

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 № 5 з дисципліни
«Основи програмування - 2.
Модульне програмування»

«Успадкування та поліморфізм»

Варіант 5

Виконав студент

ІП-15, Буяло Дмитро Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірів

Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 5

Успадкування та поліморфізм

Мета – вивчити механізми створення і використання класів та об'єктів.

Індивідуальне завдання

Варіант 5

Завдання

Створити клас `TPair`, який представляє пару чисел і містить методи для їх інкрементування / декрементування. Реалізувати класи нащадки `TTime` ("години:хвилини") та `TMoney` ("гривні.копійки"). Згенерувати поступово випадковим чином n пар (час, гроші), де час – тривалість виконання певної роботи, а гроші – вартість однієї хвилини роботи працівника. Обчислити витрати на виконання кожної з робіт.

1. Код на C++

```
Lab_5_c++.cpp*  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp  Functions.h
Lab_5_c++  (Глобальная область)  main
1  #include "Classes.h"
2  #include "Storage.h"
3  #include "Functions.h"
4  int main() {
5      srand(time(NULL));
6      setlocale(LC_ALL, "Russian"); // для '₽'
7      //system("color 02"); зелёненький текст))
8      int n = input();
9      cout << " № | Hours worked | Price per minute | To pay" << endl;
10     TStorage* arr = new TStorage[n];
11     for (int i = 0; i < n; i++) {
12         TTime b;
13         TMoney c;
14         arr[i] = TStorage(b, c);
15         beautifyOut(i, b, c, arr);
16     }
17     cout << "Total: " << arr[n-1].getTotal();
18     int fTime, sTime;
19     cout << "\nEnter hours: "; cin >> fTime;
20     cout << "Enter minutes: "; cin >> sTime;
21     invalidInputTT(sTime);
22     TTime a(fTime, sTime);
23     output(a);
24     int fMoney, sMoney;
25     cout << "\nEnter hryvnias: "; cin >> fMoney;
26     cout << "Enter kopecks: "; cin >> sMoney;
27     invalidInputTM(sMoney);
28     TMoney d(fMoney, sMoney);
29     output(d);
30     delete[] arr;
```

```

Lab_5_c++.cpp  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp  Functions.h
Lab_5_c++  TPair  print
1  #pragma once
2  #include <iostream>
3  #include <iomanip>
4
5  using namespace std;
6
7  class TPair {
8  protected:
9      int first;
10     int second;
11  public:
12     int getFirst();
13     void setFirst(int a);
14     int getSecond();
15     void setSecond(int b);
16     virtual void print(); //вывод с использованием полиморфизма
17     TPair(int a, int b) : first(a), second(b) {}
18     TPair() {}
19     ~TPair() {}
20     TPair operator++(); //перегруженные операторы для инкрементации
21     TPair operator++(int notused);
22     TPair operator--();
23     TPair operator--(int notused); // или декрементации
24     virtual int update(); // виртуальная функция для операторов
25     void transfer(int a);
26     void transferBack(int a);
27 };
28
29 class TTime : public TPair {
30  public:
31     TTime() : TPair(rand() % 24, rand() % 60) {}
32     TTime(int a, int b) : TPair(a, b) {}
33     ~TTime() {}
34     void print() override;
35     int update() override;
36 };
37
38 class TMoney : public TPair {
39  public:
40     TMoney() : TPair(rand() % 100, rand() % 100) {}
41     TMoney(int a, int b) : TPair(a, b) {}
42     ~TMoney() {}
43     void print() override;
44     int update() override;
45 };

```

Основи програмування – 2. Модульне програмування

```
Lab_5_c++.cpp  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp  Functions.h
Lab_5_c++      -> TPair      - getS

1  #include "Classes.h"
2
3
4  int TPair::getFirst() {
5      return first;
6  }
7  void TPair::setFirst(int a) {
8      first = a;
9  }
10 int TPair::getSecond() {
11     return second;
12 }
13 void TPair::setSecond(int b) {
14     second = b;
15 }
16
17 void TPair::print() {
18     cout << first << " " << second;
19 }
20
21 TPair TPair::operator++() {
22     ++second;
23     transfer(update());
24     return *this;
25 }
26 TPair TPair::operator++(int notused) {
27     second++;
28     transfer(update());
29     return *this;
30 }
31 TPair TPair::operator--() {
32     --second;
33     transferBack(update());
34     return *this;
35 }
36 TPair TPair::operator--(int notused) {
37     second--;
38     transferBack(update());
39     return *this;
40 }
41
42 int TPair::update() { return second; }
43
44 void TPair::transfer(int a) {
45     if (second > (a - 1)) {
46         first += second / a;
47         second %= a;
48     }
49 }
50
51 void TPair::transferBack(int a) {
52     if (second < 0) {
53         first--;
54         second += a;
55     }
56     if (first < 0) {
57         first = 0;
58         second = 0;
59     }
60 }
61
62 void TTime::print() {
63     if (second < 10) cout << first << ":0" << second;
64     else cout << first << ":" << second;
65 }
66
67
68
69 int TTime::update() { // максимальное значение второго числа
70     return 60;
71 }
72
73
74
75 void TMoney::print() {
76     if (second < 10) cout << first << ".0" << second;
77     else cout << first << "." << second;
78 }
79
80 int TMoney::update() {
81     return 100;
82 }
```

```

Lab_5_++.cpp  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp  Functions.h
Lab_5_++      (Глобальная область)

1  #pragma once
2  #include <iostream>
3  #include <iomanip>
4  #include "Classes.h"
5  #include "Storage.h"
6
7  using namespace std;
8
9
10 int input();
11 void invalidInput(int&);
12 void invalidInputTT(int&);
13 void invalidInputTM(int&);
14 void beautifyOut(int, TTime, TMoney, TStorage*);
15 double calculate(TTime, TMoney);
16 void output(TTime);
17 void output(TMoney);

