|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемой работы | Подпись  руководителя |
| 26.03.2021 | **Практическая работа №5**  Тема: Обработка исключительных ситуаций.  Наименование работы: Обработка исключительных ситуаций. Перехват исключений. Системные исключения и исключения приложения.  Цель: Сформировать навыки создания программ с перехватом исключений и их обработкой.  Задание 1.  Изучил предложенную теорию. Кратко законспектировал.  Обработка исключений.  В программах непредвиденные ситуации могут приводить к ошибкам. Например, когда необходимо ввести число, а введена строка, то при конвертации этой строки в численный тип программа бы аварийно завершила работу.  Непредвиденные ситуации в С# называются исключениями. Обработка исключений – это описание реакции программы на подобные события во время выполнения программы. Реакцией программы может быть корректное завершение работы программы, вывод информации об ошибке и запрос повторения действия.  Типы исключений.  Ниже приведены некоторые из часто встречаемых типов исключений.  *Exception* – базовый тип всех исключений. Блок catch, в котором указан тип Exception будет «ловить» все исключения. *FormatException* – некорректный формат операнда или аргумента.  *NullReferenceException* - В экземпляре объекта не задана ссылка на объект, объект не создан.  *IndexOutOfRangeException* – индекс вне рамок коллекции.  *FileNotFoundException* – файл не найден. *DivideByZeroException* – деление на ноль.  Несколько блоков catch.  Одному блоку try может соответствовать несколько блоков catch:  try  {  //блок1  }  catch (FormatException)  {  //блок-обработка исключения 1  }  catch (FileNotFoundException)  {  //блок-обработка исключения 2  }  Блок finally.  Оператор try-catch также может содержать блок finally. Особенность блока finally в том, что код внутри этого блока выполнится в любом случае, в независимости от того, было ли исключение или нет.  try  {  //блок1  }  catch (Exception)  {  //обработка исключения  }  finally  {  //блок кода, который выполнится обязательно  }  Задание 2.  Создал программы по образцу.  Задание 2.1  Создать приложение с проверкой на корректность данных (исключение приложения). Листинг кода задания 2.1 представлен в Приложении 4. Скриншот выполнения программы представлен на Рисунке 9    Рисунок 9 – результат выполнения программы  Задание 2.2  Создать приложение с проверкой на корректность данных (системные исключения). Листинг кода задания 2.1 представлен в Приложении 4. Скриншот выполнения программы представлен на Рисунке 10  7.png  Рисунок 10 – результат выполнения программы  **Контрольные вопросы:**   1. Опишите обработку исключительных ситуаций в C#.   Обработка исключений – это описание реакции программы на подобные события во время выполнения программы.   1. Поясните назначение блока try.   Блок кода, в котором происходит исключение.   1. Поясните назначение блока catch.   Блок кода, в котором обрабатывается исключение.   1. Поясните назначение блока finally.   Блок кода, который выполняется в любом случае.   1. Перечислите типы исключений.   Exception, FormatException, NullReferenceException, IndexOutOfRangeException, FileNotFoundException, DivideByZeroException. |  |