Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт
з лабораторної роботи №4 з дисципліни
«Основи програмування - 2»

«Успадкування та поліморфізм»

Варіант 4

Виконав: Бондаренко М.В. ІП-13

Перевірила: Вечерковська А.С.

Київ 2022

Лабораторна робота 4

Тема: успадкування та поліморфізм.

Постановка задачі

4. Створити клас TTriad, який представляє трійку цілих чисел і містить методи для їх збільшення / зменшення на вказану величину. На основі цього класу створити класи-нащадки TTime (“години:хвилини:секунди”) та TDate (“число.місяць.рік”). Випадковим чином згенерувати n дат та m об’єктів часу. Визначити, які із дат мають значення, що є допустимими, якщо їх трактувати як час. Всі інші дати зменшити на 5 днів, а весь інший час збільшити на 15 хвилин.

Код C#:

Program.cs:

```
using System;
using System.Collections.Generic;

namespace lab4
{
    internal class Program
    {
        public static void Main(string[] args)
        {
            List<TDate> dates = Funcs.DatesGenerate();
            List<TTime> times = Funcs.TimesGenerate();

            Console.WriteLine("\nInitial dates:");
            Funcs.ListOutput(dates);
            Console.WriteLine("\nInitial times:");
            Funcs.ListOutput(times);

            List<TDate> dates_like_time = Funcs.DatesLikeTime(dates);
            Console.WriteLine("\nDates that can be interpreted like time:");
            Funcs.ListOutput(dates_like_time);

            Funcs.ChangeDates(dates);
            Funcs.ChangeTimes(times);

            Console.WriteLine("\nChanged dates:");
            Funcs.ListOutput(dates);
            Console.WriteLine("\nChanged times:");
            Funcs.ListOutput(times);
        }
    }
}
```

Funcs.cs:

```
using System;
using System.Collections.Generic;

namespace lab4
{
    public class Funcs
    {
        public static List<TDate> DatesGenerate()
        {
            List<TDate> dates = new List<TDate>();
            Random random = new Random();
            Console.Write("Enter the amount of dates: ");
```

```
            int n = Convert.ToInt32(Console.ReadLine());
            for (int i = 0; i < n; i++)
            {
                int day = random.Next(1, 31);
                int month = random.Next(1, 12);
                int year = random.Next(0, 100);
                TDate date = new TDate(day, month, year);
                dates.Add(date);
            }

            return dates;
        }

        public static List<TTime> TimesGenerate()
        {
            List<TTime> times = new List<TTime>();
            Random random = new Random();
            Console.Write("Enter the amount of times: ");
            int n = Convert.ToInt32(Console.ReadLine());
            for (int i = 0; i < n; i++)
            {
                int hours = random.Next(0, 24);
                int minutes = random.Next(0, 60);
                int seconds = random.Next(0, 60);
                TTime time = new TTime(hours, minutes, seconds);
                times.Add(time);
            }

            return times;
        }

        public static void ListOutput(List<TDate> list)
        {
            foreach (var date in list)
            {
                date.Output();
            }
        }

        public static void ListOutput(List<TTime> list)
        {
            foreach (var time in list)
            {
                time.Output();
            }
        }

        public static bool IsLikeTime(TDate date)
        {
            if (date.Number1 < 24 && date.Number3 < 60)
            {
                return true;
            }

            return false;
        }


        public static void ChangeDates(List<TDate> dates)
        {
            foreach (var date in dates)
            {
                if (!IsLikeTime(date))
                {
                    date.DecreaseNumber1(5);
```

```
                }
            }
        }

        public static void ChangeTimes(List<TTime> times)
        {
            foreach (var time in times)
            {
                time.IncreaseNumber2(15);
            }
        }

        public static string ChangeFormat(int n)
        {
            string n_str = Convert.ToString(n);
            if (n < 10)
            {
                n_str = "0" + n_str;
            }

            return n_str;
        }

        public static List<TDate> DatesLikeTime(List<TDate> dates)
        {
            List<TDate> dates_like_time = new List<TDate>();
            foreach (var date in dates)
            {
                if (IsLikeTime(date)) dates_like_time.Add(date);
            }

            return dates_like_time;
        }
    }
}
```

TTriad.cs:

```
namespace lab4
{
    public abstract class TTriad
    {
        public int Number1 { get; set; }
        public int Number2 { get; set; }
        public int Number3 { get; set; }

        protected TTriad(int n1, int n2, int n3)
        {
            Number1 = n1;
            Number2 = n2;
            Number3 = n3;
        }

        public abstract void IncreaseNumber1(int n);
        public abstract void DecreaseNumber1(int n);
        public abstract void IncreaseNumber2(int n);
        public abstract void DecreaseNumber2(int n);
        public abstract void IncreaseNumber3(int n);
        public abstract void DecreaseNumber3(int n);
    }
}
```

