

Практическое занятие № 7

Тема практического занятия: Механизм наследования, его реализация.

Цель практического занятия: получить понимание механизма наследования в объектно-ориентированном программировании и научиться применять его для создания гибкого и масштабируемого кода.

В результате выполнения данной работы обучающийся должен уметь:

1. Создавать базовый класс с необходимыми свойствами и методами
2. Создавать производные классы, которые будут наследовать свойства и методы базового класса
3. Использовать доступ к элементам базового класса
4. Применять механизм наследования для создания иерархий классов
5. Применять наследование для создания гибкого, модульного и поддерживаемого кода.

знать:

1. Понятие наследования в объектно-ориентированном программировании и его цель
2. Различия между базовым (родительским) классом и производными (дочерними) классами
3. Как унаследованные свойства и методы могут быть использованы в дочерних классах

Перечень оборудования, необходимого для выполнения задания:

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
- Монитор 34", изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).

Программное обеспечение:

- Microsoft Visual Studio,

Общие теоретические сведения:

Наследование - это механизм в объектно-ориентированном программировании, который позволяет создавать новые классы, называемые производными классами или дочерними классами, на основе уже существующих классов, называемых базовыми классами или родительскими

классами. Производный класс наследует свойства и методы базового класса, и может добавлять или изменять свою собственную функциональность.

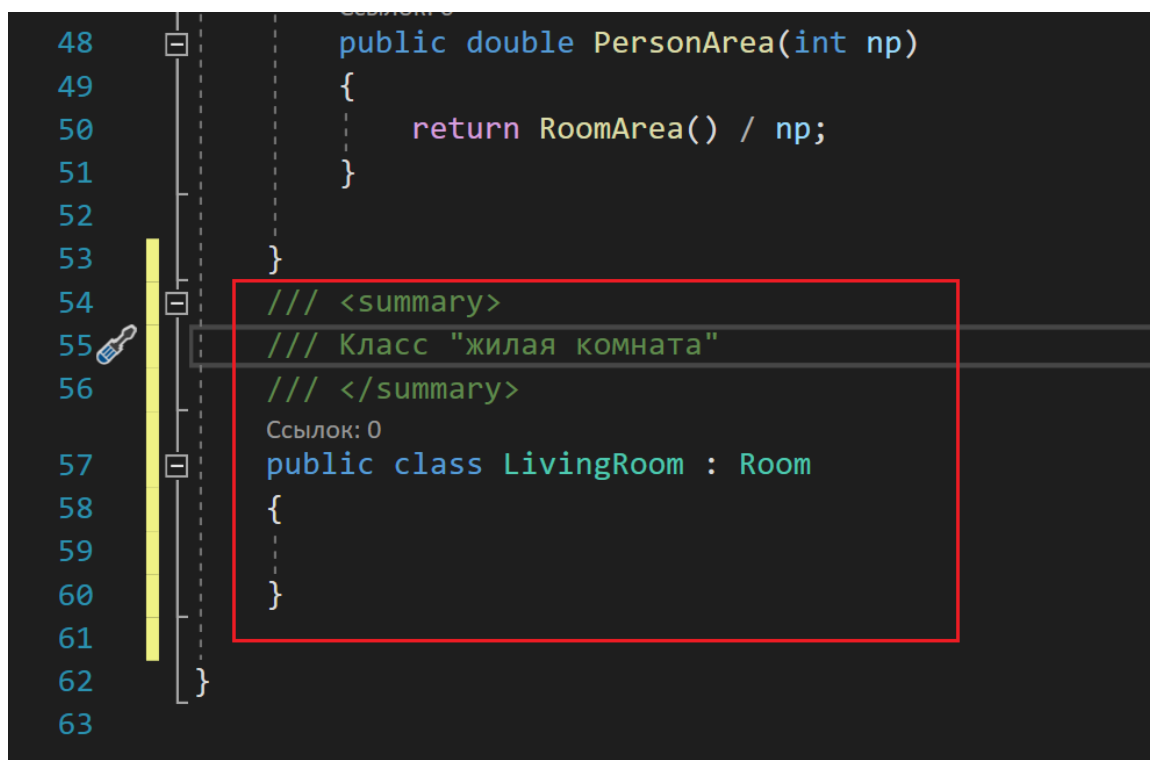
Базовый класс - это класс, от которого производные классы получают наследуемые свойства и методы. Он определяет общие черты и функциональность для всех связанных дочерних классов.

