

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Филиал
«Минский радиотехнический колледж»

Учебный предмет
«Конструирование программ и языка программирования»

Инструкция
по выполнению лабораторной работы №8
«Разработка, отладка и испытание программ с использованием классов»

Минск 2024 г.

Лабораторная работа № 8

Тема работы: «Разработка, отладка и испытание программ с использованием классов»

1 Цель работы

Сформировать умения разрабатывать алгоритмы и реализовывать программы по работе с классами.

2 Задание

Решить задачи согласно номеру варианта.

1. Создайте класс "Сотрудник" с полями "Имя", "Возраст" и "Зарплата". Создайте экземпляр класса и вызовите методы передав полям значения.

2. Создайте класс "Автомобиль" с полями "Марка", "Модель" и "Год выпуска". Создайте экземпляр класса и вызовите методы передав полям значения.

3. Создайте класс "Автомобиль". Добавьте в класс "Автомобиль" метод, который выводит информацию об автомобиле на экран.

4. Создайте класс "Книга" с полями "Название", "Автор" и "Год публикации". Создайте экземпляр класса и вызовите методы передав полям значения.

5. Создайте класс "Книга". Добавьте в класс "Книга" метод, который проверяет, является ли книга бестселлером (например, если год публикации после 2010 года).

6. Создайте класс "Студент" с полями "Имя", "Возраст" и "Средний балл". Создайте экземпляр класса и вызовите методы передав полям значения.

7. Создайте класс "Студент". Добавьте в класс "Студент" метод, который выводит информацию о студенте на экран.

8. Создайте класс "Фракция" с полями "Название" и "Количество членов". Создайте экземпляр класса и вызовите методы передав полям значения.

9. Создайте класс "Фракция". Добавьте в класс "Фракция" метод, который выводит информацию о фракции на экран.

10. Создайте класс "Заказ" с полями "Номер", "Дата" и "Сумма". Создайте экземпляр класса и вызовите методы передав полям значения.

11. Создайте класс "Страна" с полями "Название", "Столица" и "Население". Создайте экземпляр класса и вызовите методы передав полям значения.

12. Создайте класс "Страна". Добавьте в класс "Страна" метод, который выводит информацию о стране на экран.

13. Создайте класс "Товар" с полями "Название", "Цена" и "Количество". Создайте экземпляр класса и вызовите методы передав полям значения.

14. Создайте класс "Товар". Добавьте в класс "Товар" метод, который выводит информацию о товаре на экран.

15. Создайте класс "Банк" с полями "Название", "Адрес" и "Количество счетов". Создайте экземпляр класса и вызовите методы передав полям значения.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Язык C# позволяет разработчику определить новый тип данных – класс. Простейшее описание класса имеет следующий вид.

```
class имя класса {< список членов класса>}
```

имя класса – это любой идентификатор, уникальный в пределах своей сферы действия;

список членов класса – список, объявляющий члены класса с указанием уровня доступа.

Члены класса – это поля (члены-данные), методы (функции-члены), свойства и события. В данной лабораторной работе будут создаваться классы, имеющие в качестве членов поля и методы.

Например:

```
class MyClass {  
  
    public int i;  
  
    public float x;  
  
    public float f()  
  
    {return i*y;}  
  
}
```

В C# имеется пять спецификаторов доступа: public, private, protected, internal, protected internal . Значения спецификаторов доступа следующие:

- public - член класса может быть использован любым методом;
- private - член класса может быть использован только методом того же класса,
- protected - член класса может быть использован только либо методом того же класса, либо методом производного класса;

- internal- член класса может быть использован только методом из той же сборки;

- protected internal- член класса может быть использован только либо методом из той же сборки или методом производного класса из любой сборки.

Во всех объектно-ориентированных языках четко различаются понятия «класс» и «объект». Класс – это определяемый пользователем тип данных, его можно представить себе как чертеж, по которому будут строиться сами данные, т.е. объекты (экземпляры) класса.

В C# создание объектов класса происходит с помощью операции new.

