# ПРАКТИЧЕСКАЯ РАБОТА № 11

**Вложенные объекты в классе**

**Цель работы**: научиться использовать в классе в качестве полей объекты других классов.

|  |  |  |
| --- | --- | --- |
| №  п/п | Ф.И.О. | НОМЕР ВАРИАНТА/ЗАДАНИЯ |
|  | Андриуца Дарья Николаевна | Набор примера из лекционного материала+ ДОБАВЛЕНИЕ 2 ПОЛЕЙ |
|  | Варфоломеев Иван Константинович | 1 |
|  | Верховин Игорь Вадимович | 2 |
|  | Власов Тимур Михайлович | 3 |
|  | Гагауз Григорий Иванович | 4 |
|  | Голубцов Олег Русланович | Набор примера из лекционного материала + ДОБАВЛЕНИЕ 2 МЕТОДОВ |
|  | Гувирь Максим Григорьевич | Набор примера из лекционного материала + ДОБАВЛЕНИЕ 2 ПОЛЕЙ |
|  | Дорожкина Алина Игоревна | 5 |
|  | Жупанов Юлиан Игоревич | 6 |
|  | Капелько Иван Юрьевич | 4 |
|  | Кирста Андрей Александрович | 7 |
|  | Ковалев Иван Алексеевич | 8 |
|  | Лютов Глеб Васильевич | 9 |
|  | Михалаки Василий Федорович | 10 |
|  | Москалюк Иван Сергеевич | 11 |
|  | Огурцов Алексей Михайлович | 12 |
|  | Омельченко Евгений Борисович | 13 |
|  | Полякова Анна Игоревна | 14 |
|  | Редкозубов Владислав Романович | 15 |
|  | Ротарь Федор Федорович | 5 |
|  | Рошко (был Брага) Марин Михайлович | 6 |
|  | Салкуцан Анна Александровна | 14 |
|  | Скутельник Владимир Александрович | 16 |
|  | Струсевич Лев Дмитриевич | 10 |
|  | Телицкий Александр | 9 |
|  | Чернюк Егор Дмитриевич | 8 |
|  | Щитченко Николь Александровна | 7 |

# Теоретические сведения

Рассмотрим два класса *CLAX* и *CLBX*, связанных отношением вложенности. Оба класса имеют демонстрационный характер и поэтому не несут смысловой нагрузки. Рассмотрим класс *CLAX*. Он имеет два поля, конструктор, один статический и один динамический метод. Вот его текст:

*public class CLAX{*

*public CLAX(string f1, int f2) { fieldA1 = f1; fieldA2 = f2;*

*}*

*public string fieldA1; public int fieldA2;*

*public void MethodA() { Console.WriteLine( "Это класс A");*

*Console.WriteLine ("поле1 = {0}, поле2 = {1}", fieldA1, fieldA2);*

*}*

*public static void StatMethodA() {*

*string s1 = "Статический метод класса А"; string s2 = "Помните: 2\*2 = 4"; Console.WriteLine(s1 + " \*\*\*\*\* " + s2);*

*}}*

Построим теперь класс *BX* - клиента класса *AX*. Класс будет устроен похожим образом, но в дополнение будет иметь одним из своих полей объект *inr* класса *AX*:

*public class CLBX {*

*public CLBX(string f1A, int f2A, string f1B, int f2B) { innr = new CLAX (f1A, f2A);*

*fieldB1 = f1B; fieldB2 = f2B;*

*}*

*CLAX inner;*

*public string fieldB1; public int fieldB2;*

*public void MethodB1() { innr.MethodA();*

*Console.WriteLine( "Это класс B");*

*Console.WriteLine ("поле1 = {0}, поле2 = {1}", fieldB1, fieldB2);*

*}*

Здесь хотелось бы обратить внимание на то, что конструктор клиента (класса *CLBX*) отвечает за инициализацию полей класса, поэтому он должен создать объект поставщика (класса *CLAX*), вызывая, как правило, конструктор поставщика. Если для создания объектов поставщика требуются аргументы, то они должны передаваться конструктору клиента, как это сделано в нашем примере.

После того как конструктор создал поле - объект поставщика - методы класса могут использовать этот объект, вызывая доступные клиенту методы и поля класса поставщика. Метод класса *CLBX* - *MethodB1* начинает свою работу с вызова: *inn*.*MethodA*, используя сервис, поставляемый методом класса *CLAX*.

До сих пор мы рассматривали ситуацию, когда клиент содержит поле, представляющее объект класса поставщика. Это частая, но не единственная ситуация, когда класс является клиентом другого класса. Возможна ситуация, когда метод клиентского класса локально создает объект поставщика, вызывает его методы в собственных целях, но по завершении работы метода локальный объект заканчивает свое существование. Еще одна возможная ситуация - когда объекты поставщика вообще не создаются ни конструктором, ни методами класса клиента, но клиент вызывает статические методы класса поставщика. Оба эти варианта демонстрируют следующие два метода класса *CLBX*:

*public void MethodB2(){*

*CLAX loc = new CLAX("локальный объект А",77); loc.MethodA();*

*}*

*public void MethodB3(){ CLAX.StatMethodA();*

*}*

Сформулируем расширенное определение клиента.

