Домашнее задание по теме «Классы и объекты»

Выполнил: Поплавский Даниил, 11ИС-273

**Вопросы**

1. Для чего необходимо определение класса?

Определение класса необходимо по нескольким причинам: моделирование объектов реального мира, использование инкапсуляции, создание экземпляров, наследование, полиморфизм.

2. Класс имеет такое же отношение к объекту, как стандартный тип данных к переменной этого типа.

3. В определении класса члены класса с ключевым словом private доступны: а) любой функции программы; б) в случае, если вам известен пароль; в) методам этого класса; г) только открытым членам класса.

Ответ: в) методам этого класса

4. Напишите определение класса leverage, включающего одно закрытое поле типа int с именем crowbar и одним открытым методом с прототипом void pry().

class leverage {

private:

int crowbar;

public:

void pry() {

}

};

5. Истинно ли следующее утверждение: поля класса должны быть закрытыми?

Это утверждение не всегда истинно: не обязательно делать поля закрытыми, но это рекомендуется делать, чтобы обеспечить контролируемый доступ к данным и обеспечить надёжность и удобство использования класса.

6. Напишите оператор, создающий объект lever1 класса leverage, описанного в вопросе 4.

leverage lever1;

7. Операция точки (операция доступа к члену класса) объединяет следующие два элемента (слева направо): а) член класса и объект класса; б) объект класса и класс; в) класс и член этого класса; г) объект класса и член этого класса.

Ответ: г) объект класса и член этого класса

8. Напишите оператор, который вызовет метод pry() объекта lever1 (см. вопросы 4 и 6).

lever1.pry();

9. Методы класса, определенные внутри класса, по умолчанию являются методами-членами класса.

10. Напишите метод getcrow() для класса Leverage (см. вопрос 4), который будет возвращать значение поля crowbar. Метод следует определить внутри определения класса.

class leverage {

private:

int crowbar;

public:

void pry();

int getcrow() {

return crowbar;

}

};

11. Конструктор вызывается автоматически в момент создания объекта.

12. Имя конструктора совпадает с именем класса.

13. Напишите конструктор, который инициализирует нулевым значением поле crowbar класса leverage (см. вопрос 4). Конструктор следует определить внутри определения класса.

class leverage {

private:

int crowbar;

public:

leverage() : crowbar(0) {}

};

14. Верно или неверно следующее утверждение: класс может иметь более одного конструктора с одним и тем же именем?

Верно. Класс может иметь более одного конструктора с одним и тем же именем. Это называется перегрузкой конструкторов. Конструкторы могут иметь различные параметры или параметры по умолчанию, что позволяет создавать объекты с разными способами инициализации.

15. Методу класса всегда доступны данные: а) объекта, членом которого он является; б) класса, членом которого он является; в) любого объекта класса, членом которого он является; г) класса, объявленного открытым.

Ответ: а) объекта, членом которого он является

16. Предполагая, что метод getcrow(), описанный в вопросе 10, определен вне класса, объявите этот метод внутри класса.

class leverage {

private:

int crowbar;

public:

void pry();

int getcrow() {

return crowbar;

}

};

17. Напишите новую версию метода getcrow(), описанного в вопросе 10, которая определяется вне класса.

int leverage::getcrow() {

return crowbar;

}

18. Единственным формальным различием между структурами и классами в C++ является то, что по умолчанию все члены структуры являются открытыми (public), а в классе по умолчанию члены являются закрытыми (private).

19. Пусть определены три объекта класса. Сколько копий полей класса содержится в памяти? Сколько копий методов функций?

В каждом объекте класса содержится своя копия всех полей класса, т.е. в нашем случае будет три копии каждого поля класса в памяти.

Количество копий методов функций не зависит от количества объектов класса и остается одним экземпляром для всего класса.

20. Посылка сообщения объекту эквивалентна вызову метода объекта.

21. Классы полезны потому, что: а) не занимают памяти, если не используются; б) защищают свои данные от доступа со стороны других классов; в) собирают вместе все аспекты, касающиеся отдельной вещи; г) адекватно моделируют объекты реального мира.

Ответ: б), в) и г)

22. Истинно ли следующее утверждение: существует простой, но очень точный метод, позволяющий представлять решаемую задачу в виде совокупности объектов классов?

Да, это утверждение истинно. Объектно-ориентированное программирование предоставляет методологию, которая позволяет разбивать сложные задачи на более простые составляющие, а затем представлять эти составляющие в виде объектов и классов. Этот подход называется объектно-ориентированным анализом и проектированием (ООА и ООП) и является одним из основных принципов ООП.

23. Константный метод, вызванный для объекта класса: а) может изменять как неконстантные, так и константные поля; б) может изменять только неконстантные поля; в) может изменять только константные поля; г) не может изменять как неконстантные, так и константные поля.

Ответ: б) может изменять только неконстантные поля

24. Истинно ли следующее утверждение: объект, объявленный как константный, можно использовать только с помощью константных методов?

Да, это утверждение верно. Константные методы гарантируют, что они не изменяют состояние объекта, поэтому использование объекта с константным квалификатором `const` с неконстантными методами противоречит этому ограничению.

25. Напишите объявление (не определение) функции типа const void с именем aFunc(), которая принимает один константный аргумент jerry типа float.

class MyClass {

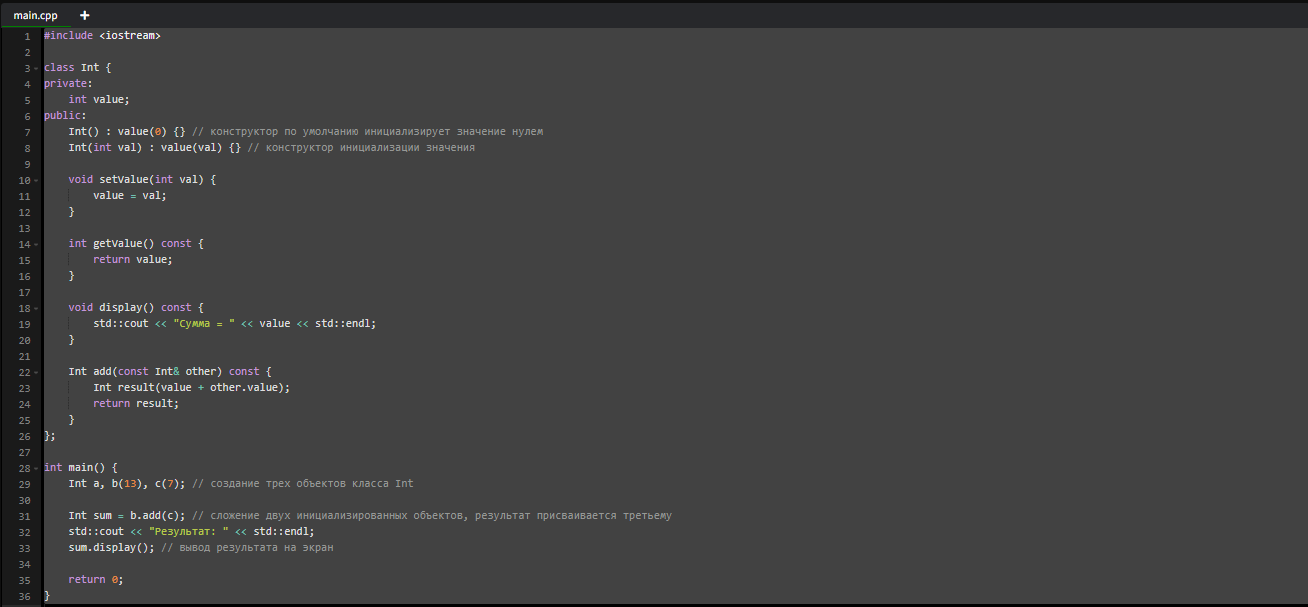
public:

const void aFunc(float jerry) const;

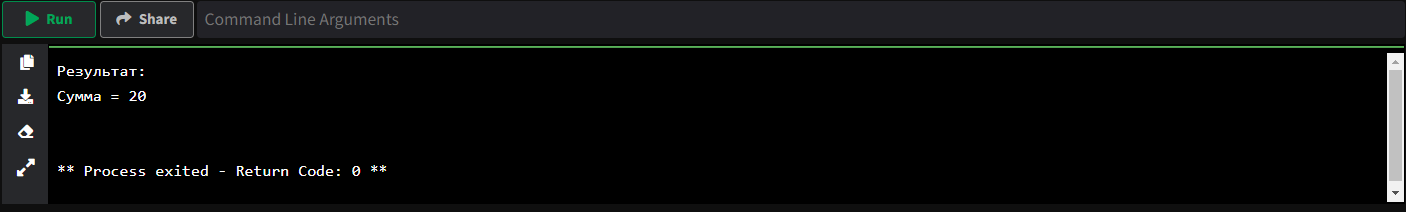
};

