

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

Учебный предмет
«Конструирование программ и языки программирования»

Инструкция
по выполнению лабораторной работы №10
«Разработка, отладка и испытание программ, реализующих механизм
наследования»

Минск 2024 г.

Лабораторная работа № 10

Тема работы: «Разработка, отладка и испытание программ, реализующих механизм наследования»

1 Цель работы

Сформировать умения разрабатывать алгоритмы и реализовать программы, реализующие механизм наследования классов.

2 Задание

Номер варианта соответствует номеру по списку в журнале.

Реализовать классы по заданию. В программе реализовать механизм наследования классов.

1. Реализуйте класс Student (Студент), который будет наследовать от класса User, подобно тому, как это сделано в теоретической части урока. Этот класс должен иметь следующие свойства: name (имя, наследуется от User), surname (фамилия, наследуется от User), year (год поступления в вуз). Класс должен иметь метод getFullName() (наследуется от User), с помощью которого можно вывести одновременно имя и фамилию студента. Также класс должен иметь метод getCourse(), который будет выводить текущий курс студента (от 1 до 5). Курс вычисляется так: нужно от текущего года отнять год поступления в вуз. Текущий год получите самостоятельно.

2. Разработать программу с использованием наследования классов, реализующую классы: железнодорожный вагон; вагон для перевозки автомобилей; цистерна. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран его вес и количество единиц товара в вагоне.

3. Разработать программу с использованием наследования классов, реализующую классы: массив; стек; очередь. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран количество элементов и добавьте элемент.

4. Разработать программу с использованием наследования классов, реализующую классы: точка; линия; круг. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран координаты и размер.

5. Разработать программу с использованием наследования классов, реализующую классы: работник больницы; медсестра; хирург. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран возраст и название должности.

6. Разработать программу с использованием наследования классов, реализующую классы: точка; квадрат пирамида. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран его размер и координаты.

7. Разработать программу с использованием наследования классов, реализующую классы: реагент; углерод; железо. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран

его количество и свойства (форма кристаллической решетки для углерода и чистота выработки руды для железа).

8. Разработать программу с использованием наследования классов, реализующую классы: работник фирмы; стажер; руководящий сотрудник; директор. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран целое число - уровень допуска, и название должности.

9. Разработать программу с использованием наследования классов, реализующую классы: молодой человек; студент; военнослужащий; военный курсант. Используя виртуальное наследование и виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран сведения о военнообязанности.

10. Разработать программу с использованием наследования классов, реализующую классы: воин; пехотинец(винтовка); матрос(кортик). Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран его возраст и вид оружия.

11. Разработать программу с использованием наследования классов, реализующую классы: графический объект; круг; квадрат. Используя виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран его размер и координаты.

12. Разработать три класса, которые следует связать между собой, используя наследование: класс Product, который имеет три элемент-данных — имя, цена и вес товара (базовый класс для всех классов); класс Buy, содержащий данные о количестве покупаемого товара в штуках, о цене за весь купленный товар и о весе товара (производный класс для класса Product и базовый класс для класса Check); класс Check, не содержащий никаких элемент-данных. Данный класс должен выводить на экран информацию о товаре и о покупке (производный класс для класса Buy);

13. Разработать программу с использованием наследования от базового класса Pet и наследников Cat, Dog, Parrot. Используя виртуальное наследование и виртуальные функции, не зная с объектом какого класса вы работаете, выведите на экран сведения о животном.

14. Создайте базовый класс «Персональные компьютеры» и на его основе подклассы: «Настольные ПК» и «Ноутбуки». В базовом классе определите общие свойства: размер памяти, диска, модель, CPU. в производных классах уникальные свойства: для настольных: монитор, клавиатура, мышь, их габариты; и метод для вывода этой информации в консоль; для ноутбуков: габариты, диагональ экрана; и метод для вывода этой информации в консоль.

15. Создайте базовый класс «Человек» и на его основе подклассы: «Мужчина» и «Женщина». Подумайте, какие общие характеристики можно выделить в суперкласс и какие частные свойства указать в подклассах.

3 Оснащение работы

ПК, среда Visual Studio 2019, MSword.

