

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Пермский национальный исследовательский политехнический университет  
Электротехнический факультет  
Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**  
**о работе по информатике**  
Семестр: 2

На тему: «Классы и объекты. Инкапсуляция»

Выполнил студент ИВТ-22-26:  
Дияров Тимур Артурович

Проверил доцент кафедры ИТАС:  
Полякова Ольга Андреевна

Пермь 2023

## Постановка задачи

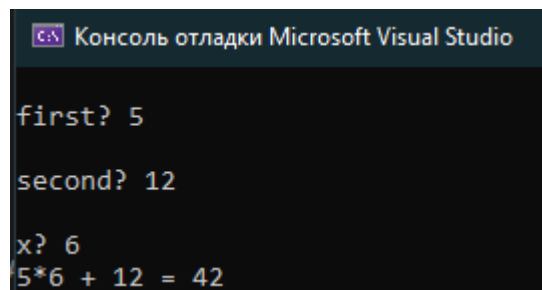
Реализовать вычисление переменной  $y$  для заданных переменных  $A$ ,  $B$ , и  $x$  в линейном уравнении  $y = Ax + B$ .

## Описание структуры

Структура “fraction” с двумя переменными “first” (типа double) и “second” (типа int). В структуре имеются три функции типа void:

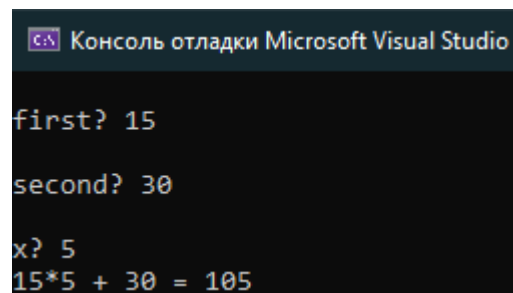
- Init(double, int): инициализация переменных с помощью двух чисел
- Read(): ручной ввод двух переменных
- Show(): вывод двух переменных в консоль

## Скриншоты работы программы



```
Консоль отладки Microsoft Visual Studio

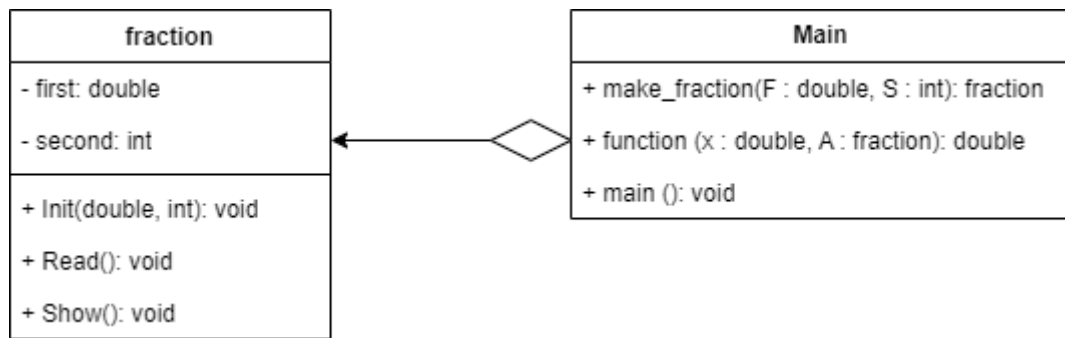
first? 5
second? 12
x? 6
5*6 + 12 = 42
```



```
Консоль отладки Microsoft Visual Studio

first? 15
second? 30
x? 5
15*5 + 30 = 105
```

## UML-диаграмма



## Контрольные вопросы

1. Что такое класс?  
Класс - абстракция, описывающая методы и свойства различных объектов.
2. Что такое объект (экземпляр) класса?  
Экземпляр класса - переменная, типом которой является сам класс.
3. Как называются поля класса?  
Поля класса называют атрибутами.
4. Как называются функции класса?  
Функции класса называют методами.
5. Для чего используются спецификаторы доступа?  
Спецификаторы доступа используются для настраивания доступа содержимого класса.
6. Для чего используется спецификатор `public`?  
Спецификатор `public` делает содержимое класса полностью доступным.
7. Для чего используется спецификатор `private`?  
Спецификатор `private` делает содержимое класса доступным лишь внутри него самого.
8. Если описание класса начинается со спецификатора `class`, то какой спецификатор доступа будет использоваться по умолчанию?  
По умолчанию используется спецификатор `internal`.
9. Если описание класса начинается со спецификатора `struct`, то какой спецификатор доступа будет использоваться по умолчанию?  
По умолчанию используется спецификатор `private`.
10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?  
Должен использоваться спецификатор `public`, ибо это позволяет другим частям программы использовать и вызывать методы класса, определенные в интерфейсе.

11. Каким образом можно изменить значения атрибутов экземпляра класса?

Значения атрибутов экземпляра класса можно изменить путем вызова методов класса, которые изменяют значения полей класса.

12. Каким образом можно получить значения атрибутов экземпляра класса?

Значения атрибутов экземпляра класса можно получить путем вызова методов класса, которые возвращают значения полей класса.

13. Класс описан следующим образом

```
struct Student
{
    string name;
    int group; .....
};
```

Объект класса определен следующим образом

```
Student *s=new Student;
```

Как можно обратиться к полю name объекта s?

Чтобы обратиться к полю name, нужно использовать оператор "->":

```
main()
{
    s->name;
}
```

14. Класс описан следующим образом

```
struct Student
{
    string name;
    int group; ..... };

```

Объект класса определен следующим образом

```
Student s;
```

Как можно обратиться к полю name объекта s?

Чтобы обратиться к полю name, нужно использовать оператор ".":

```
{
    s.name;
}
```

15. Класс описан следующим образом

```
class Student
```

```
{  
    string name;  
    int group; ..... };
```

Объект класса определен следующим образом

```
Student *s=new Student;
```

Как можно обратиться к полю name объекта s?

Чтобы обратиться к полю name, нужно использовать оператор "->":

```
main()
```

```
{  
    s->name;  
}
```

16. Класс описан следующим образом

```
class Student
```

```
{  
    string name;  
    int group;  
    public:  
    ..... };
```

Объект класса определен следующим образом

```
Student s;
```

Как можно обратиться к полю name объекта s?

Чтобы обратиться к полю name, нужно использовать оператор ".":

```
{  
    s.name;  
}
```

17. Класс описан следующим образом

```
class Student
{
public:
char* name;
int group; .....
};
```

Объект класса определен следующим образом

```
Student *s=new Student;
```

Как можно обратиться к полю name объекта s?

Чтобы обратиться к полю name, нужно использовать оператор "->":

```
main()
{
s->name;
}
```