|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 7 |

**Название:**

Простые объекты

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-25 Б |  |  |  |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Scan0023Scan0026 |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Вариант 8**

Задание: Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Протестировать все методы класса. Поля класса должны быть скрытыми (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – бонусная карта. Поля: номер карты, процент начисления бонусов за покупку, текущий бонусный баланс. Методы: процедура инициализации; процедура вывода значений полей на экран; процедура совершения покупки на заданную сумму с соответствующим начислением бонусов; процедура расходования некоторого количества бонусов; функция, возвращающая текущий бонусный баланс.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

**Код программы (без конструктора):**

#include <iostream>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

using namespace std;

class TKarta {

private:

int numb=0;

int per=0;

float bal=0;

public:

void Init(int an, int ap, float ab) {

numb = an; per = ap; bal = ab;

}

void Print() {

cout << "Number of a card: " << numb <<

"$\nThe percent of discount: " << per <<

"%\nThe balance of card: " << bal << "$\n";

}

void buy(int sum) {

float ss = sum;

bal = bal + ss\*per/100;

}

void spend(int sum) {

bal = bal - sum;

}

float cerBal() {return bal;}

};

int main() {

int n, p, summ; float b; char s[6];

TKarta card;

srand((unsigned)time(NULL));

n = 1000 + rand() % (9999 - 1000 + 1);

p = 1 + rand() % (25 - 1 + 1);

b = 150 + static\_cast <float> (rand()) / (static\_cast <float> (RAND\_MAX / (350 - 150)));

card.Init(n, p, b);

cout << "Data: \n"; card.Print();

cout << "Input the purchase amount: ";

cin >> summ; card.buy(summ);

cout << "Would you like to spend bonuses?\n";

cin.ignore();

gets\_s(s, 6);

if (strcmp(s, "Yes") == 0 || strcmp(s, "yes") == 0) {

cout << "Input the amount: "; cin >> summ;

card.spend(summ);

}

cout << "The current balance: "<<card.cerBal();

return 0;

}

**Код программы (с консруктором):**

#include <iostream>

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

using namespace std;

class TKarta {

private:

int numb;

int per;

float bal;

public:

TKarta(int an, int ap, float ab) :numb(an), per(ap), bal(ab) {}

void Print() {

cout << "Number of a card: " << numb <<

"\nThe percent of discount: " << per <<

"%\nThe balance of card: " << bal << "$\n";

}

void buy(int sum) {

float ss = sum;

bal = bal + ss \* per / 100;

}

void spend(int sum) {

bal = bal - sum;

}

float cerBal() { return bal; }

};

int main() {

int n, p, summ; float b; char s[6];

srand((unsigned)time(NULL));

n = 1000 + rand() % (9999 - 1000 + 1);

p = 1 + rand() % (25 - 1 + 1);

b = 150 + static\_cast <float> (rand()) / (static\_cast <float> (RAND\_MAX / (350 - 150)));

TKarta card(n, p, b);

cout << "Data: \n"; card.Print();

cout << "Input the purchase amount: ";

cin >> summ; card.buy(summ);

cout << "Would you like to spend bonuses?\n";

cin.ignore();

gets\_s(s, 6);//доделать

if (strcmp(s, "Yes") == 0 || strcmp(s, "yes") == 0) {

cout << "Input the amount: "; cin >> summ;

card.spend(summ);

}

cout << "The current balance: " << card.cerBal();

return 0;

}

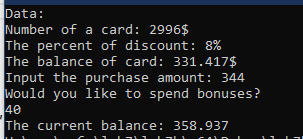


Рисунок 1 – работающая версия программы

|  |
| --- |
| TKarta |
| -numb  -per  -bal |
| +Init()  +Print()  +buy()  +spend()  +cerBal() |

|  |
| --- |
| TKarta |
| -numb  -per  -bal |
| +TKarta()  +Print()  +buy()  +spend()  +cerBal() |

Рисунок 2 – диаграмма классов (без и с конструктором)



Рисунок 3 – объектная декомпозиция программы без конструктора



Рисунок 4 – объектная декомпозиция программы с конструктором

**Вывод:** мы научились работать с классами и объектами как без конструктора, так и с его наличием на языке C++