ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

навчальна дисципліна «Об'єктно-орієнтоване програмування»

ЗВІТ

про виконання лабораторної роботи № 3

на тему: «Розроблення застосунків з використанням базових елементів ООП »

Варіант № 11

Виконав:

студент 2 курсу

групи 6.04.121.010.23.2

інститу ІТ

Місюра О.А

Перевірив:

доцент кафедри ІС

Бредіхін В.М.

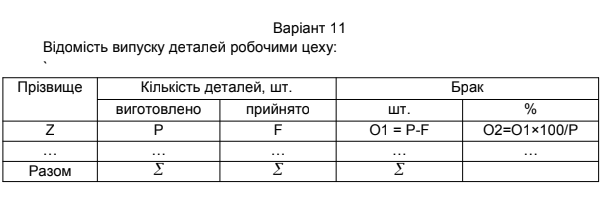
м. Харків – 2024 рік

**Цілі лабораторної роботи:**

1. Придбання практичних навичок використання основних елементів ООП та відносин агрегації і композиції при розробленні програм мовами Java і C#.

2. Удосконалення навичок роботи з певними інтегрованими середовищами програмування.

Завдання 1.



Умова завдання:

Потрібно створити застосунок на Java або C#, який автоматизує облік відвідуваності занять студентами. Необхідно:

1. Створити два класи:
   * Клас для опису запису відомості (прізвище, пропущені години, виправдані години, розрахунок невиправданих годин і відсотка).
   * Головний клас програми з методом main() для введення, обробки та виведення даних.
2. Реалізувати:
   * Введення даних через консоль.
   * Збереження даних у масив об'єктів.
   * Обчислення та форматований вивід даних у вигляді таблиці.
   * Текстове меню для вибору дій.

Код:

using System;

namespace WorkshopDetailsApp

{

class PhoneCall

{

public string WorkerName { get; private set; }

public int Produced { get; private set; }

public int Accepted { get; private set; }

public PhoneCall(string workerName, int produced, int accepted)

{

WorkerName = workerName;

Produced = produced;

Accepted = accepted;

}

public int GetDefectiveCount()

{

return Produced - Accepted;

}

public double GetDefectivePercentage()

{

return Produced > 0 ? (double)GetDefectiveCount() \* 100 / Produced : 0;

}

public void Display()

{

Console.WriteLine($"Прізвище: {WorkerName}, Виготовлено: {Produced}, Прийнято: {Accepted}, " +

$"Брак (шт.): {GetDefectiveCount()}, Брак (%): {GetDefectivePercentage():F2}");

}

}

class MainProgram

{

public static int GetTotalProduced(PhoneCall[] phoneCalls)

{

int total = 0;

foreach (var call in phoneCalls)

{

total += call.Produced;

}

return total;

}

public static int GetTotalAccepted(PhoneCall[] phoneCalls)

{

int total = 0;

foreach (var call in phoneCalls)

{

total += call.Accepted;

}

return total;

}

public static void DisplayAll(PhoneCall[] phoneCalls)

{

Console.WriteLine("Відомість:");

foreach (var call in phoneCalls)

{

call.Display();

}

}

public static void Main(string[] args)

{

PhoneCall[] phoneCalls = new PhoneCall[]

{

new PhoneCall("Іваненко", 100, 90),

new PhoneCall("Петренко", 80, 75),

new PhoneCall("Сидоренко", 120, 110)

};

DisplayAll(phoneCalls);

int totalProduced = GetTotalProduced(phoneCalls);

int totalAccepted = GetTotalAccepted(phoneCalls);

int totalDefective = totalProduced - totalAccepted;

double totalDefectivePercentage = totalProduced > 0 ? (double)totalDefective \* 100 / totalProduced : 0;

Console.WriteLine("\nПідсумки:");

Console.WriteLine($"Загальна кількість виготовлених деталей: {totalProduced}");

Console.WriteLine($"Загальна кількість прийнятих деталей: {totalAccepted}");

Console.WriteLine($"Загальний брак (шт.): {totalDefective}");

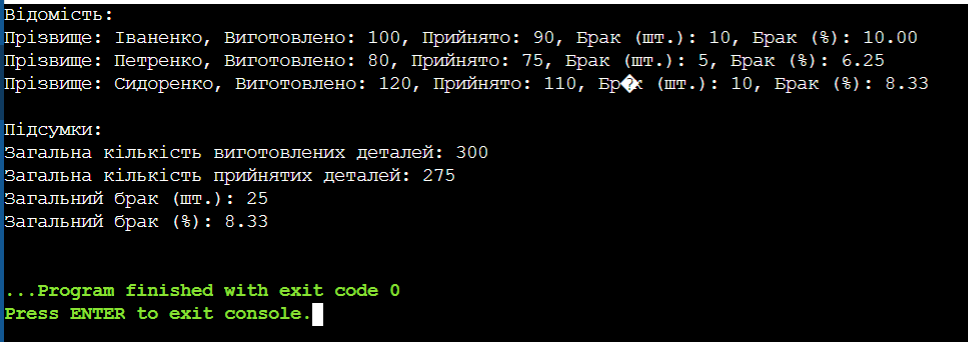
Console.WriteLine($"Загальний брак (%): {totalDefectivePercentage:F2}");

}

}

}

Результат виконання:



Завдання 2.

