**Государственное образовательное учреждение**

**среднего профессионально образования «Тираспольский техникум информатики и права»**

**ОТЧЁТ**

**Практическое задание №7**

по дисциплине «Разработка программных модулей»

**тема: «Обработка исключений»**

Работу выполнил

студент группы 215

Иванченко Даниил Богданович

Руководитель,

преподаватель

Подсекина Т.С.

Тирасполь, 2025

**Вариант 11**

**Задание 1**. Разработать программу, выбрасывающую исключение и использующую:

не менее 3-х обработчиков: Сотрудник офиса работает с поступившими документами, оставляя у себя, их копии. Если на столе хранится более и документов, то он перекладывает их в ящики стола, и складывает новые документы, если в ящиках стола хранится более m документов, то сдает их в архив. Если сотрудника увольняют, то он удаляет все документы из своего стола.

В листинге 1 приведен код программы.

|  |
| --- |
| using System;  using System.Collections.Generic;  class Document  {  public string Title { get; set; }  public Document(string title)  {  Title = title;  }  public override string ToString() => Title;  }  class TooManyDocumentsOnDeskException : Exception  {  public TooManyDocumentsOnDeskException(string message) : base(message) { }  }  class TooManyDocumentsInDrawersException : Exception  {  public TooManyDocumentsInDrawersException(string message) : base(message) { }  }  class EmployeeFiredException : Exception  {  public EmployeeFiredException(string message) : base(message) { }  }  class Archive  {  private List<Document> archivedDocs = new List<Document>();  public void StoreDocuments(List<Document> docs)  {  archivedDocs.AddRange(docs);  Console.WriteLine("Документы сданы в архив.");  }  }  class Desk  {  public List<Document> OnDesk = new List<Document>();  public List<Document> InDrawers = new List<Document>();  public int DeskLimit { get; }  public int DrawerLimit { get; }  public Desk(int deskLimit, int drawerLimit)  {  DeskLimit = deskLimit;  DrawerLimit = drawerLimit;  }  public void AddDocument(Document doc)  {  if (OnDesk.Count >= DeskLimit)  throw new TooManyDocumentsOnDeskException("На столе слишком много документов!");  OnDesk.Add(doc);  Console.WriteLine($"Документ \"{doc.Title}\" добавлен на стол.");  }  public void MoveDeskToDrawers()  {  if (InDrawers.Count + OnDesk.Count > DrawerLimit)  throw new TooManyDocumentsInDrawersException("В ящиках стола слишком много документов!");  InDrawers.AddRange(OnDesk);  OnDesk.Clear();  Console.WriteLine("Документы со стола перемещены в ящики стола.");  }  public List<Document> ClearAll()  {  var allDocs = new List<Document>();  allDocs.AddRange(OnDesk);  allDocs.AddRange(InDrawers);  OnDesk.Clear();  InDrawers.Clear();  return allDocs;  }  }  class OfficeWorker  {  public string Name { get; }  private Desk desk;  private Archive archive;  private bool isFired = false;  public OfficeWorker(string name, int deskLimit, int drawerLimit, Archive archive)  {  Name = name;  desk = new Desk(deskLimit, drawerLimit);  this.archive = archive;  }  public void WorkWithDocument(Document doc)  {  if (isFired)  throw new EmployeeFiredException($"Сотрудник {Name} уволен и не может работать с документами.");  try  {  desk.AddDocument(doc);  }  catch (TooManyDocumentsOnDeskException ex)  {  Console.WriteLine(ex.Message);  try  {  desk.MoveDeskToDrawers();  desk.AddDocument(doc);  }  catch (TooManyDocumentsInDrawersException dex)  {  Console.WriteLine(dex.Message);  archive.StoreDocuments(desk.InDrawers);  desk.InDrawers.Clear();  desk.AddDocument(doc);  }  }  }  public void Fire()  {  isFired = true;  var docsToRemove = desk.ClearAll();  Console.WriteLine($"Сотрудник {Name} уволен. Все документы удалены со стола ({docsToRemove.Count} документов).");  }  }  class Program  {  static void Main()  {  Archive archive = new Archive();  OfficeWorker worker = new OfficeWorker("Иванов", deskLimit: 3, drawerLimit: 5, archive);  var docs = new List<Document>  {  new Document("Документ 1"),  new Document("Документ 2"),  new Document("Документ 3"),  new Document("Документ 4"),  new Document("Документ 5"),  new Document("Документ 6"),  new Document("Документ 7"),  };  foreach (var doc in docs)  {  try  {  worker.WorkWithDocument(doc);  }  catch (Exception ex)  {  Console.WriteLine("Ошибка: " + ex.Message);  }  }    try  {  worker.Fire();  worker.WorkWithDocument(new Document("Документ после увольнения"));  }  catch (EmployeeFiredException ex)  {  Console.WriteLine("Обработка увольнения: " + ex.Message);  }  Console.Read();  }  } |

Листинг 1 – программа определения алгоритмов

Ниже на рисунках 1 представлены тесты программы при разных входных параметрах.

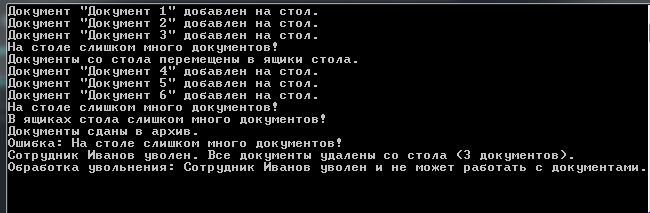


Рисунок 1 – тест 1