Производный класс - это класс, который наследует свойства и методы от базового класса и добавляет или изменяет их для предоставления своей собственной специфической функциональности. Производные классы могут также называться дочерними классами, наследниками или внутренними классами в зависимости от контекста.

Наследование позволяет использовать и переиспользовать код, сокращает дублирование и упрощает разработку и поддержку программного обеспечения. Оно также обеспечивает иерархичность и группировку классов с общими характеристиками, что помогает в логическом организации кода и его понимании.

Задание:

1. Доработайте библиотеку классов, разработанную во время практического занятия №5, добавив к базовому классу два производных: «Столица» и «Портовый город». Пример создания производного класса:



```
48         public double PersonArea(int np)
49         {
50             return RoomArea() / np;
51         }
52     }
53 }
54
55 /// <summary>
56 /// Класс "жилая комната"
57 /// </summary>
58 Ссылка: 0
59 public class LivingRoom : Room
60 {
61 }
62 }
63 }
```

2. В класс «Столица» должны быть добавлены поля: адрес правительства, количество приезжих (считается, что каждый день в столице оказывается одинаковое число приезжих). Каждому полю должно соответствовать свойство.

3. В класс «Столица» должен быть добавлен метод, рассчитывающий сколько м^2 приходится на одного человека с учетом приезжих.
4. В класс «Портовый город» должно быть добавлено поле площадь порта. Полно должно соответствовать свойство.
5. В класс «Портовый город» должен быть добавлен метод, рассчитывающий площадь самого города без учета площади порта.
6. Производные классы должны содержать конструкторы.
7. Разработайте приложение, которое позволит вводить данные о городах трех типов и выдавать о них информацию: название и число человек на 1км^2 .

Указания по технике безопасности:

Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

Технология выполнения работы (этапы, последовательность действий):

1. Доработайте библиотеку классов в соответствии с заданием. При расчете числа м^2 на одного человека не забудьте, что площадь города задается в км^2 .
2. Пересоберите библиотеку классов, убедитесь в успешности сборки.
3. Создайте новый проект для приложения.
4. Разработайте интерфейс приложения, в котором будут предусмотрены поля для ввода данных городов каждого типа. Вы можете сделать три области на одном окне (на каждый тип города по области), можете реализовать работу с каждым типом города в отдельном окне.
5. Реализуйте приложение в соответствии с заданием.
6. Проверьте правильность работы приложения.

Требование к отчету:

1. В библиотеке классов создано два производных класса: класс «Столица» и класс «Портовый город».
2. В классе «Столица» присутствуют поля для адреса правительства и числа приезжих, созданы соответствующие свойства.
3. В класс «Столица» добавлен конструктор и метод, позволяющий высчитать число м^2 на одного человека, расчет выполняется корректно, с учетом приезжих.
4. В классе «Портовый город» добавлено поле площадь порта и соответствующее ему свойство.
5. В класс «Портовый город» добавлен конструктор и метод, позволяющий определить площадь города без учета площади порта, вычисления производятся корректно.
6. Разработано приложение, к нему подключена библиотека.
7. В приложении реализована работа с тремя типами городов.
8. У пользователя есть возможность ввести все необходимые данные.
9. Пользователю выдается информация о городе: название и число человек на 1км^2 .

Контрольные вопросы:

1. Что такое наследование?
2. Что такое базовый класс?
3. Что такое производный класс?
4. Как обозначить при объявлении классов, что данный класс является производным от другого?
5. У нас есть базовый класс и производный. Будут ли доступны в производном классе поля, если они будут в базовом объявлены как приватные?
6. Какой модификатор доступа должен стоять у полей базового класса, если мы хотим, чтобы они были доступны в производном, но недоступны для других модулей?
7. Есть базовый класс и производный. Будут ли доступны в производном классе методы, объявленные в базовом классе? Если нет, то почему. Если да, то при каких условиях.
8. Есть базовый класс, есть его производный. Можем ли мы объявить класс, для которого этот производный будет базовым?
9. Вы что-то поменяли в библиотеке классов. Что надо сделать, чтобы в приложении использовалась уже новая версия этой библиотеки?
10. Какие признаки наследования вы можете выделить в своей библиотеке класса?

Основные и дополнительные источники, электронные ресурсы:

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1913989>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1927269>.