TTime.cs:

```
using System;

namespace lab4
{
```

```
public class TTime : TTriad
{
    public TTime(int n1, int n2, int n3) : base(n1, n2, n3)
    {
    }

    public override void IncreaseNumber1(int n)
    {
        Number1 += n;
        while (Number1 >= 24)
        {
            Number1 -= 24;
        }
    }

    public override void DecreaseNumber1(int n)
    {
        Number1 -= n;
        while (Number1 < 0)
        {
            Number1 += 24;
        }
    }

    public override void IncreaseNumber2(int n)
    {
        Number2 += n;
        while (Number2 >= 60)
        {
            Number2 -= 60;
            IncreaseNumber1(1);
        }
    }

    public override void DecreaseNumber2(int n)
    {
        Number2 -= n;
        while (Number2 < 0)
        {
            Number2 += 60;
            DecreaseNumber1(1);
        }
    }

    public override void IncreaseNumber3(int n)
    {
        Number3 += n;
        while (Number3 >= 60)
        {
            Number3 -= 60;
            IncreaseNumber2(1);
        }
    }

    public override void DecreaseNumber3(int n)
    {
        Number3 -= n;
        while (Number3 < 0)
        {
            Number3 += 60;
            DecreaseNumber2(1);
        }
    }

    public void Output()
```

```
        {

Console.WriteLine($"{Funcs.ChangeFormat(Number1)}:{Funcs.ChangeFormat(Number2)}:{Funcs.
ChangeFormat(Number3)}");
        }
    }
}
```

TDate.cs:

```
using System;

namespace lab4
{
    public class TDate : TTriad
    {
        public TDate(int n1, int n2, int n3) : base(n1, n2, n3)
        {
        }

        public override void IncreaseNumber1(int n)
        {
            Number1 += n;
            while (Number1 > 30)
            {
                Number1 -= 30;
                IncreaseNumber2(1);
            }
        }

        public override void DecreaseNumber1(int n)
        {
            Number1 -= n;
            while (Number1 <= 0)
            {
                Number1 += 30;
                DecreaseNumber2(1);
            }
        }

        public override void IncreaseNumber2(int n)
        {
            Number2 += n;
            while (Number2 > 12)
            {
                Number2 -= 12;
                IncreaseNumber3(1);
            }
        }

        public override void DecreaseNumber2(int n)
        {
            Number2 -= n;
            while (Number2 <= 0)
            {
                Number2 += 12;
                DecreaseNumber3(1);
            }
        }

        public override void IncreaseNumber3(int n)
        {
            Number3 += n;
        }

        public override void DecreaseNumber3(int n)
        {
```

```
            Number3 -= n;
        }

        public void Output()
        {

Console.WriteLine($"{Funcs.ChangeFormat(Number1)}.{Funcs.ChangeFormat(Number2)}.{Funcs.
ChangeFormat(Number3)}");
        }
    }
}
```

Console:

```
Enter the amount of dates: 3
Enter the amount of times: 3

Initial dates:
11.03.15
19.01.74
20.01.25

Initial times:
03:22:19
11:43:37
20:48:57

Dates that can be interpreted like time:
11.03.15
20.01.25

Changed dates:
11.03.15
14.01.74
20.01.25

Changed times:
03:37:19
11:58:37
21:03:57
```

Код Python:

main.py:

```
import Funcs

dates = Funcs.DatesGenerate()
times = Funcs.TimesGenerate()

print("\nInitial dates:")
Funcs.ListOutput(dates)
print("\nInitial times:")
Funcs.ListOutput(times)

dates_like_time = Funcs.DatesLikeTime(dates)
print("\nDates that can be interpreted like time:")
Funcs.ListOutput(dates_like_time)

Funcs.ChangeDates(dates)
Funcs.ChangeTimes(times)

print("\nChanged dates:")
Funcs.ListOutput(dates)
print("\nChanged times:")
Funcs.ListOutput(times)
```

Funcs.py:

```
import random
from TDate import TDate
from TTime import TTime


def DatesGenerate():
    dates = []
    n = int(input("Enter the amount of dates: "))
    for i in range(n):
        day = random.randint(1, 30)
        month = random.randint(1, 12)
        year = random.randint(0, 100)
        date = TDate(day, month, year)
        dates.append(date)
    return dates

def TimesGenerate():
    times = []
    n = int(input("Enter the amount of times: "))
    for i in range(n):
        hours = random.randint(0, 23)
        minutes = random.randint(0, 59)
        seconds = random.randint(0, 59)
        time = TTime(hours, minutes, seconds)
        times.append(time)
    return times

def ListOutput(list):
    for obj in list:
        obj.Output()

def IsLikeTime(date):
    if date.Number1 < 24 and date.Number3 < 60:
        return True
    return False

def ChangeDates(dates):
    for date in dates:
        if not IsLikeTime(date):
```

```
            date.DecreaseNumber1(5)

def ChangeTimes(times):
    for time in times:
        time.IncreaseNumber2(15)

def ChangeFormat(n):
    n_str = str(n)
    if n < 10:
        n_str = "0" + n_str
    return n_str

def DatesLikeTime(dates):
    dates_like_time = []
    for date in dates:
        if IsLikeTime(date):
            dates_like_time.append(date)
    return dates_like_time
```

