**Задачи на экзамен**

**Задание 1. (**– **2 бала)**

Создать класс окружностей на плоскости, описав в нём все необходимые свойства, подобрав им понятные имена и правильные типы данных. Описать в классе конструктор, позволяющий при создании нового объекта явно задать все его свойства, а также конструктор, позволяющий задать свойства с клавиатуры. Если это необходимо, то проверить допустимость значений свойств. Создать в классе метод, вычисляющий площадь круга. Создать в классе метод, вычисляющий расстояние между центрами окружностей. Создать в классе метод, сообщающий, касаются ли окружности.

#include<iostream>

#include<string>

#include<windows.h>

#include<stdlib.h>

#include<fstream>

#include<vector>

#include<algorithm>

#include<list>

#include<map>

#include<ctime>

#include<math.h>

using namespace std;

//l = 2pr

class circles

{

private:

int x;

int y;

int r;

int l;

public:

circles(int a, int b, int c, int d)

{

x = a;

y = b;

r = c;

l = d;

}

circles()

{

cin >> x;

cin >> y;

cin >> r;

cin >> l;

}

int getsqueare()

{

return r \* r \* 3.14;

}

int getdistance(circles b)

{

int len;

len = sqrt((x - b.x) \* (x - b.x) + (y - b.y) \* (y - b.y));

return len;

}

bool touch(circles b)

{

if (getdistance(b) > (r + b.r))

{

cout << "Окружности не пересекаються" << endl;

return false;

}

else if (getdistance(b) <= (r + b.r))

{

cout << "Окружности пересекаються" << endl;

return true;

}

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

circles a(2, 4, 1, 6), b(-1, -2, 1, 6);

cout << a.getsqueare() << endl;

cout << a.getdistance(b) << endl;

a.touch(b);

return 0;

}

**Задание 2. (3**– **4 бала)**

Создать класс рациональных дробей, выбрав для него необходимые свойства. Создать конструкторы, позволяющие явно задавать дробь, указывая числитель и знаменатель, а также создавать дробь, вводя числитель и знаменатель с клавиатуры. Добавить необходимые ограничения на допустимые значения. Реализовать в классе операции умножения и сложения дробей, с приведением к наибольшему общему знаменателю. **3 – 4 бали**

#include<iostream>

#include<string>

#include<windows.h>

#include<stdlib.h>

#include<fstream>

#include<vector>

#include<algorithm>

#include<list>

#include<map>

#include<ctime>

#include<math.h>

using namespace std;

//l = 2pr

class drib

{

private:

int chis;

int znam;

public:

drib(int x, int y)

{

chis = x;

znam = y;

while (znam == 0)

{

cout << "Znam ne mozhet = 0, vvedite znamennyk snova" << endl;

cin >> znam;

}

}

drib()

{

cin >> chis >> znam;

while (znam == 0)

{

cout << "Znam ne mozhet = 0, vvedite znamennyk snova" << endl;

cin >> znam;

}

}

drib(drib\* b)

{

znam = b->znam;

chis = b->chis;

}

drib \*operator +(drib b)

{

if (znam != b.znam)

{

b.chis = b.chis \* znam;

znam = znam \* b.znam;

chis = chis \* b.znam;

chis = chis + b.chis;

return this;

}

else

{

chis = chis + b.chis;

return this;

}

}

drib\* operator \*(drib b)

{

chis = chis \* b.chis;

znam = znam \* b.znam;

return this;

}

drib\* operator -(drib b)

{

if (znam != b.znam)

{

b.chis = b.chis \* znam;

znam = znam \* b.znam;

chis = chis \* b.znam;

chis = chis - b.chis;

return this;

}

else

{

chis = chis - b.chis;

return this;

}

}

drib\* operator /(drib b)

{

chis = chis / b.chis;

znam = znam / b.znam;

return this;

}

void print()

{

cout << chis << endl << "--" << endl << znam << endl << endl;;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

drib a(5, 7);

drib b(4, 2);

a - b;

a.print();

a\* b;

a.print();

a / b;

a.print();

a + b;

a.print();

return 0;

}

**Задание 3. Программа «Статистика» ( - 4 балов)**

Подсчитать, сколько раз каждое слово встречается в заданном тексте. Результат записать в коллекцию **map<TKey, TValue>**. Текс использовать из приложения

1. Вывести статистику по тексту в виде таблицы (рисунок 1).

**Приложение 1.**

Вот дом, Который построил Джек. А это пшеница, Которая в темном чулане хранится В доме, Который построил Джек. А это веселая птица-синица, Которая часто ворует пшеницу, Которая в темном чулане хранится В доме, Который построил Джек.



* Рисунок 1.

**Задание 4. Программа «Статистика» ( - 6 балов)**

В текстовом файле есть ведомость результатов сдачи экзаменов студенческой группы. Ведомость содержит для каждого студента фамилия, имя, отчество и оценки по пяти предметам. Студентов в группе не более 20 человек.

Написать программу (используя класс), которая предоставляет следующую информацию:

* список студентов (ФИО)
* список студентов, сдавших все экзамены только на 5,
* список студентов, имеющих хотя-бы одну тройку по экзаменам;
* список студентов, у которых есть двойки. Если студент имеет более чем одну двойку, он исключается из списка.

#include<iostream>

#include<string>

#include<windows.h>

#include<stdlib.h>

#include<fstream>

#include<vector>

#include<algorithm>

#include<list>

#include<map>

#include<ctime>

#include<math.h>

using namespace std;

class student

{

private:

char name[50];

char lastname[50];

char otch[50];

char math;

char physics;

char chemistry;

char english;

char biology;

public:

student(char \*n, char\* l, char \*o,char m, char p, char c, char e, char b)

{

strcpy(name, n);

strcpy(lastname, l);

strcpy(otch, o);

math = m;

physics = p;

chemistry = c;

english = e;

biology = b;

}

void print()

{

cout << name << " " << lastname << " " << otch << endl << "Math: " << math << "\nPhysics: " << physics << "\nChemistry: " << chemistry << "\nEnglish: " << english << "\nBiology: " << biology << endl;

}

char getm()

{

return math;

}

char getp()

{

return physics;

}

char getc()

{

return chemistry;

}

char gete()

{

return english;

}

char getb()

{

return biology;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

list<student> a;

int x = 0;

char q = 'q';

char name[50];

char lastname[50];

char otch[50];

char math;

char physics;

char chemistry;

char english;

char biology;

fstream file("d:\\spisok.txt", ios::in);

for (int i = 0; i < 5; i++)

{

file.get(name,50, ' ');

file.get(q);

file.get(lastname, 50, ' ');

file.get(q);

file.get(otch, 50, ' ');

file.get(q);

file.get(math);

file.get(q);

file.get(physics);

file.get(q);

file.get(chemistry);

file.get(q);

file.get(english);

file.get(q);

file.get(biology);

file.get(q);

student b(name, lastname, otch, math, physics, chemistry, english, biology);

a.push\_back(b);

}

cout << "1 - spisok studentov; 2 - zdavshie vse na 5; 3 - hotya bi odny 3;4 - s dvoikoi; 9 - exit" << endl;

while(x != 9)

{

cin >> x;

if (x == 1)

{

for (auto it = a.begin(); it != a.end(); it++)

{

it->print();

}

}

if (x == 2)

{

for (auto it = a.begin(); it != a.end(); it++)

{

if (it->getm() == '5' && it->getp() == '5' && it->getc() == '5' && it->gete() == '5' && it->getb() == '5')

{

it->print();

}

}

}

if (x == 3)

{

for (auto it = a.begin(); it != a.end(); it++)

{

if (it->getm() == '3' || it->getp() == '3' || it->getc() == '3' || it->gete() == '3' || it->getb() == '3')

{

it->print();

}

}

}

if (x == 4)

{

int count = 0;

for (auto it = a.begin(); it != a.end(); it++)

{

if (it->getm() == '2')

count++;

if (it->getp() == '2')

count++;

if (it->getc() == '2')

count++;

if (it->gete() == '2')

count++;

if (it->getb() == '2')

count++;

if (count == 1)

{

it->print();

}

count = 0;

}

}

}

return 0;

}

//Vasya Pupukin Vasilich 5 5 5 5 5

//Oleg Smolyakin Oleksiyovich 5 5 5 5 5

//Liudmilya Romaniuk Bogdanivna 3 4 4 5 4

//Nikita Kuznietsov Illich 5 5 2 5 3

//Tsipluk Dmytro Igorovich 3 2 2 4 5

**Задание 5. Реализация карточной игры - "Покер". (-12 балов)**