```csharp
namespace Delegado6
{
    public class Student
    {
        private string _name;
        private int _age;

        public string Name
        {
            get { return _name; }
            set { _name = value; }
        }
        public int Age
        {
            get { return _age; }
            set { _age = value; }
        }
        public Student()
        {

        }
        public Student (string name, int age)
        {
            _name = name;
            _age = age;
        }
    }
}
```

```csharp
namespace Delegado6
{
    public delegate bool DelegateFilter(Student student);
    public class Classroom
    {
        private List<Student> _studentsList = new List<Student>();

        public void CreateList()
        {
            Student s1 = new Student() { Age = 15, Name = "Peter" };
            _studentsList.Add(s1);

            Student s2 = new Student() { Age = 25, Name = "Sarah" };
            _studentsList.Add(s2);

            Student s3 = new Student() { Age = 42, Name = "Leroy" };
            _studentsList.Add(s3);

            Student s4 = new Student() { Age = 2, Name = "Nina" };
            _studentsList.Add(s4);

            Student s5 = new Student() { Age = 65, Name = "Large" };
            _studentsList.Add(s5);
        }

        public List<string> Filter(DelegateFilter del)
        {
            List<string> listResultNames = new List<string>();
            for(int i = 0; i < _studentsList.Count; i++)
            {
                if (del(_studentsList[i]))
                {
                    listResultNames.Add(_studentsList[i].Name);
                }
            }
            return listResultNames;

        }

        public void DisplayNames()
        {
            foreach(Student s in _studentsList)
            {
                Console.WriteLine(s.Name);
            }
        }
    }
}
```

```csharp
namespace Delegado6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Classroom c1 = new Classroom();
            c1.CreateList();

            c1.DisplayNames();

            List<string> list = new List<string>();
            list = c1.Filter((student) =>
            {
                if (student.Name == "Sarah")
                    return true;
                return false;
            });

            foreach(string s in list)
            {
                Console.WriteLine(s);
            }
        }
    }
}
```