**Лабораторная работа № 6**

**Тема:** Разработка и отладка программ демонстрации полиморфизма

**Цель работы:** получить практические навыки определения межклассовых отношений и разработки программ демонстрации явления полиморфизма.

*Указания:*

Условие задачи по своему варианту смотри в предыдущей работе (лабораторная работа № 5). В предыдущей работе требовалось описать базовый (родительский) класс и 1-3 наследованных класса. Главная функция класса Program или особый класс должен содержать массив или список, объявленный базовым типом, и заполненный ссылками на объекты базового и (или) производных классов. Программа должна демонстрировать вызов методов разных классов, в том числе правильный выбор переопределенных методов (принцип полиморфизма).

**Разработка должна вестись по следующему плану:**

1. Проверьте правильность описания базового и дочерних классов. В каждом классе должен быть описан свой конструктор. В дочерних классах **не могут** повторяться те же поля, что и в родительском классе, т.к. они наследуются.
2. Некоторые виртуальные методы могут быть переопределены в дочерних классах. Переопределённые методы пометьте словом override.
3. В случае необходимости опишите одно или несколько свойств для получения и (или) установки значений полей. Свойство описываются внутри того класса, где задано объявлено скрытое поле, с которым оно связано.
4. Постройте диаграмму классов в графическом редакторе MS Visio. Выбираем раздел фигур **Дополнительные фигуры→ Программы и базы данных→ Программное обеспечение → Класс UML.**



Рисунок 1 Пример диаграммы классов

1. Начните разработку главной функции. Объявите массив или список типом родительского класса. Организуйте заполнение массива объектами с учётом, что реально он может заполняться элементами как родительского, так и дочерних классов (или только дочерних).
2. В некоторых вариантах массив или список является включаемым полем другого класса. Такой вид отношений может быть агрегацией или композицией. Тогда принцип полиморфизма будет демонстрироваться в этом классе. Включаемое поле массив (или список) объектов может принимать объекты разных дочерних типов.
3. Организуйте вывод на экран полученного массива или списка при помощи отдельного цикла, в котором будет вызываться метод Show().
4. Выполните остальные задания по обработке данных. Работу с массивом или списком объектов организуйте при помощи меню.

**Пример меню:** вывод всего списка, добавление объектов в список, удаление объекта из списка, поиск объекта по определенному полю или нескольким полям, выход из программы.

**Пример оформления главной функции:**

namespace MyGame

{

class Program

{

static void Main(string[] args)

{

Monster[] stado = new Monster[4];

Console.WriteLine("Введите имя своего персонажа:");

Monster myMonster = new Monster(Console.ReadLine()); // мой игровой персонаж

myMonster.Show();

monsters[0] = new Monster("Арагон"); // боты

monsters[1] = new Monster("Гора");

monsters[2] = new Demon("Сатана");

monsters[3] = new Demon("Цербер");

for (i = 0; i < 4; i++)

{

stado[i].Show();

}

Console.ReadKey();

}

}

**Требования к оформлению отчета:**

По лабораторной работе 6 оформите *отчет*. Отчет должен содержать:

1. Название и тему лабораторной работы;
2. Цель работы;
3. Условие задачи по вашему варианту;
4. Диаграмму классов с указанием полей и их типов, методов каждого класса и отношений классов;
5. Текст всех программных модулей с комментариями. Обязательно прокомментируйте назначение всех методов;
6. Скриншоты выполнения программы.

**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 (перечисляемый тип)), а также иметь метод вывода на экран сведений о транспортном средстве. Организовать вывод на экран сведений из списка 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) и дочерний класс Circle (окружность), причем окружность характеризуется точкой своего центра и радиусом. Описать методы для вывода на экран сведений об объекте, получения и установки координат точки, а также для изменения цвета и получения текущего цвета и прорисовки всех видов объектов в графическом режиме. Создать массив объектов (цветных линий и окружностей) и выполнить их прорисовку на экране консоли.

**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. Описать класс Airport (аэропорт), содержащий включаемое поле массив рейсов. Создать метод добавления рейса, метод удаления рейса, а также метод вывода всего списка на экран. В основной программе продемонстрировать создание объекта аэропорт и работу со списком рейсов.