**Определение 3.** Класс BX называется клиентом класса *AX*, если в

классе *BX* создаются объекты класса *AX* - поля или локальные переменные -

или вызываются статические поля или методы класса *AX*.

Класс-поставщик создает свойства (поля) и сервисы (методы), предоставляемые своим клиентам. Клиенты создают объекты поставщика. Вызывая доступные им методы и поля объектов, они управляют работой созданных объектов поставщика. Клиенты не могут ни повлиять на поведение методов поставщика, ни изменить состав предоставляемых им полей и методов, они не могут вызывать закрытые поставщиком поля и методы класса.

Класс-поставщик доступен клиентам своей открытой частью, составляющей интерфейс класса. Но большая часть класса может быть закрыта для клиентов - им незачем вникать в детали представления и в детали реализации. В общем случае сокрытие означает, что классы-клиенты строят свою реализацию, основываясь только на интерфейсной части класса- поставщика. Поставщик закрывает поля и часть методов класса от клиентов, задавая для них атрибут доступа *private* или *protected*. Он может некоторые классы считать привилегированными, предоставляя им методы и поля, недоступные другим классам. В этом случае поля и методы, предназначенные для таких *vip*-персон, снабжаются атрибутом доступа *internal*, а классы с привилегиями должны принадлежать одной сборке.

В заключение рассмотрим тест, проверяющий работу с объектами классов *CLAX* и *CLBX*:

*public void TestClientSupplier(){*

*CLBX objB = new CLBX("AA",22, "BB",33);*

*objB.MethodB1(); objB.MethodB2(); objB.MethodB3();*

*}*

Зададим следующие вопросы:

* 1. может ли класс быть сам себе клиентом, т.е. может ли поле класса быть объектом этого же класса;
  2. могут ли два класса быть одновременно клиентами и поставщиками друг для друга?

Ответы на оба вопросы положительны, и подобные ситуации типичны и не являются какой-либо экзотикой.

Первая ситуация характерна для динамических структур данных.

Элемент односвязного списка имеет поле, представляющее элемент односвязного списка; элемент двусвязного списка имеет два таких поля; узел двоичного дерева имеет два поля, представляющих узлы двоичного дерева.

Эта ситуация характерна не только для рекурсивно определяемых структур данных.

Рассмотрим следующий пример. В классе *Pers* могут быть заданы два поля - *Father* и *Mother*, задающие родителей персоны, и массив *Children*.

Очевидно, что все эти объекты относятся к одному и тому же классу *Pers*.

Не менее часто встречается ситуация, когда классы имеют поля, взаимно ссылающиеся друг на друга. Типичным примером могут служить классы *Man* и *Woman*, первый из которых имеет поле *wife* класса *Woman*, а второй - поле *husband* класса *Man*.

Как правило, классы устроены достаточно просто - их тексты понятны, отношения между классами очевидны. Гораздо сложнее выглядит динамический мир объектов - может быть гораздо более сложным, а отношения между объектами могут быть запутанными.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

**Класс Сессия**. Вложенные объекты: **группа, предмет, аудитория, преподаватель**.

Сформировать расписание сессии (4 экзамена). Выдать на экран по требованию оператора:

* + данные о том, когда какой экзамен, какому преподавателю сдает группа;
  + список группы;
  + данные об аудитории, где принимают экзамен (номер, корпус, количество мест);
  + данные о преподавателе, принимающем экзамен;
  + данные о группе, сдающей экзамен (факультет, курс, специальность, кол- во студентов).

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс **Завод.** Вложенные объекты**: цех1, цех2, склад1, склад2, заводоуправление, изделие1, изделие2, изделие3**.

По требованию оператора выдать на экран информацию о:

* + выбранном цехе (название, ФИО начальника, кол-во работающих, что выпускает);
  + выбранном складе (название, ФИО начальника, кол-во работающих, что хранит);
  + предприятии (название, ФИО директора, продукция, кол-во работающих и т.п.).

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс **Автомобиль**. Вложенные объекты: **кузов, двигатель, владелец**, **сведения о страховке**.

По требованию оператора выдать информацию о:

* + дате изготовления автомобиля;
  + технических параметрах;
  + владельце;
  + данные о страховке;
  + и т.п.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **ЭВМ**. Вложенные объекты: **материнская плата, винчестер, процессор, монитор, видеокарта, владелец**.

Программа по требованию оператора должна выдать информацию о:

* технических параметрах ЭВМ;
* кому принадлежит ЭВМ;
* выполненные ремонты ЭВМ;
* сколько лет в эксплуатации (в отдельности для: системного блока, винчестера, монитора, видеокарты ).

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс **Корабль**. Вложенные объекты: **порт приписки**, **компания- владелец**, **экипаж**, **технические данные корабля**.