```

```

Lab_5_++.cpp  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp  Functions.h
Lab_5_++      TStorage

1  #pragma once
2  #include <iostream>
3  #include <iomanip>
4  #include "Classes.h"
5
6  using namespace std;
7
8  class TStorage {
9      TTime time;
10     TMoney money;
11     double result;
12     double total;
13 public:
14     double getRes();
15     void setRes(double r);
16     double getTotal();
17     void setTotal(double t);
18     double calculate(TTime a, TMoney b);
19     TStorage(TTime a, TMoney b);
20     TStorage() {}
21     ~TStorage() {}
22 };

```

```

Lab_5_c++.cpp  Classes.cpp  Classes.h  Storage.cpp*  Storage.h  Functions.cpp  Functions.h
Lab_5_c++  → TStorage  calcul

1  #include "Storage.h"
2  double TStorage::getRes() {
3      return result;
4  }
5
6  void TStorage::setRes(double r) {
7      result = r;
8  }
9
10 double TStorage::getTotal() {
11     return total;
12 }
13
14 void TStorage::setTotal(double t) {
15     total = t;
16 }
17 double TStorage::calculate(TTime a, TMoney b) {
18     double res;
19     double minutes, price;
20     minutes = a.getFirst() * 60 + a.getSecond();
21     price = b.getFirst() + b.getSecond() * 0.01;
22     res = minutes * price;
23     return res;
24 }
25 TStorage::TStorage(TTime a, TMoney b) {
26     time = a;
27     money = b;
28     result = calculate(time, money);
29     total += result;
30 }

```


Основи програмування – 2. Модульне програмування

```
Lab_5_c++.cpp  Classes.cpp  Classes.h  Storage.cpp  Storage.h  Functions.cpp*  Functions.h
Lab_5_c++      (Глобальная область)  invali

1  #include "Functions.h"
2
3
4  int input() {
5      int n;
6      cout << "Enter the number of pairs: "; cin >> n;
7      invalidInput(n);
8      cout << endl;
9      return n;
10 }
11 void beautifyOut(int i, TTime time, TMoney money, TStorage* arr) {
12     cout << setw(2) << (i + 1) << " |";
13     cout << setw(6); time.print(); cout << setw(6) << "|";
14     cout << setw(8); money.print(); cout << setw(8) << "|";
15     cout << " " << fixed << setprecision(2) << arr[i].getRes() << endl;
16 }
17 void invalidInput(int& n) {
18     while (!cin || n < 1) {
19         cin.clear();
20         cin.ignore(64, '\n');
21         cout << "Your input is wrong, try again: "; cin >> n;
22     }
23 }
24 void invalidInputTT(int& t) {
25     while (!cin || t < 0 || t > 59) {
26         cin.clear();
27         cin.ignore(64, '\n');
28         cout << "Your input is wrong, try again: "; cin >> t;
29     }
30 }
31 void invalidInputTM(int& m) {
32     while (!cin || m < 0 || m > 99) {
33         cin.clear();
34         cin.ignore(64, '\n');
35         cout << "Your input is wrong, try again: "; cin >> m;
36     }
37 }
38 void output(TTime a) {
39     cout << "\nNew TTime object: "; a.print();
40     a++;
41     cout << "\nPostfix incrementing: "; a.print();
42     ++a;
43     cout << "\nPrefix incrementing: "; a.print();
44     a--;
45     cout << "\nPostfix decrementing: "; a.print();
46     --a;
47     cout << "\nPrefix decrementing: "; a.print();
48     cout << endl;
49 }
50 void output(TMoney a) {
51     cout << "\nNew TMoney object: "; a.print();
52     a++;
53     cout << "\nPostfix incrementing: "; a.print();
54     ++a;
55     cout << "\nPostfix decrementing: "; a.print();
56     --a;
57     cout << "\nPrefix decrementing: "; a.print();
58     ++a;
59     cout << "\nPrefix incrementing: "; a.print();
60     cout << endl;
}
```