**Вариант 4**

Создать базовый класс Woker (работник), содержащий поля: ФИО работника, должность (перечисляемый тип), 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. Описать класс Personnel (персонал) с включенным полем массив работников, и методами добавления и удаления работников в массив объектов, а также вывода на экран списка персонала и зарплаты каждого работника.

**Вариант 5**

Создать базовый класс Tovar (товар), содержащий поля: код товара, название, цена, количество на складе. Создать конструктор класса, метод для вывода на экран сведений об объекте товар, а также методы для увеличения количества товара на 1 и уменьшения количества товара на 1.

Создать производный класс Product (продукт) c дополнительным полем срок реализации продукта (то есть сколько суток он годен). Создать конструктор класса, метод для вывода на экран сведений об объекте.

Описать класс Order (заказ), у которого имеются следующие поля: номер заказа, дата и время заказа, статус заказа (формируется или готов), а также включаемое поле массив товаров с указанием количества каждой позиции. Создать метод добавления товара в заказ (товары выбираются из заранее созданного списка имеющихся товаров), метод увеличения или уменьшения количества данного товара в заказе на 1, метод удаления товара из заказа, а также метод окончательного формирования заказа с расчётом полной стоимости заказа, и метод вывода заказа на экран. В основной программе продемонстрировать создание заказа.

**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. Описать методы для вывода на экран записей, получения и установки значений некоторых полей. Описать класс AdressBook (телефонная книга), содержащий включаемое поле массив записей Note. Создать метод добавления записи в телефонную книгу, метод удаления записи из телефонной книги, а также метод вывода всего списка на экран. В основной программе продемонстрировать создание телефонной книги.

**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. Создать массив объектов и продемонстрировать вызов методов для них.

**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. Создать массив объектов и продемонстрировать вызов методов для них.

**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. Создать массив объектов и продемонстрировать вызов методов для них.

**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. Создать массив объектов и продемонстрировать вызов методов для них.

**Вариант 11**

Создать базовый класс Patch (путь), содержащий поля: пункт назначения, длина пути (в км). Создать производный класс Marsh (туристический маршрут), содержащий дополнительные поля: название маршрута (перечисляемый тип), уровень сложности, количество дней в пути. Описать методы для вывода на экран сведений о маршрутах, получения и установки значений некоторых полей. Создать список маршрутов. Организовать вывод на экран сведений из списка маршрутов, 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 (перечисляемый тип). Описать методы для вывода на экран записей, получения и установки значений некоторых полей. Создать список объектов. Организовать вывод на экран сведений из списка объектов 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 (перечисляемый тип), категория. Создать производный класс ElectronBook (электронная книга), содержащий дополнительное поле: электронный источник. Описать методы для вывода на экран сведений о книгах, получения и установки значений некоторых полей. Описать класс Library (библиотека), содержащий включаемое поле массив книг. Создать метод добавления книги, метод удаления книги, а также метод вывода всего списка на экран. В основной программе продемонстрировать создание объекта библиотека и работу со списком книг.

**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 отделения больницы (перечисляемый тип), 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. Создать список объектов. Организовать вывод на экран сведений из списка 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. Создать массив объектов и продемонстрировать вызов методов для них.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 (перечисляемый тип), цвет. Создать класс Avtostoyanka (автостоянка), содержащий поля: фамилия владельца, автомобиль (включаемое поле), признак присутствия автомобиля на стоянке. Описать методы для вывода на экран сведений об автомобилях, получения и установки значений некоторых полей. Создать массив объектов класса Avtostoyanka (автостоянка). 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, проверка на равенство двух строк, 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, перевод восьмеричного числа в десятичную систему счисления. Создать массив объектов и продемонстрировать вызов методов для них.

**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, проверка на равенство двух строк, 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, перевод двоичного числа в десятичную систему счисления. Создать массив объектов и продемонстрировать вызов методов для них.

**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 частей, разделённых знаком «i». Первая половина строки задаёт значение действительной части числа, вторая половина после знака «i» – значение мнимой части. Каждая часть инициализирующей строки 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: дополнительные поля, хранящие действительную и мнимую часть числа в виде целочисленных значений, конструктор, принимающий строку, конструктор, принимающий на вход два целочисленных значения (действительную и мнимую часть числа в целом виде), проверка на равенство двух комплексных чисел, сложение, вычитание, умножение комплексных чисел. Создать массив объектов и продемонстрировать вызов методов для них.

**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. Создать массив объектов и продемонстрировать вызов методов для них.