дополнительное поле: второй радиус. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади фигур. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

**Вариант 10**

Создать базовый класс Rectangle (прямоугольник), содержащий поля: длина, ширина. Создать производные классы Parallelepiped (параллелепипед) и Pyramid (пирамида), содержащие дополнительное поле: высота. Описать методы для вывода на экран сведений об объекте, получения и установки значений полей, методы вычисления площади поверхности и объёма фигур. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

**Вариант 11**

Создать базовый класс Patch (путь), содержащий поля: пункт назначения, длина пути (в км). Создать производный класс Marsh (туристический маршрут), содержащий дополнительные поля: номер маршрута, уровень сложности, количество дней в пути. Описать методы для вывода на экран сведений о маршрутах, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива о маршрутах, уровень сложности которых не превышает заданный.

**Вариант 12**

Создать базовый класс Note (запись), содержащий поля: ФИО человека, номер телефона. Создать производный класс Woker (сотрудник), содержащий дополнительные поля: электронная почта, должность. Описать методы для вывода на экран записей, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива обо всех людях, фамилия которых начинается с указанной буквы (если таких нет, то вывести соответствующее сообщение).

**Вариант 13**

Создать базовый класс Book (книга), содержащий поля: ФИО автора, название, год издания, категория. Создать производный класс Electron\_Book (электронная книга), содержащий дополнительное поле: электронный источник. Описать методы для вывода на экран сведений о книгах, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива о книгах, обычных и электронных, указанного автора (если таких нет, то вывести соответствующее сообщение).

**Вариант 14**

Создать абстрактный класс Person (человек), содержащий поля: ФИО человека, год рождения. Создать производные классы Doctor (доктор) и Patient (пациент), содержащие дополнительные поля: для обоих производных классов номер отделения больницы, для докторов специализация, для пациентов дата поступления в больницу и дата выписки (если пациент выписался из больницы). Описать методы для вывода на экран сведений о врачах и пациентах больницы, получения и установки значений некоторых свойств. Организовать вывод на экран сведений из массива обо всех людях, врачах и пациентах, находящихся в указанном отделении (если таких нет, то вывести соответствующее сообщение).

**Вариант 15**

Создать базовый класс Triangle (равносторонний треугольник), содержащий поле: сторона треугольника. Создать производные классы Tetrahedron (тетраэдр), Octahedron (октаэдр), Icosuhedron (икосаэдр). Описать методы для вывода на экран названия правильного многогранника и сведений об объекте, получения значений полей, методы вычисления площади поверхности фигур. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Многогранник** | | **Вершины** | **Рёбра** | **Грани** |
| тетраэдр | [Тетраэдр](https://commons.wikimedia.org/wiki/File:Tetrahedron.svg?uselang=ru) | 4 | 6 | 4 |
| [октаэдр](https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%82%D0%B0%D1%8D%D0%B4%D1%80) | [Октаэдр](https://commons.wikimedia.org/wiki/File:Octahedron.svg?uselang=ru) | 6 | 12 | 8 |
| [икосаэдр](https://ru.wikipedia.org/wiki/%D0%98%D0%BA%D0%BE%D1%81%D0%B0%D1%8D%D0%B4%D1%80) | [Икосаэдр](https://commons.wikimedia.org/wiki/File:Icosahedron.svg?uselang=ru) | 12 | 30 | 20 |

**Вариант 16**

Создать абстрактный класс Car (автомобиль), содержащий поля: госномер, марка, цвет. Создать производный класс Avtostoyanka (автостоянка), содержащий дополнительные поля: фамилия владельца, признак присутствия автомобиля на стоянке. Описать методы для вывода на экран сведений об автомобилях, получения и установки значений некоторых полей. Организовать вывод на экран сведений из массива об автомобилях, присутствующих на стоянке.

**Вариант 17**

Создать базовый класс BaseString (простая строка), содержащий поля: поле для хранения символов строки, длина строки. В базовом классе описать методы: конструктор без параметров, конструктор, принимающий в качестве параметра строку, метод получения длины строки, метод очистки строки (сделать строку пустой).

Создать производный класс OctString (восьмеричная строка). Строки данного класса могут содержать только символы “01234567”. Если в составе инициализирующей строки встретятся другие символы, то строка принимает пустое значение. В производном классе описать методы: конструктор, принимающий в качестве параметра строку, проверка на равенство двух строк, перевод восьмеричного числа в десятичную систему счисления.

**Вариант 18**

Создать базовый класс BaseString (простая строка), содержащий поля: поле для хранения символов строки, длина строки. В базовом классе описать методы: конструктор без параметров, конструктор, принимающий в качестве параметра строку, метод получения длины строки, метод очистки строки (сделать строку пустой).

Создать производный класс BinString (двоичная строка). Строки данного класса могут содержать только символы “01”. Если в составе инициализирующей строки встретятся другие символы, то строка принимает пустое значение. В производном классе описать методы: конструктор, принимающий в качестве параметра строку, проверка на равенство двух строк, перевод двоичного числа в десятичную систему счисления.

**Вариант 19**

Создать базовый класс BaseString (простая строка), содержащий поля: поле для хранения символов строки, длина строки. В базовом классе описать методы: конструктор без параметров, конструктор, принимающий в качестве параметра строку, метод получения длины строки, метод очистки строки (сделать строку пустой).

Создать производный класс ComplexString (комплексная строка). Строки данного класса состоят из двух полей, разделённых знаком “i”. Первое поле задаёт значение действительной части числа, второе поле – значение мнимой части. Каждое поле может содержать десятичные цифры и знак + или –, задающие знак числа, только в первой позиции строки. Если в составе инициализирующей строки встретятся другие символы, то строка принимает пустое значение. В производном классе описать методы: конструктор, проверка на равенство двух комплексных чисел, сложение, умножение комплексных чисел.

**Вариант 20**

Создать базовый класс Simetric (правильный многоугольник), содержащий поля: количество вершин, сторона многоугольника. Создать производные классы Tangle (равносторонний треугольник), Kvadrat (квадрат), FiveSimetric (правильный пятиугольник), SixSimetric (правильный шестиугольник). Описать методы для вывода на экран названия фигуры, сведений об объекте, методы вычисления площади фигур, радиуса описанной и вписанной окружности. Показать вызов методов на примере массива базового типа, содержащего ссылки на объекты производных типов.

Площадь правильного многоугольника с числом сторон n и длиной стороны a : S = \frac{n}{4}\ a^2 \mathop{\mathrm{}}\, \operatorname{ctg} \frac{\pi}{n}

Длина стороны многоугольника через радиус описанной и вписанной окружности: a =  2R \sin \frac{\pi}{n} =  2r \mathop{\mathrm{tg}}\, \frac{\pi}{n}