|  |  |
| --- | --- |
| **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 **09.03.01 Информатика и вычислительная техника**

**Отчет**

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

**Название:**

Простые объекты

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | С.В. Астахов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**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 – ангар. Параметры: площадь и процент занятой площади. Методы: процедура инициализации полей, процедура вывода на экран значений полей и функция, определяющая и возвращающая по запросу свободную площадь помещения.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

**Исходный код**

(файл Source.cpp)

*#include <iostream>*

*#include <conio.h>*

*using namespace std;*

*int main()*

*{*

*class CA {*

*protected: int s;*

*int perc;*

*public:*

*void set(int argS, int argPerc) {*

*s = argS;*

*perc = argPerc;*

*}*

*void get() {*

*printf("\n square: %d \n percent occupied: %d %%", s, perc);*

*}*

*float fspace() {*

*return (s - (s \* (perc / 100.0)));*

*}*

*};*

*class CB : public CA {*

*public:*

*CB(int argS, int argPerc) {*

*s = argS;*

*perc = argPerc;*

*}*

*CB() {*

*s = 0;*

*perc = 100;*

*}*

*};*

*int x1, y1, x2, y2;*

*CA a;*

*//CB b1();*

*puts("Enter squareA, percentA, squareB, percentB");*

*scanf\_s("%d %d %d %d", &x1, &y1, &x2, &y2);*

*puts("\n >>> for a:");*

*a.set(x1, y1);*

*a.get();*

*printf("\n free space: %3.2f", a.fspace());*

*puts("\n \n >>> for b:");*

*CB b1(x2, y2);*

*b1.get();*

*printf("\n free space: %3.2f", b1.fspace());*

*puts("\n \n press any key");*

*\_getch();*

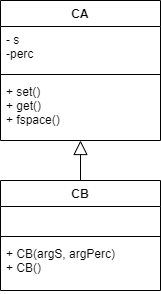
*return 0;*

*}*

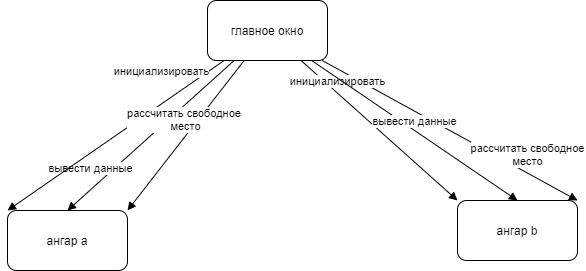
**Тесты**

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемые выходные данные | Выходные данные |
| 100 20 200 40 | >>> for a:  square: 100  percent occupied: 20 %  free space: 80.00  >>> for b:  square: 200  percent occupied: 40 %  free space: 120.00  press any key | >>> for a:  square: 100  percent occupied: 20 %  free space: 80.00  >>> for b:  square: 200  percent occupied: 40 %  free space: 120.00  press any key |
| 33 50 45 25 | >>> for a:  square: 33  percent occupied: 50 %  free space: 16.50  >>> for b:  square: 45  percent occupied: 25 %  free space: 33.75  press any key | >>> for a:  square: 33  percent occupied: 50 %  free space: 16.50  >>> for b:  square: 45  percent occupied: 25 %  free space: 33.75  press any key |
| 120 15 150 30 | >>> for a:  square: 120  percent occupied: 15 %  free space: 102.00  >>> for b:  square: 150  percent occupied: 30 %  free space: 105.00  press any key | >>> for a:  square: 120  percent occupied: 15 %  free space: 102.00  >>> for b:  square: 150  percent occupied: 30 %  free space: 105.00  press any key |

**Диаграмма классов**

****

**Объектная декомпозиция**

****

**Вывод**

* В общем и целом объектная модель в C++ напоминает Delphi, однако стоит отметить, что в C++ существуют различные типы конструкторов, вызываемых при выделении памяти под объект и наличие спецификаторов доступа при наследовании.