Перетворити застосунок для обробки відомості так, щоб в ньому використовувалося відношення агрегації або композиції, а його функціональні можливості залишилися тими ж. Він має містити три класи:

1. Клас - «ціле», що описує відомість з безліччю записів. 2. Клас - «частина», який описує будь-який запис відомості. 3. Клас, що містить головний метод програми, в якому повинен створюватися один об'єкт класу - «цілого» і з використанням посилання на цей об'єкт викликатися методи класу - «цілого».

Код:

using System;

namespace WorkshopDetailsApp

{

class PhoneCall

{

public string WorkerName { get; private set; }

public int Produced { get; private set; }

public int Accepted { get; private set; }

public PhoneCall(string workerName, int produced, int accepted)

{

WorkerName = workerName;

Produced = produced;

Accepted = accepted;

}

public int GetDefectiveCount()

{

return Produced - Accepted;

}

public double GetDefectivePercentage()

{

return Produced > 0 ? (double)GetDefectiveCount() \* 100 / Produced : 0;

}

public void Display()

{

Console.WriteLine($"Прізвище: {WorkerName}, Виготовлено: {Produced}, Прийнято: {Accepted}, " +

$"Брак (шт.): {GetDefectiveCount()}, Брак (%): {GetDefectivePercentage():F2}");

}

}

class AttendanceReport

{

private PhoneCall[] phoneCalls;

public AttendanceReport(PhoneCall[] phoneCalls)

{

this.phoneCalls = phoneCalls;

}

public int GetTotalProduced()

{

int total = 0;

foreach (var call in phoneCalls)

{

total += call.Produced;

}

return total;

}

public int GetTotalAccepted()

{

int total = 0;

foreach (var call in phoneCalls)

{

total += call.Accepted;

}

return total;

}

public void DisplayAll()

{

Console.WriteLine("Відомість:");

foreach (var call in phoneCalls)

{

call.Display();

}

}

public void DisplaySummary()

{

int totalProduced = GetTotalProduced();

int totalAccepted = GetTotalAccepted();

int totalDefective = totalProduced - totalAccepted;

double totalDefectivePercentage = totalProduced > 0 ? (double)totalDefective \* 100 / totalProduced : 0;

Console.WriteLine("\nПідсумки:");

Console.WriteLine($"Загальна кількість виготовлених деталей: {totalProduced}");

Console.WriteLine($"Загальна кількість прийнятих деталей: {totalAccepted}");

Console.WriteLine($"Загальний брак (шт.): {totalDefective}");

Console.WriteLine($"Загальний брак (%): {totalDefectivePercentage:F2}");

}

}

class MainProgram

{

public static void Main(string[] args)

{

PhoneCall[] phoneCalls = new PhoneCall[]

{

new PhoneCall("Іваненко", 100, 90),

new PhoneCall("Петренко", 80, 75),

new PhoneCall("Сидоренко", 120, 110)

};

AttendanceReport report = new AttendanceReport(phoneCalls);

report.DisplayAll();

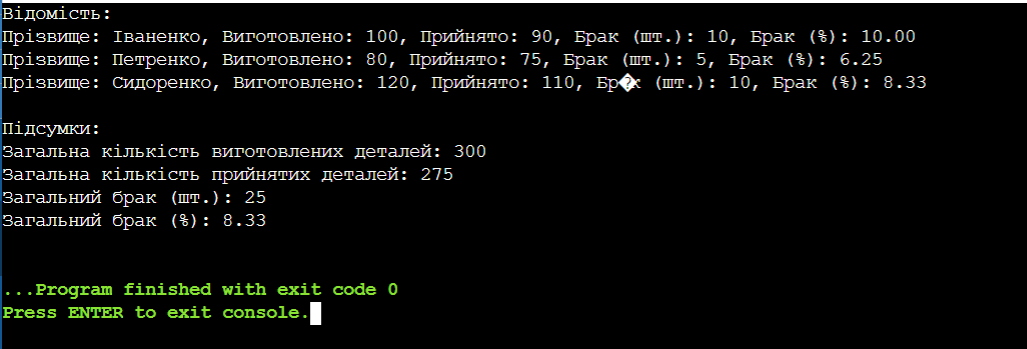
report.DisplaySummary();

}

}

}

Результат виконання:



**Висновок:** У ході виконання лабораторної роботи я навчився застосовувати принципи об'єктно-орієнтованого програмування для розв'язання практичних завдань. Я ознайомився з основними концепціями, такими як агрегація та композиція, і застосував їх на практиці в контексті розробки застосунку для обробки відомості про виготовлені деталі.

Виконуючи завдання, я створив класи для опису окремих записів відомості та для управління всією колекцією цих записів. Це дозволило розділити обов'язки між класами і зробити програму більш організованою та розширюваною. Я також навчився працювати з масивами об'єктів, обчислювати підсумкові значення та виводити відформатовані результати на екран.

Загалом, лабораторна робота дала мені можливість краще зрозуміти, як організувати структуру програм, використовуючи об'єктно-орієнтовані підходи, та надала практичний досвід у написанні програм на мові C#