Программа, по требованию оператора должна выдать информацию о:

* + названии корабля;
  + технических параметрах корабля;
  + кому принадлежит корабль;
  + экипаже корабля;
  + порт приписки;
  + последний рейс, который совершил корабль.

Программа должна обеспечить возможность изменения некоторых

данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

# Класс Экипаж авиалайнера. Вложенные объекты: самолет, командир, второй пилот, штурман, стюарт, список пассажиров (до 10 человек), авиакомпания.

Программа по требованию оператора должна выдать информацию о:

* + членах экипажа;
  + пассажирах;
  + авиакомпании;
  + технических параметрах самолета.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс **Тест**. Вложенные объекты: **вопрос1, вопрос2, вопрос3, ответы**.

Каждый объект-вопрос должен содержать номер правильного варианта ответа, а также количество баллов за правильный ответ. Объект **ответы** должен содержать номера выбранных пользователем ответов.

Программа по требованию оператора должна:

* выдать информацию о вопросе и вариантах ответов;
* ввести номер выбранного варианта ответа;
* вывести результат тестирования (кол-во правильных и неправильных ответов, кол-во набранных баллов).

Программа должна обеспечить возможность изменения некоторых данных

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы

работоспособны.

Класс **Самолет**. Вложенные объекты: **технические данные, экипаж, рейс, места**.

Программа по требованию оператора должна:

* выдать информацию технических параметрах самолета;
* выдать информацию об экипаже;
* выдать информацию о количестве занятых и свободных мест (отдельно в эконом - классе, отдельно в бизнес - классе);
* выдать информацию о выполняемом рейсе.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **Организация**. Вложенные объекты: **город**, **адрес**, **здание**. Программа по требованию оператора должна:

* выдать информацию о городе, в котором расположена организация;
* выдать информацию о здании, в котором расположена организация;
* выдать информацию об адресе организации, ее телефонах, руководстве;
* выдать информацию о том, чем занимается эта организация.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **Книга**. Вложенные объекты: **издательство, автор, переплет, жанр**. Программа по требованию оператора должна:

* выдать информацию об авторе;
* выдать информацию об издательстве;
* выдать информацию о переплете;
* выдать информацию о книге (кол-во страниц, стоимость, тираж);
* выдать информацию о жанре книги.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **Танк**. Вложенные объекты: **командир, механика-водитель, стрелок, боеприпасы**.

Программа по требованию оператора должна:

* выдать информацию об экипаже танка (ФИО, личные данные, должности);
* выдать информацию о наличии боеприпасов;
* выдать информацию о наличии горючего;
* выдать информацию о техническом состоянии танка.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **Цех**. Вложенные объекты: **изделия, работники, комплектующие**. Программа по требованию оператора должна:

* выдать информацию об изделиях, изготавливаемых в данном цехе;
* выдать информацию о работающих в этом цехе с указанием должности;
* выдать информацию о комплектующих, их количестве для указанного изделия;
* выдать информацию о поставщиках комплектующих.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование

разработанных классов, убедившись в том, что все методы работоспособны. Класс **Комната**. Вложенные объекты: **мебель, сотрудник, ЭВМ**. Программа по требованию оператора должна:

* выдать информацию о мебели, находящейся в этой комнате;
* выдать информацию о сотрудниках, работающих в этой комнате;
* выдать информацию об ЭВМ, установленных в этой комнате;
* выдать информацию о принадлежности этой комнаты отделу, цеху и т.п.
* выдать информацию о размерах комнаты (ширина, высота, длина, площадь, количестве окон, дверей).

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс **Сотрудник**. Вложенные объекты: **образование, квалификация, место жительства, место работы**.

Программа по требованию оператора должна:

* выдать информацию о ФИО сотрудника;
* выдать информацию о членах семьи;
* выдать информацию об образовании сотрудника (что и когда закончил, специальность по диплому и т.п);
* выдать информацию о месте жительства сотрудника, его телефоне и т.п.
* выдать информацию о месте работы (название организации, структурного подразделения, должности, окладе и т.п.);
* выдать информацию о квалификации сотрудника (наличие категории, сроке ее действия, срок получения и т.п.).

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование

разработанных классов, убедившись в том, что все методы работоспособны. Класс **Город**. Вложенные объекты: **улица, предприятие, прочие данные**.

Программа по требованию оператора должна:

* выдать информацию о названии города, его расположении, количестве жителей;
* выдать информацию о списке улиц;
* выдать информацию о выбранной оператором улице;
* выдать информацию о списке предприятий;
* выдать информацию об истории города.

Программа должна обеспечить возможность изменения некоторых данных.

1. Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны. Класс **Журнал**. Вложенные объекты: **статья, редакция, издательство**. Класс **Статья** содержит вложенный объект класса **Автор**.

Программа по требованию оператора должна:

* выдать информацию о списке статей данного номера;
* выдать информацию об авторах данного номера;
* выдать краткое содержимое указанной статьи (до 10 строк).
* выдать информацию о выбранном авторе;
* выдать информацию о редакции (главный редактор и т.п.);
* выдать информацию об издательстве.

Программа должна обеспечить возможность изменения некоторых данных.