Например:

```
Myclass a1=new Myclass();
```

или

```
Myclass a2;
```

```
a2=new Myclass();
```

Ключевое слово new означает, что среде выполнения следует вызвать конструктор класса и выделить необходимое количество оперативной памяти под экземпляр класса. Поскольку класс относится к ссылочному типу, то выделении памяти производится из «кучи», находящейся в распоряжении среды выполнения .NET . Команды Myclass a1, Myclass a2, определяют ссылки на объекты класса, а не сами объекты, объекты создаются с помощью операции new . Но после того как объекты будут созданы, мы будем их называть объект (экземпляр) a1, и объект (экземпляр) a2.

Рассмотрим еще один пример:

```
Myclass c=new Myclass();
```

```
Myclass d;
```

```
d=c;
```

В этом примере создан один объект и две ссылки на него.

Доступ к членам класса осуществляется с помощью операции точка «.».

Например:

```
c.x=7; //полю x объекта c присвоено значение 7.
```

```
d.i=6; //полю i объекта d присвоено значение 6.
```

a1.x=c.f();//полю x объекта a1 присвоено значение, которое возвращает метод f, вызванный для объекта c, поскольку ссылка d также указывает на этот же объект, то метод f вернет значение 42.

Конструктор класса C#

Конструктор класса – это метод класса, который предназначен для создания объектов класса и инициализации их.

Конструктор обладает следующими свойствами:

1. Имя конструктора совпадает с именем класса.
2. Конструктор не имеет возвращаемого значения, даже типа void
3. Класс может содержать несколько конструкторов, отличающихся либо количеством, либо типами параметров (отличие в этом случае должно быть существенным), либо и тем и другим.
4. Конструктор без параметров называется конструктором по умолчанию.
5. Если ни один конструктор не был написан программистом, то конструктор по умолчанию генерируется компилятором.
6. Если в классе объявлен хотя бы один конструктор, то компилятор не создает своего.
7. В C# , в отличие от многих других языков, любой конструктор, если не указано иное, обнуляет значения полей.
8. Конструктор не может быть вызван явно, он вызывается автоматически при создании экземпляров класса.

Рассмотрим пример.

Определим в классе A два конструктора.

```
class A {  
  
    public int x,y;  
  
    public int f()  
  
    {return x+y;}  
  
    public A() { x=1; y=2; }  
  
    public A (int xx, int yy) { x=xx; y=yy; }  
  
}
```

При выполнении команды `A a=new A();` для создания объекта класса используется конструктор без параметров. Полям объекта `a` будут присвоены, соответственно, значения 1 и 2.

Если при создании объекта требуется использовать конструктор с параметрами, то в операторе `new` после имени класса в круглых скобках указываются значения параметров, например, `A b=new A(6,13);`

А команда `A c=new A(3);` вызовет ошибку, потому что класс `A` не содержит конструктора с одним параметром, которому можно было бы присвоить значение 3.

В языке `C#` существует упрощенная форма написания конструктора: при описании полей класса можно написать инициализирующие выражения.

Пример.

```
class B {  
  
    public int i=10;  
  
}
```

Все строки, содержащие инициализирующие выражения, включаются в начало всех конструкторов. В данном примере строка `i=10;` будет включена в конструктор, определяемый компилятором.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. Что такое классы?
2. Как создать класс?
3. Какие основные элементы входят в класс?
4. Опишите значения спецификаторов доступа?
5. Какие методы и свойства могут быть определены в классе?
6. Какие конструкторы могут быть определены в классе?

8. Рекомендуемая литература

1. Рихтер, Дж. CLR via `C#`. Программирование на платформе Microsoft .NET Framework 4.5 на языке `C#` / Дж. Рихтер. СПб. : Изд-во Питер, 2021. 896 с.

- 2. Прайс, М. Дж.** С# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.
- 3. Васильев, А.Н.** Программирование на С# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.
- 4. Фримен, А.** ASP.NET Core 3 с примерами на С# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.