

Практическое занятие № 5

Тема практического занятия: Описание подпрограмм, их вызов.

Цель практического занятия: овладение навыками описания и вызова подпрограмм.

В результате выполнения данной работы обучающийся должен уметь:

1. Описывать подпрограммы, включая их имя, параметры и возвращаемые значения, используя правильный синтаксис и структуру.
2. Вызывать подпрограммы из основной программы или из других подпрограмм для выполнения определенных задач.
3. Обращаться к методам, описанным внутри класса.

знать:

1. Что такое подпрограммы (также известные как функции или методы) и как они используются для организации и структурирования кода в программировании.
2. Принципы передачи аргументов при вызове подпрограммы, включая передачу по значению и передачу по ссылке.

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 :

В программировании, подпрограммы (также называемые функциями или методами) - это фрагменты кода, которые выполняют определенную задачу и могут быть вызваны из других частей программы для выполнения этой задачи. Подпрограммы позволяют организовать и структурировать код, а также повторно использовать определенные операции или блоки кода.

Методы являются также элементами класса.

Основные характеристики подпрограмм:

1. Имя: подпрограммы обладают уникальным идентификатором, который используется для их вызова в других частях программы.

2. Параметры: подпрограммы могут принимать параметры (аргументы), которые являются входными данными для выполнения операций внутри подпрограммы. Параметры могут быть переданы по значению или по ссылке.

3. Возвращаемое значение: подпрограммы могут возвращать результат своей работы в виде значения, которое может быть использовано в других частях программы.

4. Вызов и возврат: подпрограммы вызываются из других частей программы, их код выполняется, а затем управление возвращается в место вызова, после чего продолжается выполнение кода программы.

Подпрограммы обеспечивают модульность и повторное использование кода, что упрощает разработку и поддержку программных систем. Они позволяют разбить сложные задачи на более мелкие блоки кода, что улучшает структуру и читаемость программы.

Задание:

1. Доработайте библиотеку классов, разработанную ранее во время практического занятия №4, добавив класс методы:
 - подсчитать число человек на 1 км²,
 - подсчитать число дней на посещение города (число достопримечательностей, которое может посетить путешественник за день, задается как параметр).

Пример описания методов для другого класса:

```
/// <summary>
///
/// </summary>
/// <returns></returns>
Ссылка: 0
public double RoomPerimeter()
{
    return 2 * (roomLength + roomWidth);
}
```

Для класса «Город» выполните задание по аналогии.

2. Добавьте в класс конструктор, который позволит инициализировать объект, передав туда необходимые значения как параметры.
3. Доработайте приложение, разработанное ранее во время практического занятия №4, добавив возможность вводить все необходимые данные (данные о городе и получать о городе и число достопримечательностей, которое может посетить путешественник за день).
4. Замените в приложении конструктор по умолчанию на собственный конструктор.

5. Измените вывод информации: выводиться должно название города, число человек на 1км², число дней на посещение города.

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.
2. Пересоберите библиотеку.
3. Доработайте интерфейс приложения в соответствии с заданием.
4. Замените конструктор в приложении.
5. Измените вывод информации в соответствии с заданием.
6. Убедитесь в работоспособности приложения.

Требование к отчету:

1. В библиотеку добавлены все требуемые методы, включая конструктор, библиотека пересобрана без ошибок.
2. При описании методов корректно выбраны типы данных для параметров и для возвращаемых методами значений.
3. Приложение доработано в соответствии с заданием.
4. Использован собственный конструктор вместо конструктора по умолчанию.
5. Приложение компилируется, работоспособно, все данные выводятся корректно.

Контрольные вопросы:

1. Что такое методы?
2. Где указывается тип возвращаемого методом значения? Что писать, если метод не возвращает значения?
3. Где перечисляются параметры метода? Что делать, если параметров нет?
4. С помощью чего возвращается значение из метода?
5. Может ли один метод класса обращаться к другому?
6. Обязательно ли имя передаваемого из приложения параметра должно совпадать с именем параметра, описанного в заголовке метода?
7. Что означает слово void в заголовке метода?

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>.