**Задания**

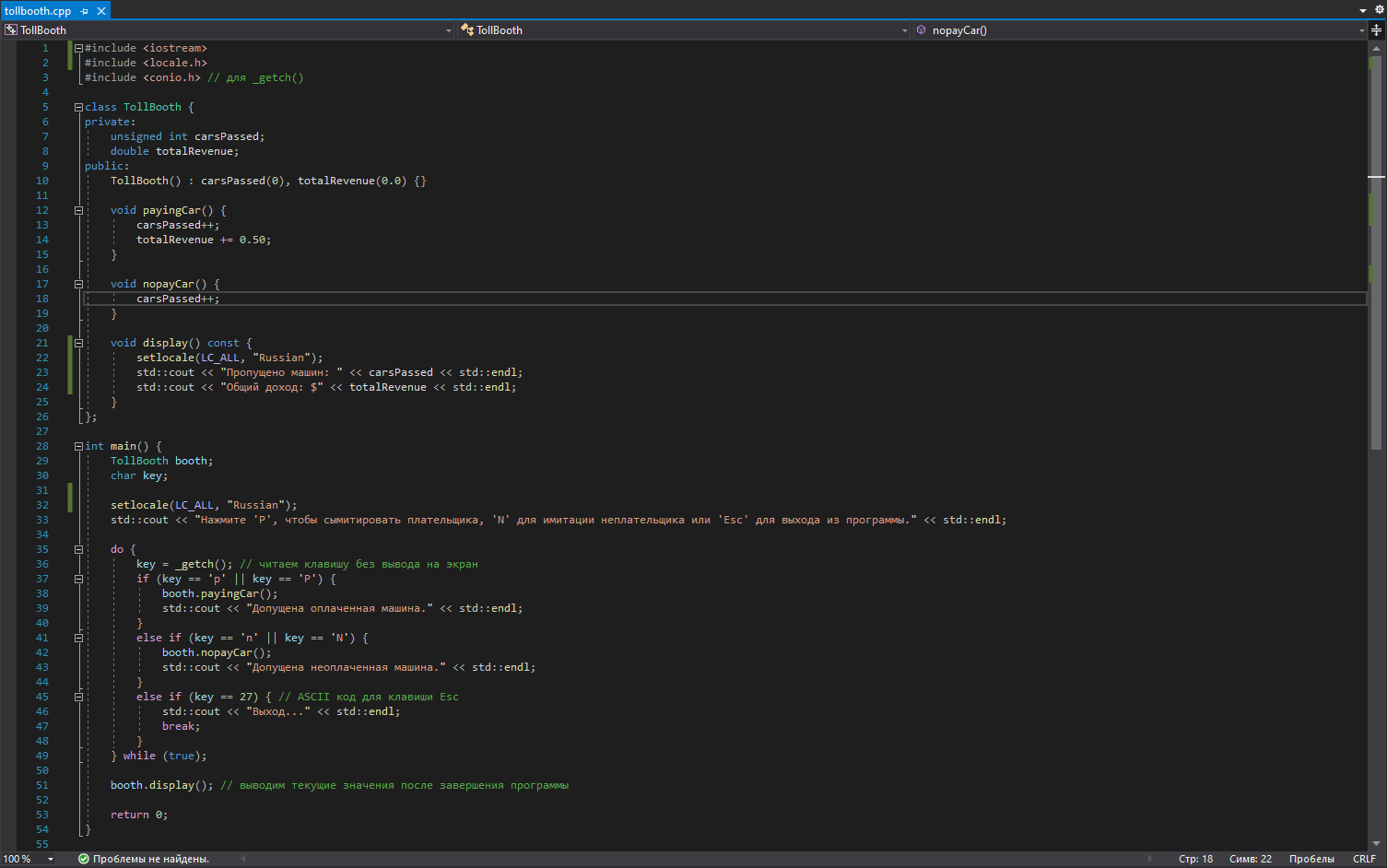
*Код к заданию 1*



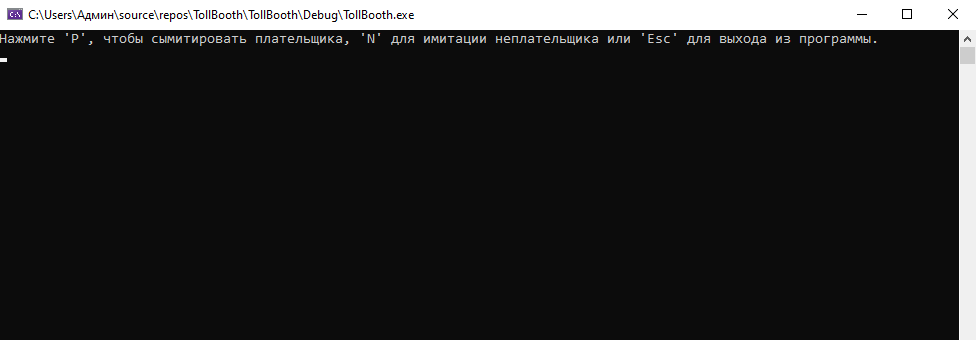
Результат выполнения:



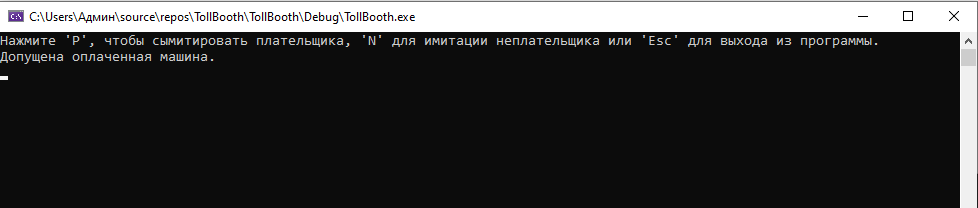
*Код к заданию 2*



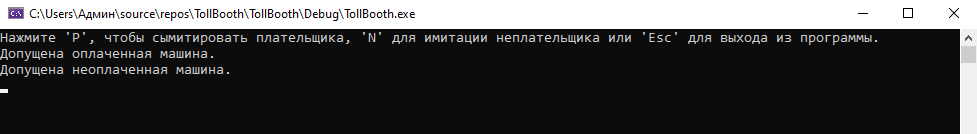
Результат выполнения:



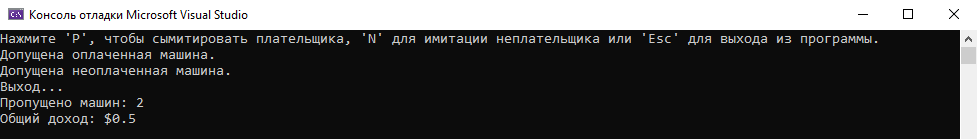
После нажатия P:



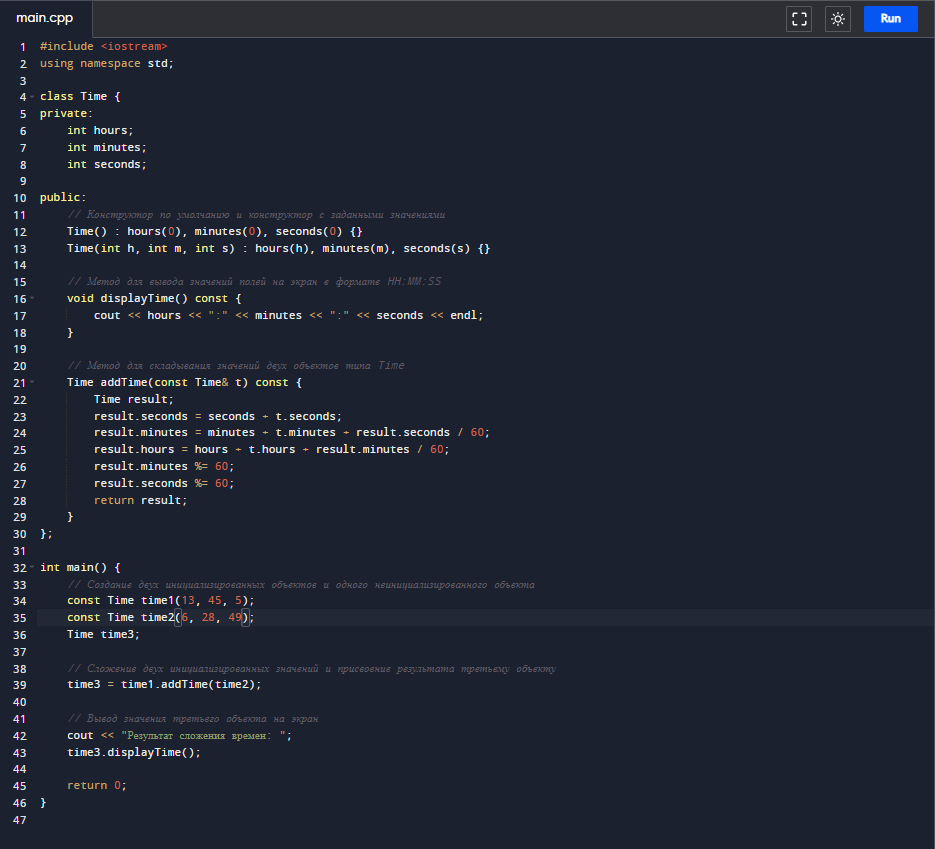
После нажатия N:



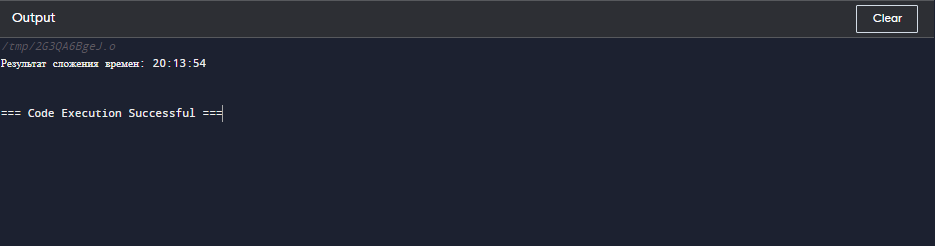
После нажатия Esc:



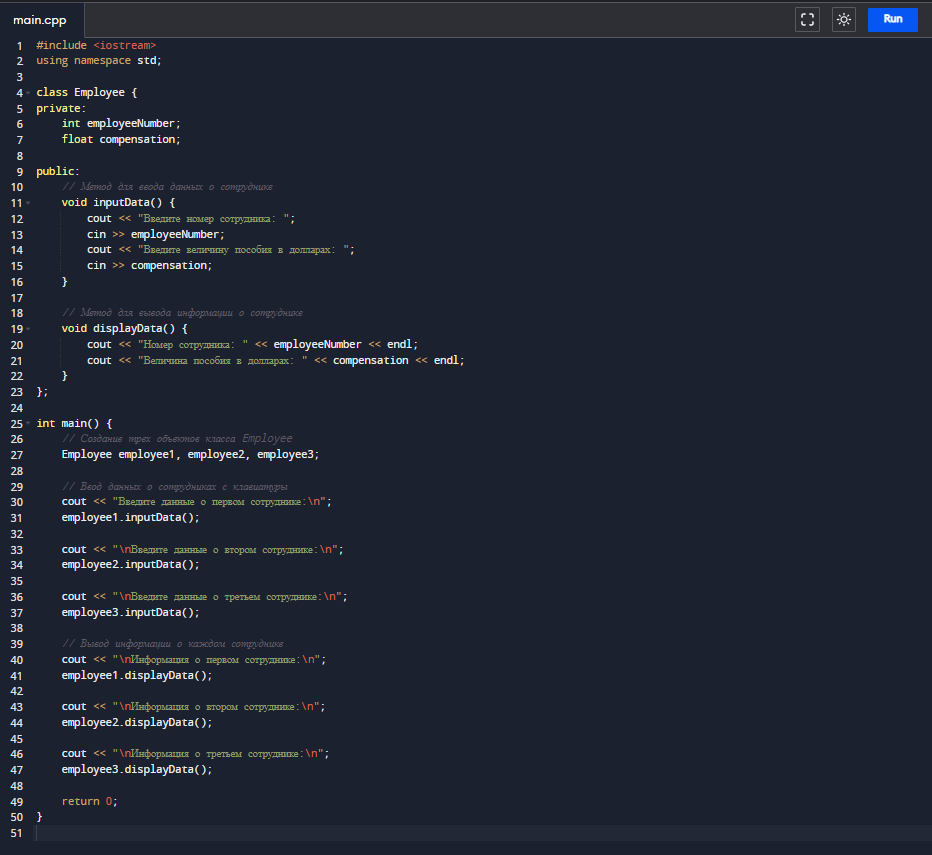
*Код к заданию 3*



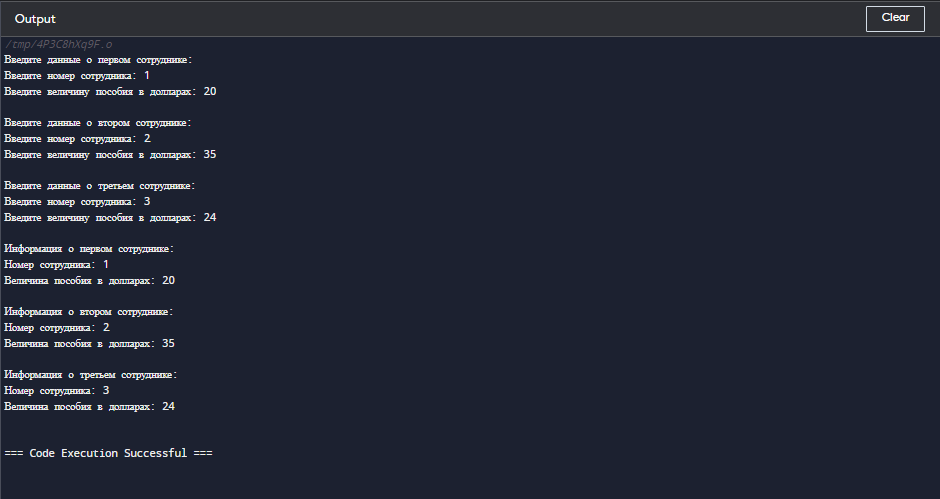
Результат:



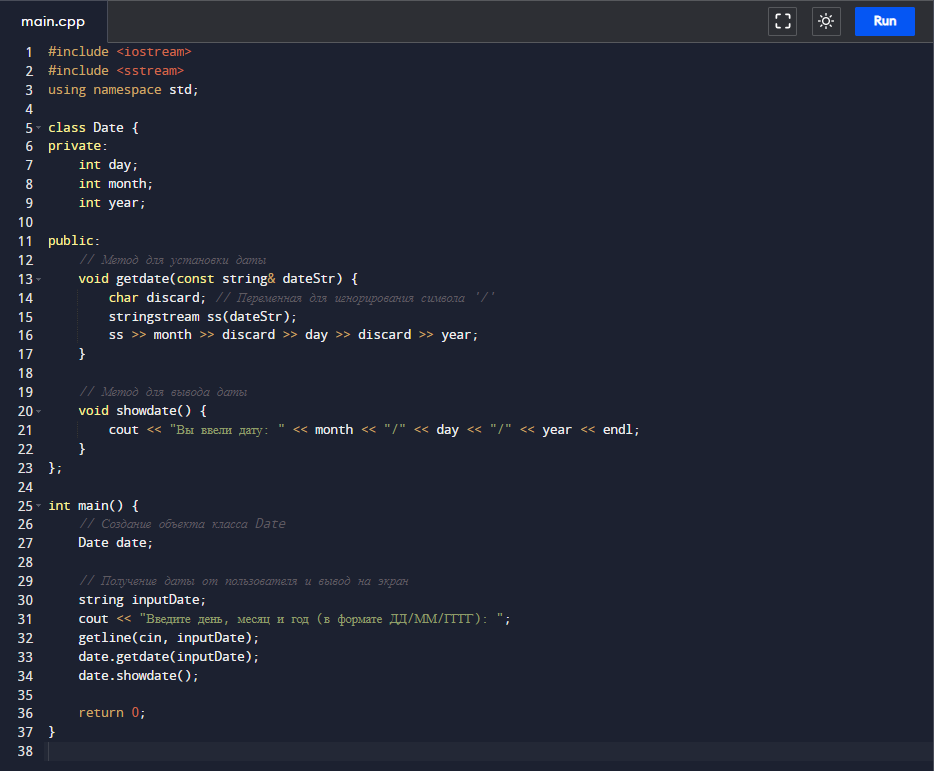
*Код к заданию 4*



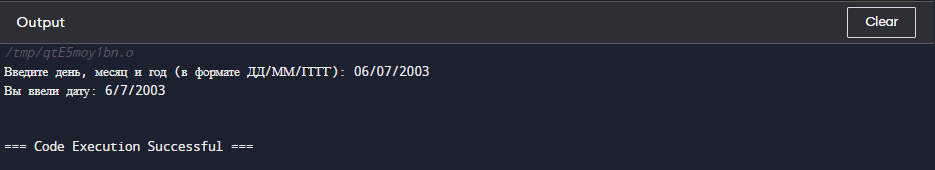
Результат после ввода всех данных:



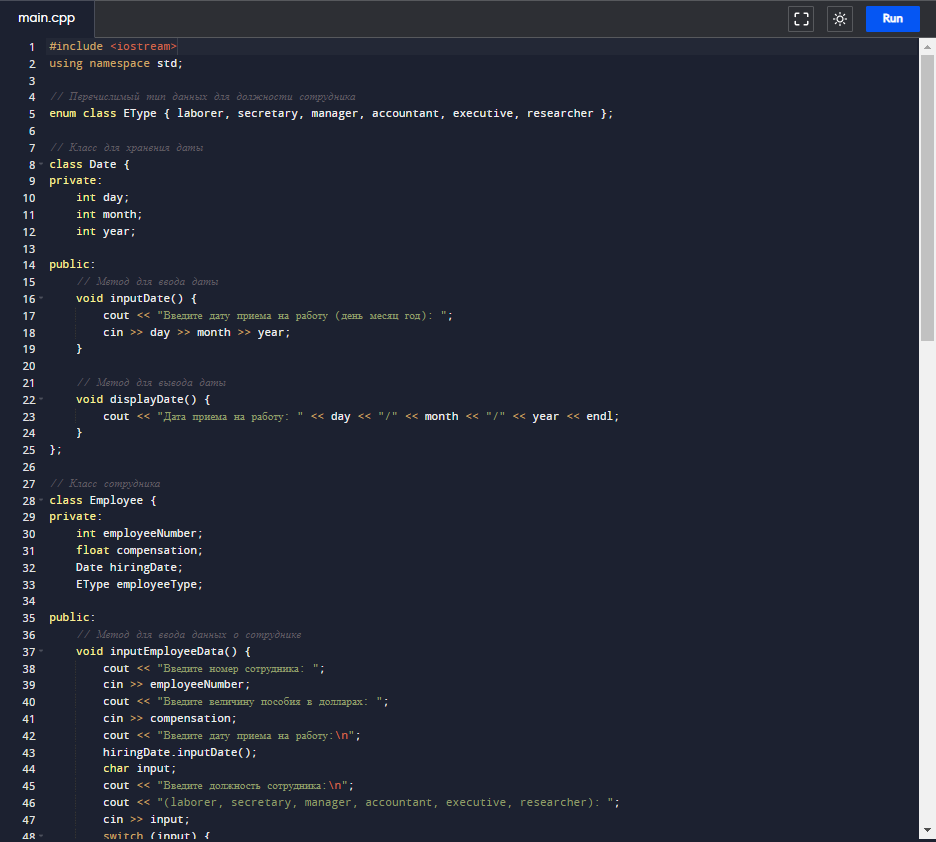
*Код к заданию 5*

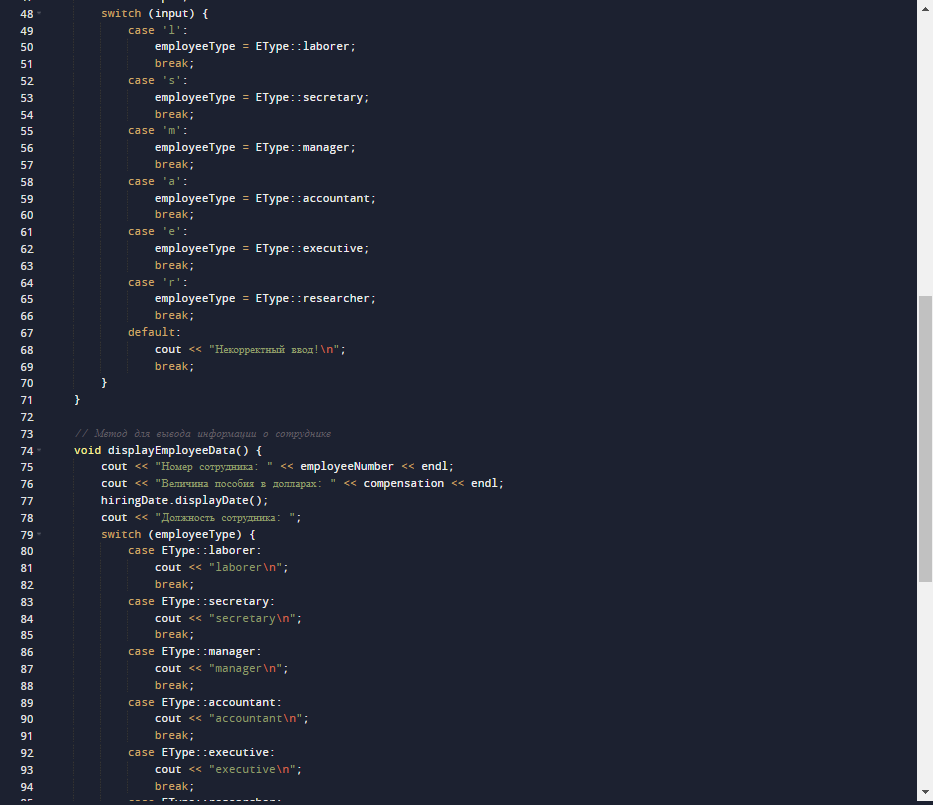


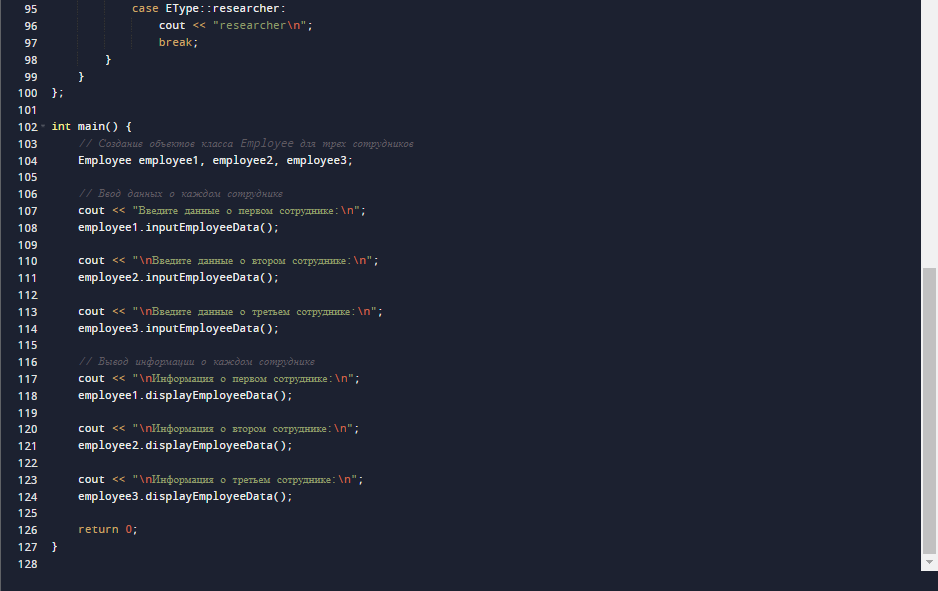
Результат:



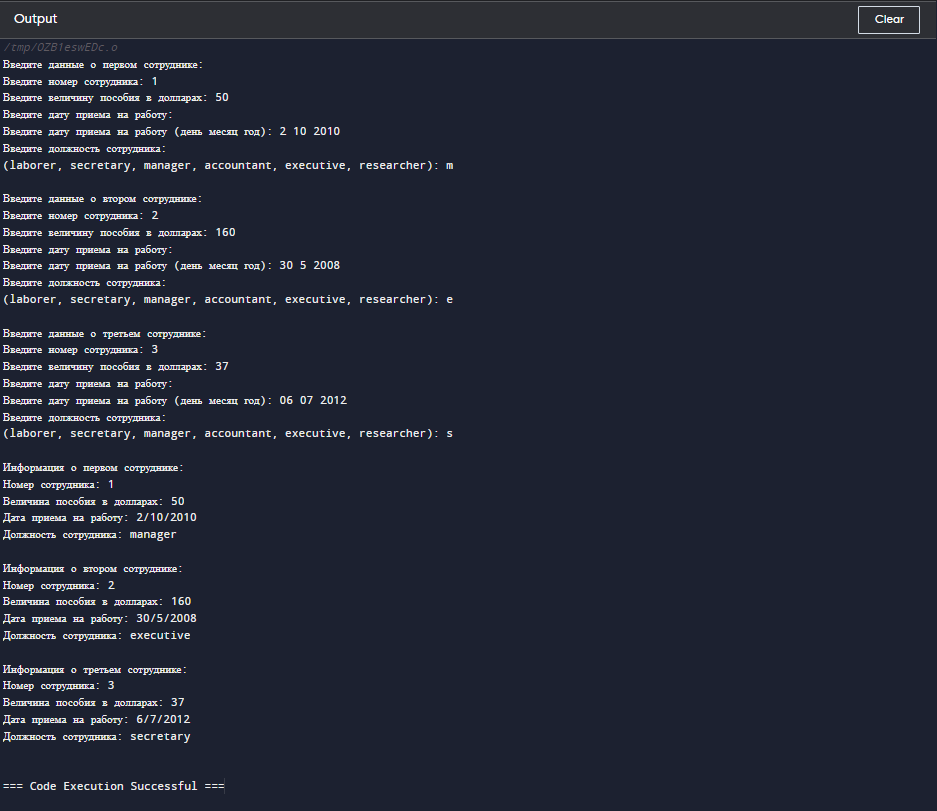
*Код к заданию 6*



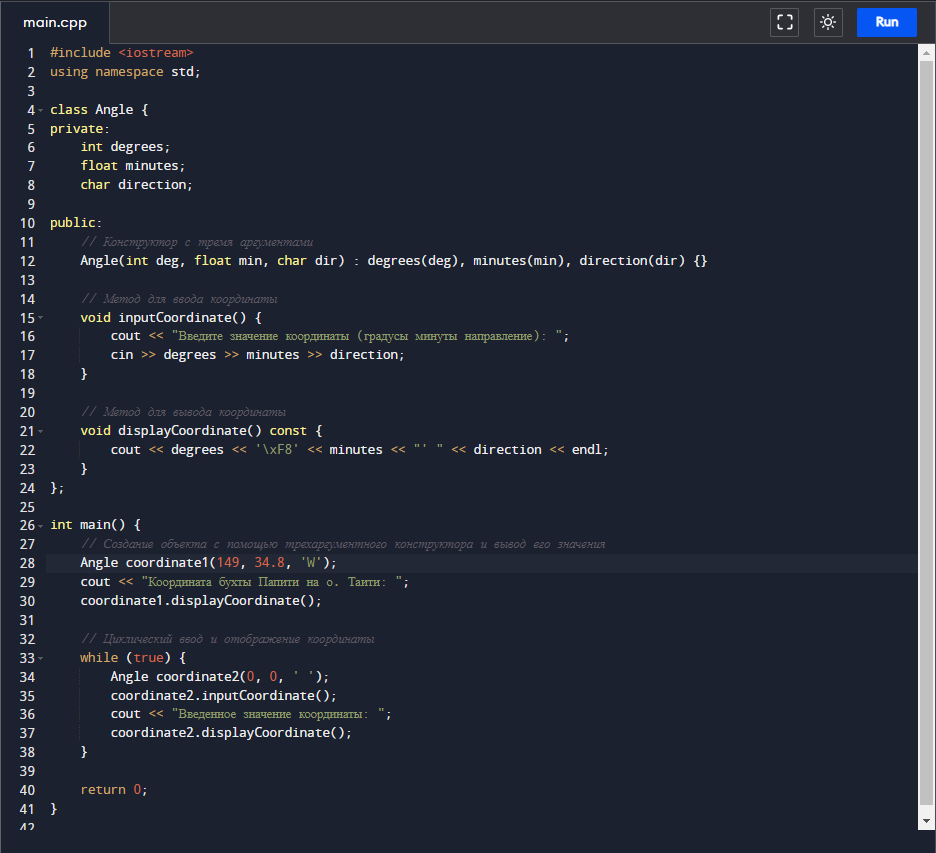




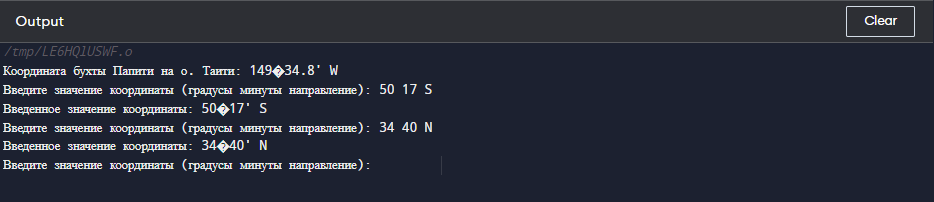
Результат:



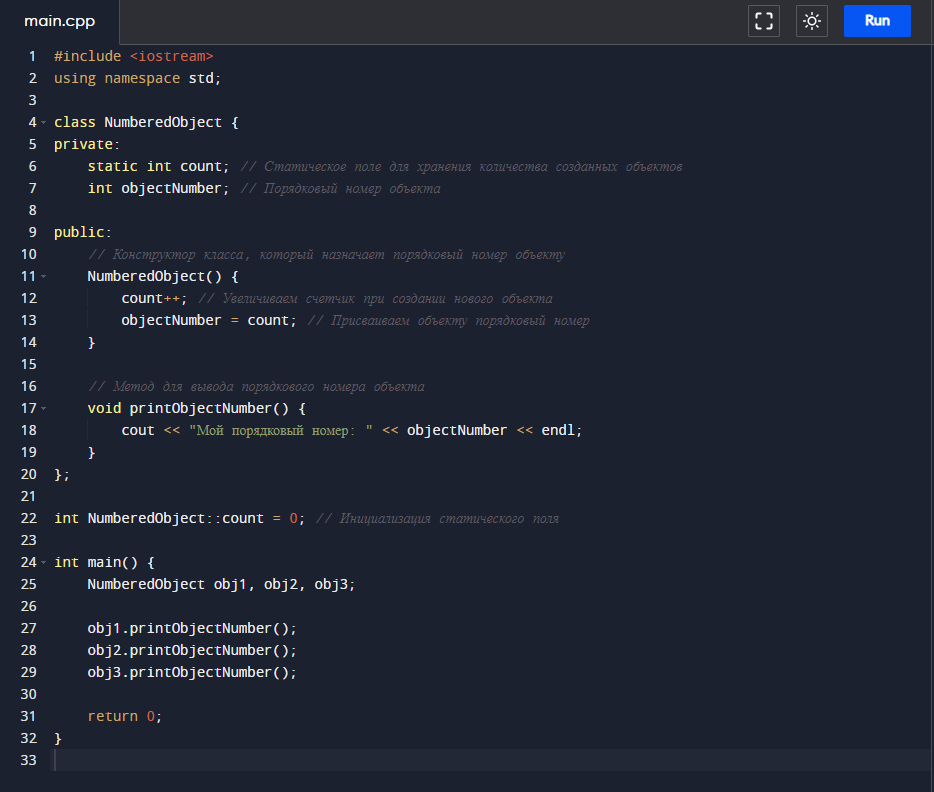
*Код к заданию 7*



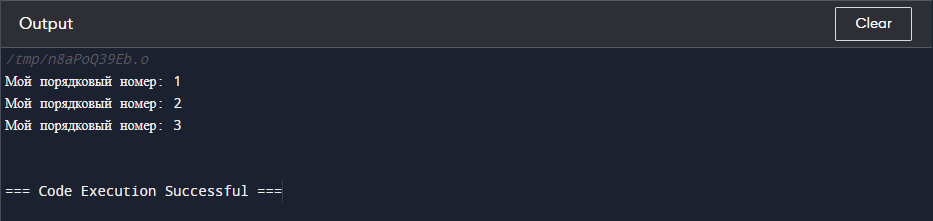
Результат:



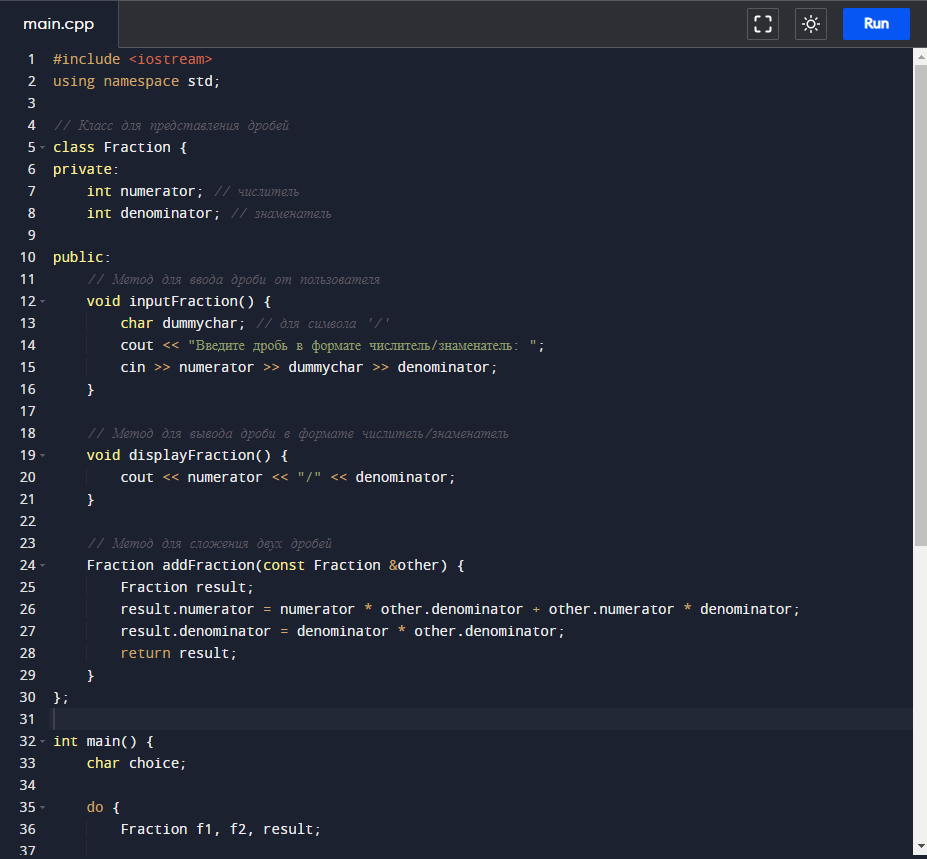
*Код к заданию 8*

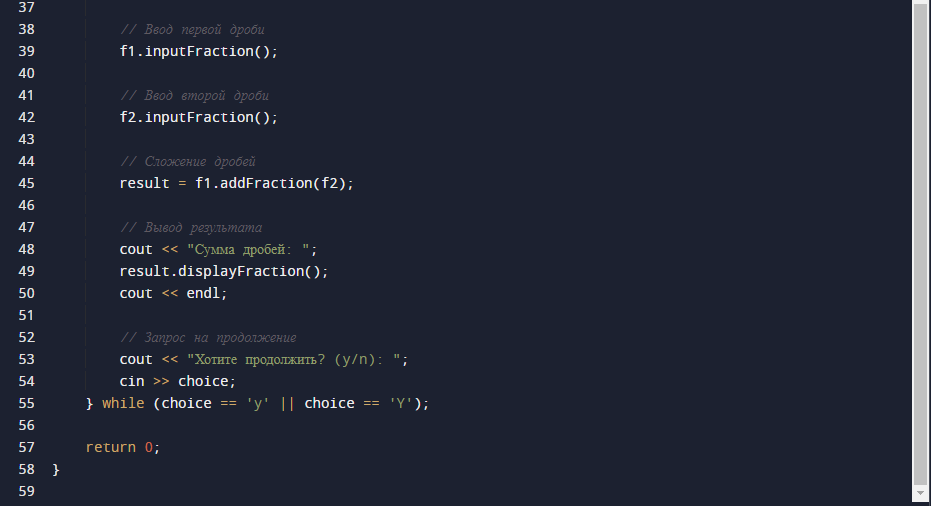


Результат:

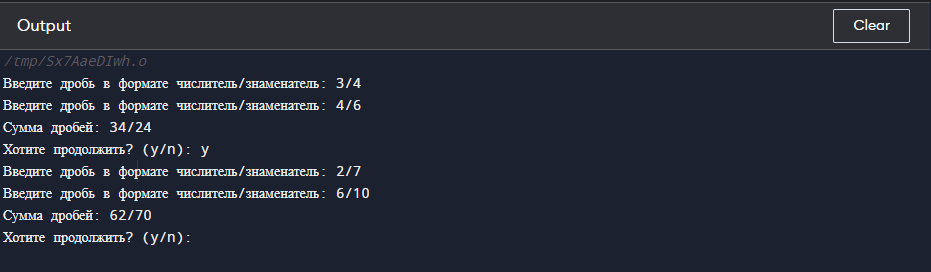


*Код к заданию 9*

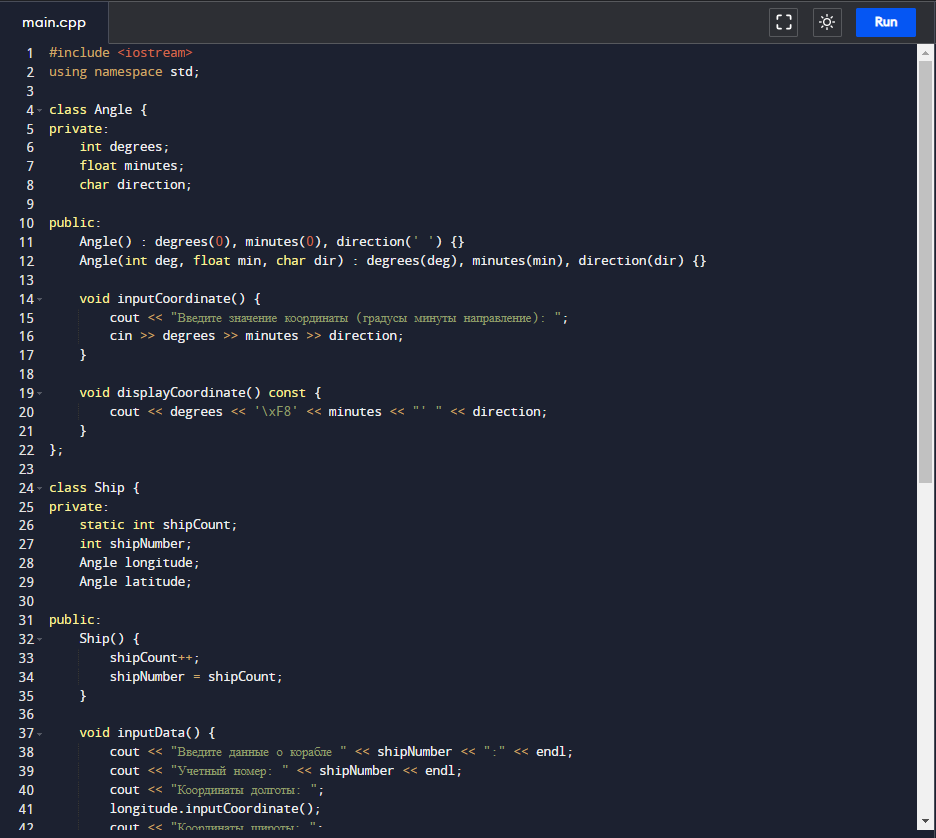


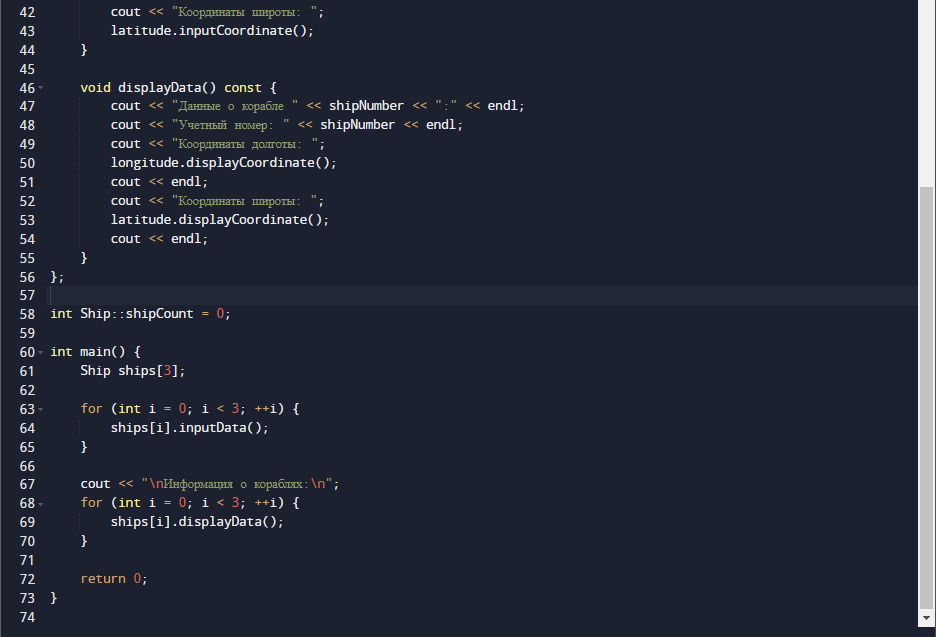


Результат:

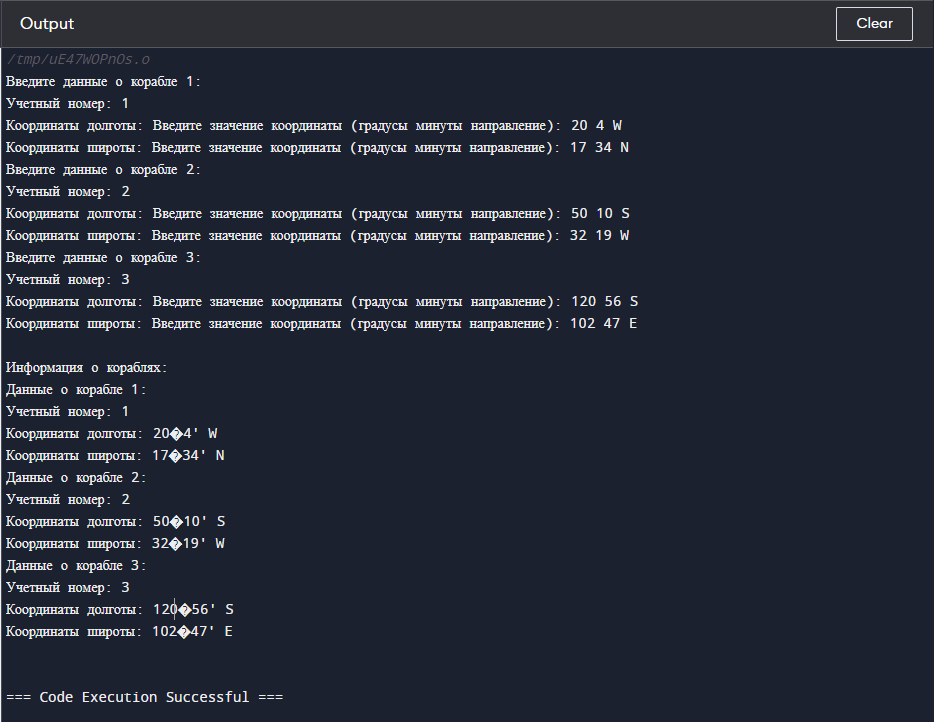


*Код к заданию 10*

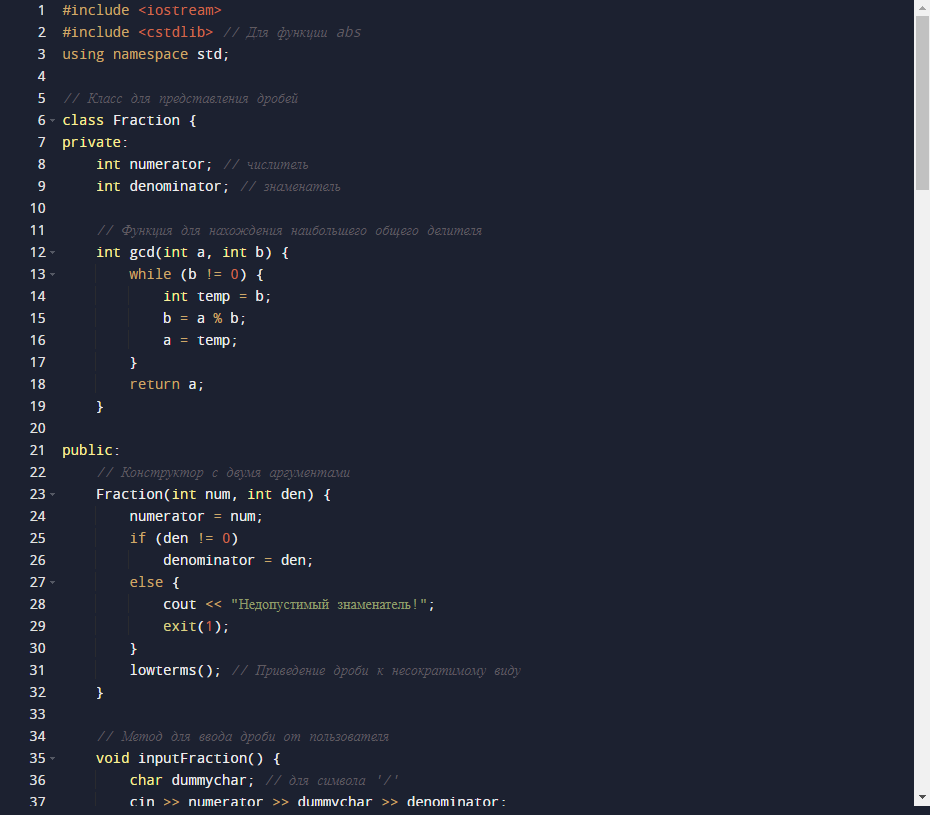


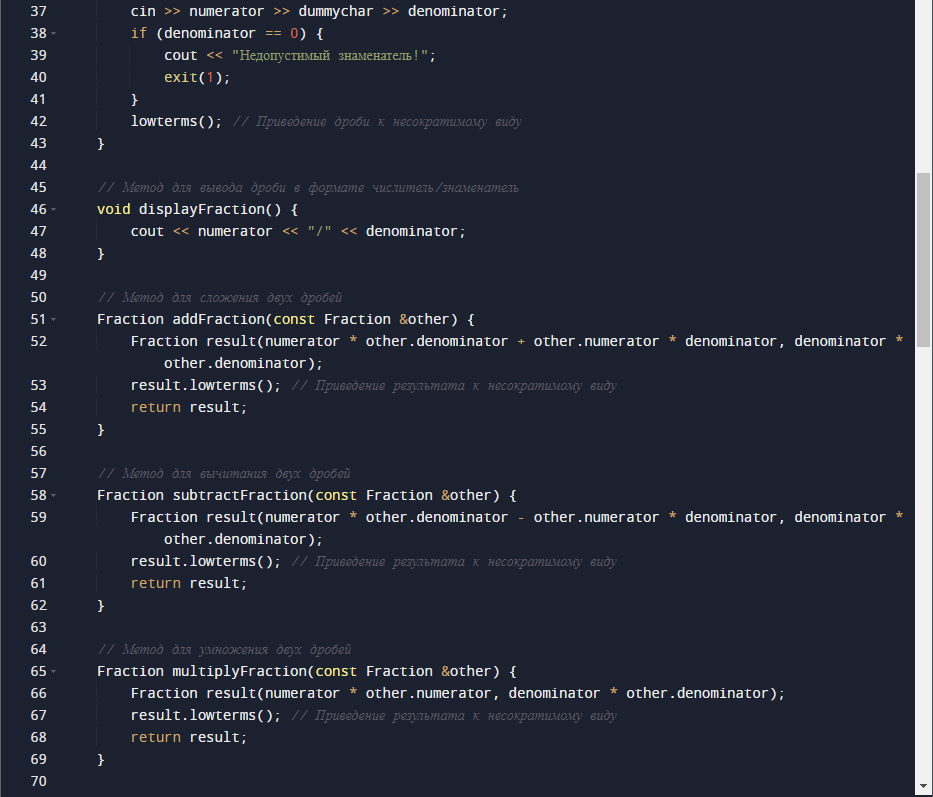


Результат:

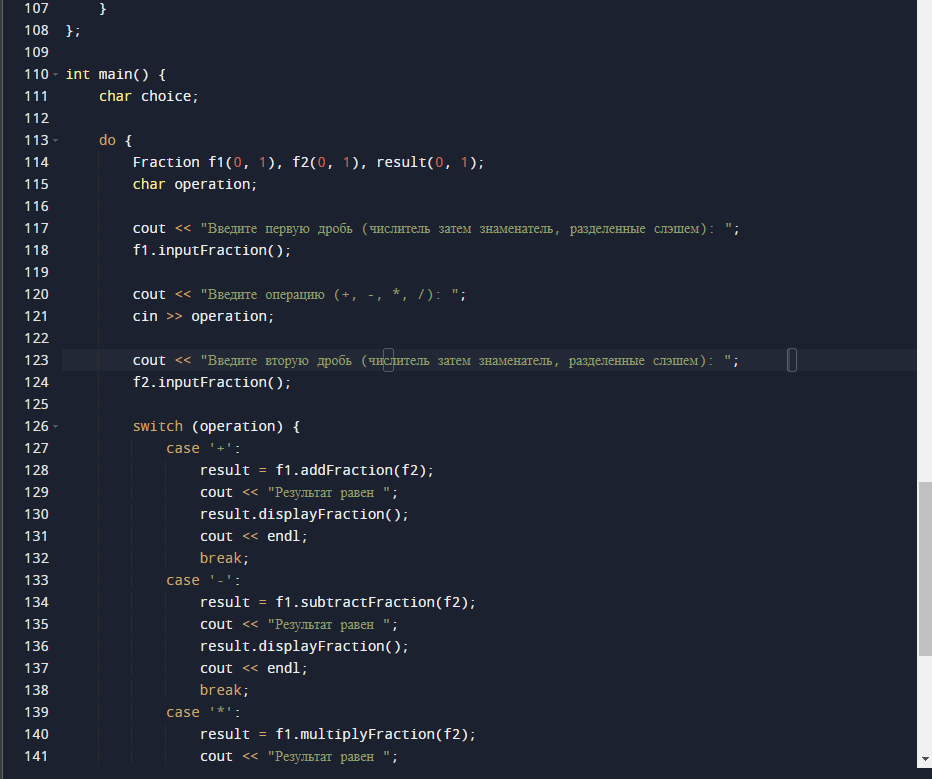


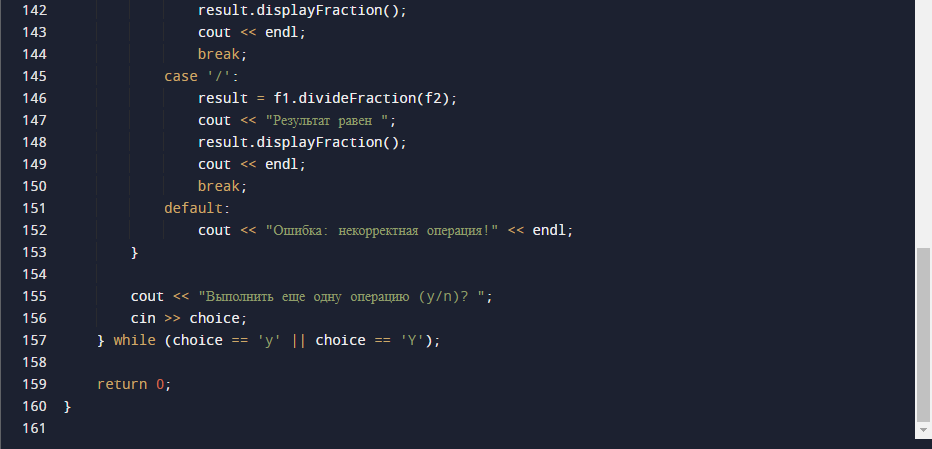
*Код к заданию 11*



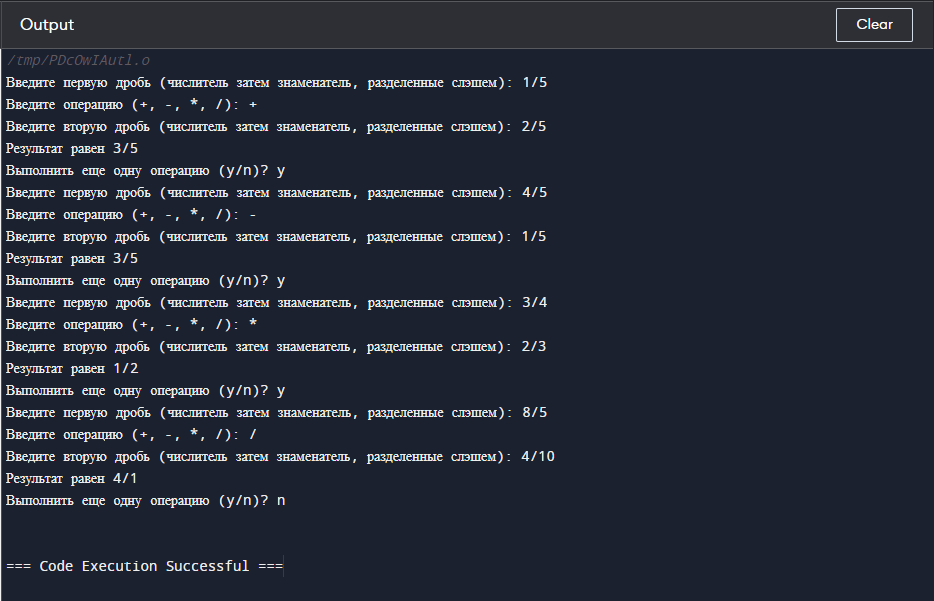




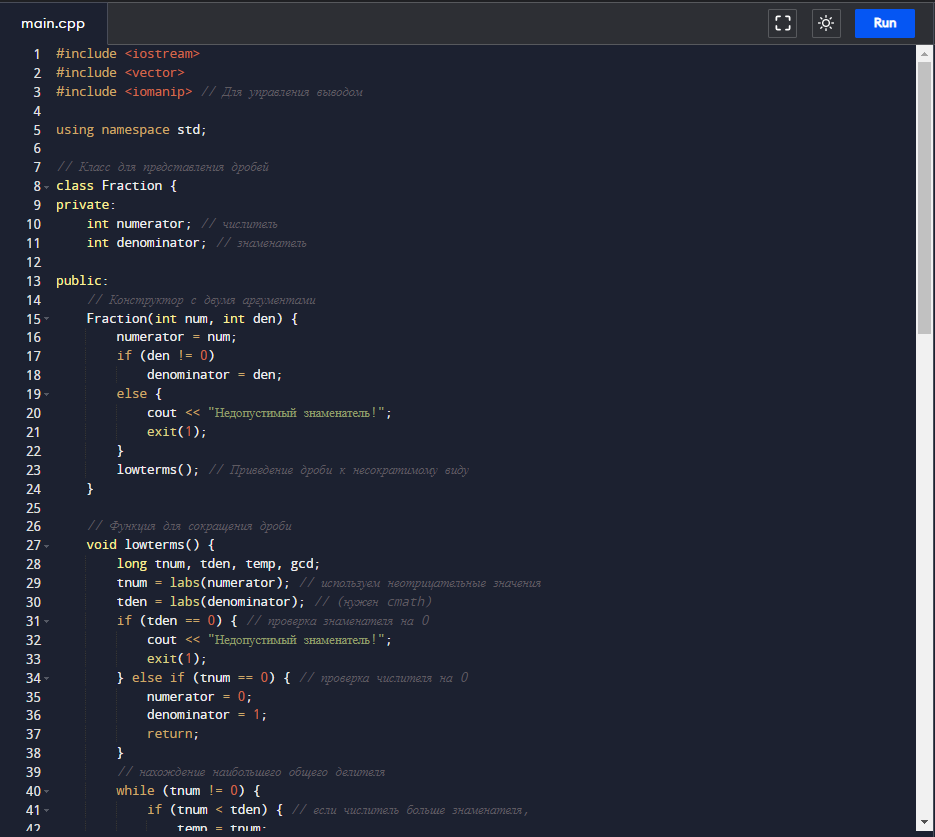


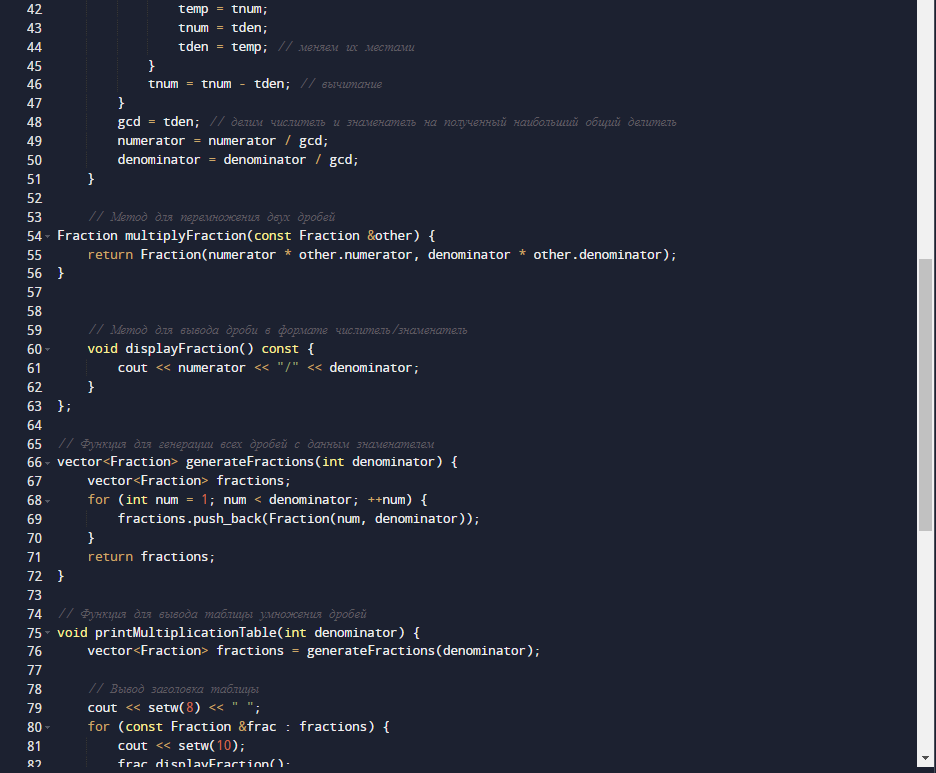


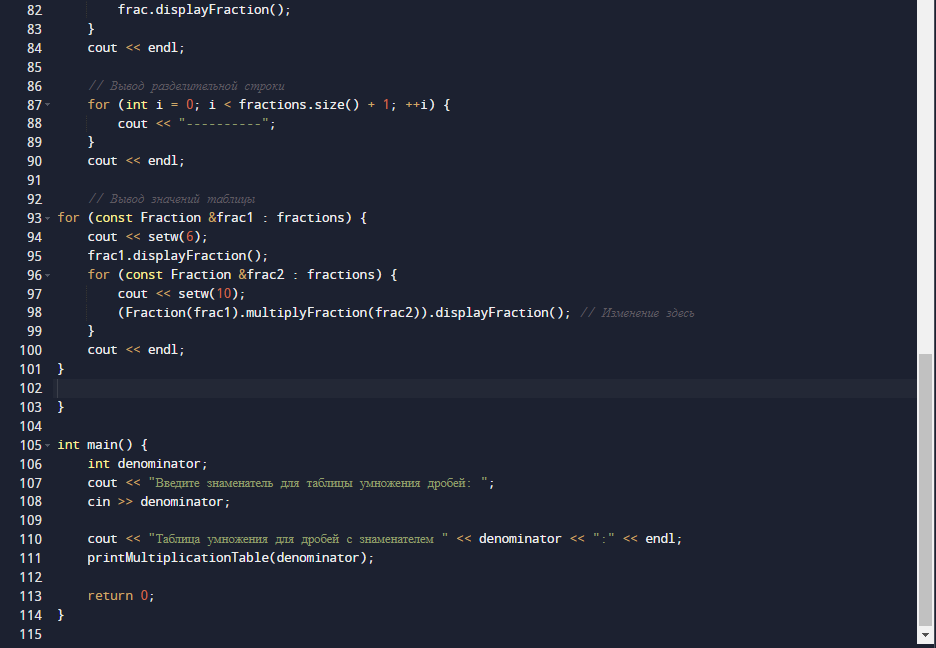
Результат:



*Код к заданию 12*







Результат:

