**Новосибирский государственный технический университет**

Лабораторная работа №1 по дисциплине **программирование**

Тема: Разработка классов, создание конструкторов и деструкторов. Использование статических членов класса

|  |  |
| --- | --- |
| Группа: АВТ-907  Студент: Смоленский И.В. | Преподаватель: Булатов А. Д. |

Новосибирск, 2020

**Задание**:

Вариант 14.

Постpоить класс Bankomat, моделирующий работу банкомата. Класс должен включать следующие поля: id-номер банкомата (динамическая строка, содержащая цифры и буквы), текущая сумма денег в банкомате, максимальная сумма, которую можно снять разово. Класс должен обеспечивать пpостейшие функции для pаботы с данными класса: загрузка денег в автомат, снятие определенной суммы денег, строковое представление объекта.

Теоретические сведения:

## Компоненты класса

Класс - это определяемый пользователем тип. Описание класса очень похоже на описание структуры в Си. В этом смысле класс является расширением понятия структуры. В простейшем случае класс можно определить с помощью конструкции:

***тип\_класса имя\_класса {список\_членов\_класса};***

где

*тип\_класса* – одно из служебных слов **class, struct, union**;

*имя\_класса* – идентификатор;

*список\_членов\_класса* – определения и описания типизированных данных и принадлежащих классу функций.

*Функции* – это методы класса, определяющие операции над объектом.

*Данные* – это поля объекта, образующие его структуру. Значения полей определяет состояние объекта.

Для описания объекта класса (экземпляра класса) используется конструкция

***имя\_класса имя\_объекта;***

Для изменения видимости компонент в определении класса можно использовать спецификаторы доступа: **public, private, protected**.

Общедоступные (public) компоненты класса доступны в любой части программы. Они могут использоваться любой функцией как внутри данного класса, так и вне его. Доступ извне осуществляется через имя объекта:

***имя\_объекта.имя\_члена\_класса***

***ссылка\_на\_объект.имя\_члена\_класса***

***указатель\_на\_объект->имя\_члена\_класса***

Собственные (private) компоненты класса локализованы в классе и не доступны извне. Они могут использоваться функциями – членами данного класса и функциями – “друзьями” класса.

Защищенные (protected) компоненты доступны внутри класса и в производных классах.

Проектирование программы:

Класс Fraction включает в себя функции по работе с классом и переменные типа int numerator - числитель и denominator – знаменатель.

Для работы с классом Fraction были созданы следующие функции:

Конструкторы:

Конструктор по умолчанию

Bankomat::Bankomat()

{

m\_id = s\_idGenerator++;

id = new char[100];

sumBank = 100000;

sumMaxGet = 5000;

};

Конструктор копирования, который создаёт новый объект, с такими же параметрами, как и переданный.

Bankomat::Bankomat(const Bankomat& Bankomat) :

id(Bankomat.id), sumBank(Bankomat.sumBank),sumMaxGet(Bankomat.sumMaxGet)

{

m\_id = s\_idGenerator++;

}

Конструктор, которому передаются параметры на вход

Bankomat::Bankomat(char\* id1, unsigned sumBank1, unsigned sumMaxGet1)

{

id = id1;

sumBank = sumBank1;

sumMaxGet = sumMaxGet1;

m\_id = s\_idGenerator++;

}

Деструктор  
  
Bankomat::~Bankomat()

{

delete[] id;

}

Функции, которые отвечают за загрузку и снятие денег:

unsigned loadMoney(unsigned);

unsigned takeMoney(unsigned);

Функция вывода всей информации о банкомате:

char\* toString();

Реализация класса Fraction:

#include <iostream>

#include "stdio.h"

#include <time.h>

#include <algorithm>

#include <exception>

#include "Bankomat.h"

using namespace std;

int Bankomat::s\_idGenerator = 1;

Bankomat::Bankomat()

{

m\_id = s\_idGenerator++;

id = new char[100];

sumBank = 100000;

sumMaxGet = 5000;

};

Bankomat::Bankomat(const Bankomat& Bankomat) :

id(Bankomat.id), sumBank(Bankomat.sumBank),sumMaxGet(Bankomat.sumMaxGet)

{

m\_id = s\_idGenerator++;

}

Bankomat::Bankomat(char\* id1, unsigned sumBank1, unsigned sumMaxGet1)

{

id = id1;

sumBank = sumBank1;

sumMaxGet = sumMaxGet1;

m\_id = s\_idGenerator++;

}

Bankomat::Bankomat(const char\* id1, unsigned sumBank1, unsigned sumMaxGet1)

{

strcpy(id, id1);

sumBank = sumBank1;

sumMaxGet = sumMaxGet1;

m\_id = s\_idGenerator++;

}

Bankomat::~Bankomat()

{

delete[] id;

}

unsigned Bankomat::loadMoney(unsigned sumLoad)

{

try

{

if (sumLoad < 0)

{

throw std::invalid\_argument("Invalid arg");

}

}

catch (const std::invalid\_argument & err)

{

std::cerr << " Invalid argument: " << err.what() << std::endl;

}

sumBank += sumLoad;

return sumBank;

}

unsigned Bankomat::takeMoney(unsigned sumTake)

{

try

{

if (sumTake < 0)

{

throw std::invalid\_argument("Invalid arg");

}

}

catch (const std::invalid\_argument & err)

{

std::cerr << " Invalid argument: " << err.what() << std::endl;

}

if ((sumBank >= sumTake) && (sumTake <= sumMaxGet))

{

sumBank -= sumTake;

}

return sumBank;

}

char\* Bankomat::toString()

{

char\* str = new char[100];

sprintf\_s(str, 100, "ID = %s, Sum in Bankomat = %d, Max sum of getting = %d", this->id, this->sumBank, this->sumMaxGet);

return str;

}

Вывод:  
Мной был разработан класс работы Bankomat, который содержит приватные данные, имеет методы работы с зачислениями и снятиями денег из него.