2. Результат виконання на C++

Консоль отладки Microsoft Visual Studio

```
Enter the number of pairs: 5

№ | Hours worked | Price per minute | To pay
1 | 8:32 | 42.66 | 21841.92
2 | 23:59 | 79.03 | 113724.17
3 | 14:49 | 71.48 | 63545.72
4 | 11:01 | 60.60 | 40056.60
5 | 14:19 | 24.65 | 21174.35
Total: 260342.76
Enter hours: 3
Enter minutes: 59

New TTime object: 3:59
Postfix incrementing: 4:00
Prefix incrementing: 4:01
Postfix decrementing: 4:00
Prefix decrementing: 3:59

Enter hryvnias: 40
Enter kopecks: 0

New TMoney object: 40.00
Postfix incrementing: 40.01
Postfix decrementing: 40.00
Prefix decrementing: 39.99
Prefix incrementing: 40.00
```

3. Код на Python

```
main.py x TTime.py x TMoney.py x TPair.py x TStorage.py x
1 from TPair import TPair
2
3 |
4 class TTime(TPair):
5     def __init__(self, first: int, second: int):
6         super().__init__(first, second)
7
8     def printT(self):
9         if self.second < 10:
10             print(f'Time worked: {self.first}:0{self.second}')
11         else:
12             print(f'Time worked: {self.first}:{self.second}')
13
14     def updated(self):
15         return 60
```



```

main.py × TTime.py × TMoney.py × TPair.py × TStorage.py ×
1 class TPair:
2     def __init__(self, first: int, second: int):
3         self.first = first
4         self.second = second
5
6     def get_first(self):
7         return self.first
8
9     def set_first(self, value: int):
10        self.first = value
11
12    def get_second(self):
13        return self.second
14
15    def set_second(self, value: int):
16        self.second = value
17
18    first1 = property(get_first, set_first)
19    second2 = property(get_second, set_second)
20
21    def ink(self):
22        self.second += 1
23        self.transfer(self.updated())
24
25    def dec(self):
26        self.second -= 1
27    def dec(self):
28        self.second -= 1
29        self.transferBack(self.updated())
30
31    def updated(self):
32        return self.second
33
34    def transfer(self, tran: int):
35        if self.second > (tran - 1):
36            self.first += self.second // tran
37            self.second %= tran
38
39    def transferBack(self, tran8: int):
40        if self.second < 0:
41            self.first -= 1
42            self.second += tran8
43        if self.first < 0:
44            self.first = 0
45            self.second = 0
46
47    def printT(self):
48        print(self.first, " ", self.second)

```

```

main.py × TTime.py × TMoney.py × TPair.py × TStorage.py ×
1  from TPair import TPair
2
3  |
4  class TMoney(TPair):
5      def __init__(self, first: int, second: int):
6          super().__init__(first, second)
7
8      def printT(self):
9          if self.second < 10:
10             print(f'Price per minute: {self.first}.0{self.second}')
11          else:
12             print(f'Price per minute: {self.first}.{self.second}')
13
14      def updated(self):
15          return 100

```

```

main.py × TTime.py × TMoney.py × TPair.py × TStorage.py ×
1  from TTime import TTime
2  from TMoney import TMoney
3
4
5  class TStorage:
6      def __init__(self, time: TTime, money: TMoney):
7          self.time = time
8          self.money = money
9          self.result = calculate(time, money)
10         # self.total += self.result
11
12     def getRes(self):
13         return self.result
14
15     def setRes(self, value: float):
16         self.result = value
17
18
19     def calculate(a: TTime, b: TMoney):
20         minutes = a.first1 * 60 + a.second2
21         price = b.first1 + b.second2 * 0.01
22         return minutes * price

```

```
main.py x TTime.py x TMoney.py x TPair.py x TStorage.py x
1  from random import randint
2  from TStorage import *
3  n = int(input("Enter the number of pairs: "))
4  arr = [None] * n
5  for i in range(n):
6      b = TTime(randint(0, 23), randint(0, 59))
7      c = TMoney(randint(0, 99), randint(0, 99))
8      arr[i] = TStorage(b, c)
9      print(f'\ni = {i + 1}:')
10     b.printT()
11     c.printT()
12     print("To pay:", round(arr[i].getRes(), 2))
13     print()
14     a = TTime(5, 0)
15     a.printT()
16     a.ink()
17     a.printT()
18     a.dec()
19     a.dec()
20     a.printT()
21     print()
22     d = TMoney(0, 99)
23     d.printT()
24     d.ink()
25     d.ink()
26     d.printT()
```

4. Результат виконання на Python

```
Enter the number of pairs: 2

i = 1:
Time worked: 14:21
Price per minute: 86.52
To pay: 74493.72

i = 2:
Time worked: 17:09
Price per minute: 34.25
To pay: 35243.25

Time worked: 5:00
Time worked: 5:01
Time worked: 4:59

Price per minute: 0.99
Price per minute: 1.01

Process finished with exit code 0
```

5. Висновок

Під час виконання п'ятої лабораторної роботи, розглянули на практиці роботу з класами з використанням перевантажених операторів, поліморфних функцій, різних конструкторів та наслідуванням.

В результаті була створена програма, яка за допомогою різних класів представляє пару чисел (час та ціна) і методи їх інкрементування та декрементування.