4 Основные теоретические сведения

Наследование является одним из фундаментальных атрибутов объектно-ориентированного программирования. Оно позволяет определить дочерний класс, который использует (наследует), расширяет или изменяет возможности родительского класса. Класс, члены которого наследуются, называется базовым классом. Класс, который наследует члены базового класса, называется производным классом.

C# и .NET поддерживают только одиночное наследование. Это означает, что каждый класс может наследовать члены только одного класса. Но зато поддерживается транзитивное наследование, которое позволяет определить иерархию наследования для набора типов. Другими словами, тип D может наследовать возможности типа C, который в свою очередь наследует от типа B, который наследует от базового класса A. Благодаря транзитивности наследования члены типа A будут доступны для типа D.

Не все члены базового класса наследуются производными классами. Следующие члены не наследуются.

- Статические конструкторы, которые инициализируют статические данные класса.

- Конструкторы экземпляров, которые вызываются для создания нового экземпляра класса. Каждый класс должен определять собственные конструкторы.

- Методы завершения, которые вызываются сборщиком мусора среды выполнения для уничтожения экземпляров класса.

По умолчанию все классы наследуются от базового класса Object, даже если мы явным образом не устанавливаем наследование.

Все классы по умолчанию могут наследоваться.

Однако существует ряд ограничений:

- Не поддерживается множественное наследование, класс может наследоваться только от одного класса.

- При создании производного класса надо учитывать тип доступа к базовому классу - тип доступа к производному классу должен быть таким же, как и у базового класса, или более строгим. То есть, если базовый класс у нас имеет тип доступа internal, то производный класс может иметь тип доступа internal или private, но не public.

- Однако следует также учитывать, что если базовый и производный класс находятся в разных сборках (проектах), то в этом случае производный класс может наследовать только от класса, который имеет модификатор public.

- Если класс объявлен с модификатором sealed, то от этого класса нельзя наследовать и создавать производные классы.

- Нельзя унаследовать класс от статического класса.

Производный класс может иметь доступ только к тем членам базового класса, которые определены с модификаторами `private` `protected` (если базовый и производный класс находятся в одной сборке), `public`, `internal` (если базовый и производный класс находятся в одной сборке), `protected` и `protected internal`.

С помощью ключевого слова `base` можно обратиться к базовому классу.

Конструкторы не передаются производному классу при наследовании. И, если в базовом классе не определен конструктор по умолчанию без параметров, а только конструкторы с параметрами, то в производном классе необходимо вызвать один из этих конструкторов через ключевое слово `base`.

При вызове конструктора класса сначала отрабатывают конструкторы базовых классов и только затем конструкторы производных.

Функция-элемент может быть объявлена как `virtual`. Ключевое слово `virtual` предписывает компилятору генерировать некоторую дополнительную информацию о функции. Если функция переопределяется в производном классе и вызывается с указателем (или ссылкой) базового класса, ссылающимся на представитель производного класса, эта информация позволяет определить, какой из вариантов функции должен быть выбран: такой вызов будет адресован функции производного класса.

Для виртуальных функций существуют следующие правила: виртуальную функцию нельзя объявлять как `static`. Спецификатор `virtual` необязателен при переопределении функции в производном классе. Виртуальная функция должна быть определена в базовом классе и может быть переопределена в производном.

5. Порядок выполнения работы

1. Выделить ключевые моменты задачи.
2. Построить алгоритм и теоретическую объектную модель решения задачи.
3. Запрограммировать полученные алгоритмы и объектную модель.

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. Наследование – это...?
2. Какие члены базового класса не могут наследоваться производными?
3. Как будет проходить процесс наследования от класса, объявленного с модификатором `Sealed`?

4. Для чего необходимо служебное слово base?

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. C# 10 и .NET 6. Современная кросс-платформенная разработка / М. Дж. Прайс. СПб : Изд-во Питер, 2023. 848 с.

3. Васильев, А.Н. Программирование на C# для начинающих. Особенности языка / А.Н. Васильев. М. : Эксмо, 2022. 528 с.

4. Фримен, А. ASP.NET Core 3 с примерами на C# для профессионалов / А. Фримен. СПб. : Изд-во Вильямс, 2021. 1184 с.