TTriad.py:

```
from abc import abstractmethod


class TTriad:
    def __init__(self, number1, number2, number3):
        self.__number1 = number1
        self.__number2 = number2
        self.__number3 = number3

    @property
    def Number1(self):
        return self.__number1

    @Number1.setter
    def Number1(self, number1):
        self.__number1 = number1

    @property
    def Number2(self):
        return self.__number2

    @Number2.setter
    def Number2(self, number2):
        self.__number2 = number2

    @property
    def Number3(self):
        return self.__number3

    @Number3.setter
    def Number3(self, number3):
        self.__number3 = number3


    @abstractmethod
    def IncreaseNumber1(self, n):
        pass

    @abstractmethod
    def DecreaseNumber1(self, n):
        pass

    @abstractmethod
    def IncreaseNumber2(self, n):
        pass
```

```
    @abstractmethod
    def DecreaseNumber2(self, n):
        pass

    @abstractmethod
    def IncreaseNumber3(self, n):
        pass

    @abstractmethod
    def DecreaseNumber3(self, n):
        pass
```

TTime.cs:

```
from TTriad import TTriad
import Funcs


class TTime(TTriad):
    def __init__(self, number1, number2, number3):
        super().__init__(number1, number2, number3)

    def IncreaseNumber1(self, n):
        self.Number1 += n
        while (self.Number1 >= 24):
            self.Number1 -= 24

    def DecreaseNumber1(self, n):
        self.Number1 -= n
        while self.Number1 < 0:
            self.Number1 += 24

    def IncreaseNumber2(self, n):
        self.Number2 += n
        while self.Number2 >= 60:
            self.Number2 -= 60
            self.IncreaseNumber1(1)

    def DecreaseNumber2(self, n):
        self.Number2 -= n
        while self.Number2 < 0:
            self.Number2 += 60
            self.DecreaseNumber1(1)

    def IncreaseNumber3(self, n):
        self.Number3 += n
        while self.Number3 >= 60:
            self.Number3 -= 60
            self.IncreaseNumber2(1)

    def DecreaseNumber3(self, n):
        self.Number3 -= n
        while self.Number3 < 0:
            self.Number3 += 60
            self.DecreaseNumber2(1)


    def Output(self):
        print(

f"{Funcs.ChangeFormat(self.Number1)}:{Funcs.ChangeFormat(self.Number2)}:{Funcs.ChangeFo
rmat(self.Number3)}")
```

TDate.cs:

```
import Funcs
from TTriad import TTriad
```

```
class TDate(TTriad):
    def __init__(self, number1, number2, number3):
        super().__init__(number1, number2, number3)

    def IncreaseNumber1(self, n):
        self.Number1 += n
        while self.Number1 > 30:
            self.Number1 -= 30
            self.IncreaseNumber2(1)

    def DecreaseNumber1(self, n):
        self.Number1 -= n
        while self.Number1 <= 0:
            self.Number1 += 30
            self.DecreaseNumber2(1)


    def IncreaseNumber2(self, n):
        self.Number2 += n
        while self.Number2 > 12:
            self.Number2 -= 12
            self.IncreaseNumber3(1)

    def DecreaseNumber2(self, n):
        self.Number2 -= n
        while self.Number2 <= 0:
            self.Number2 += 12
            self.DecreaseNumber3(1)


    def IncreaseNumber3(self, n):
        self.Number3 += n

    def DecreaseNumber3(self, n):
        self.Number3 -= n

    def Output(self):

print(f"{Funcs.ChangeFormat(self.Number1)}.{Funcs.ChangeFormat(self.Number2)}.{Funcs.Ch
angeFormat(self.Number3)}")
```

Console:

```
Enter the amount of dates: 3
Enter the amount of times: 3

Initial dates:
08.08.10
25.06.06
18.05.54

Initial times:
02:14:40
10:41:53
01:32:41

Dates that can be interpreted like time:
08.08.10
18.05.54

Changed dates:
08.08.10
20.06.06
18.05.54

Changed times:
02:29:40
10:56:53
01:47:41
```