Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.1**

Дисциплина: основы алгоритмизации и программирования

Тема: “Классы и объекты. Инкапсуляция”

Вариант 10

Выполнил:

студент группы ИВТ-20-2Б

Галинов О.Ю.

Проверила: доцент кафедры ИТАС

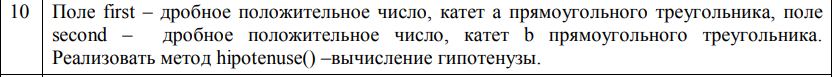
Полякова О.А.

Пермь, 2021

**Постановка задачи**

1. Реализовать определение нового класса. Продемонстрировать разные способы создания объектов и массивов объектов.

2. Структура-пара – структура с двумя полями, которые обычно имеют имена first и second.



**Анализ задачи**

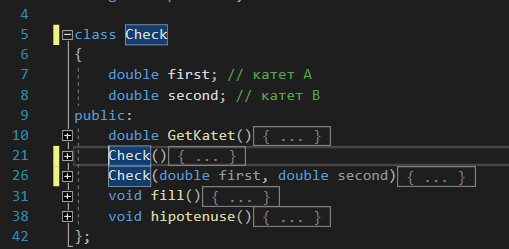
**1.** Для решения задачи необходимо:

**1.1.** Организовать класс Check для хранения значений первого и второго катета.

**1.2.** Организовать необходимые методы для ввода данных в поля first и second: гетторы, сетторы, конструкторы, а также метод для подсчета гипотенузы.

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Класс Check для хранения левой и правой границы отрезка через соответствующие поля.



**2.2.** Переменные типа Check: hipot, A – объекты класса Check.

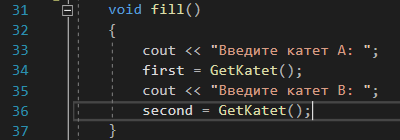
Check hipot;

Check A(6, 8);

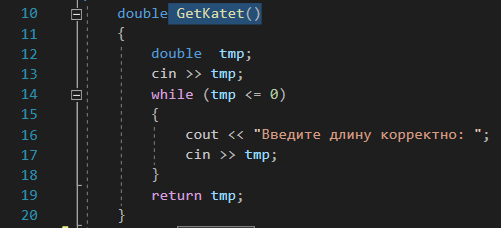
**2.3.** Переменные типа double: first, second, где first – первый катет, second – второй катет.

**3.** Для решения задачи данные были представлены в следующем виде:

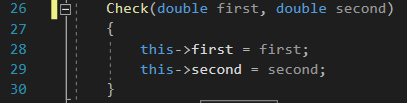
**3.1.** Данные вводятся через объекты класса Check через метод fill().



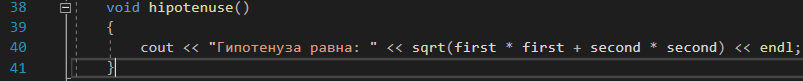
**3.2.** Проводится проверка на ввод при помощи метода GetKatet() .



**3.3.** Данные также вводятся через объекты класса Check через конструктор

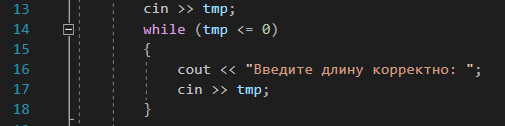


**3.4.** Полученные катеты передаются в метод hipotenuse() и рассчитывается гипотенуза



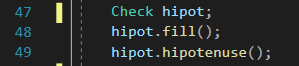
**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Для ввода правой и левой границы используются гетторы (методы, описанные в классе для ввода), в которых используется функция cin.



**5.** Поставленные задачи будут решены следующими действиями:

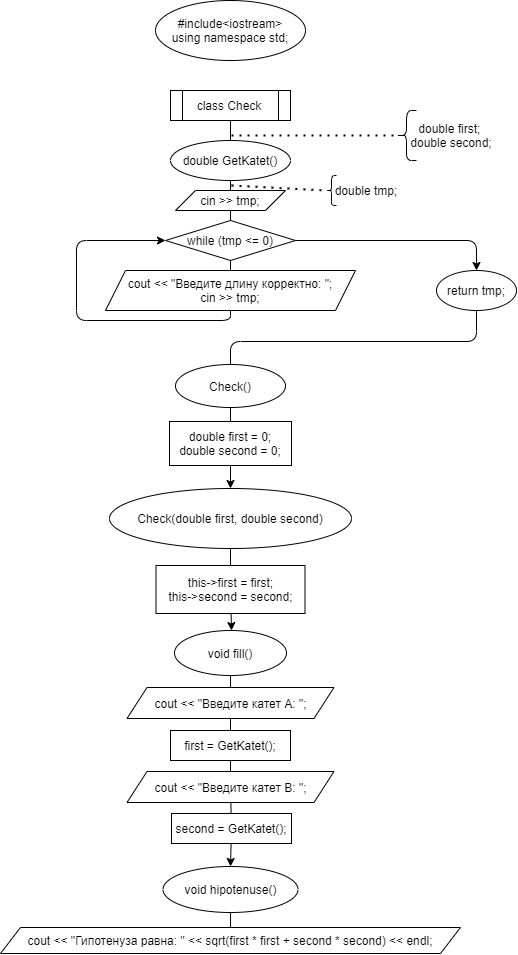
**5.1.** В функции main() вызываются метод ввода для катетов.

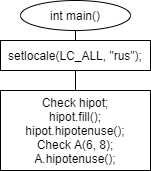


**5.2.** Можно так же ввести данные через конструктор, который получает на вход 2 числа типа double (6.0, 8.0).



**Блок-схема**





**Код**

#include <iostream>

using namespace std;

class Check

{

double first; // катет А

double second; // катет В

public:

double GetKatet()

{

double tmp;

cin >> tmp;

while (tmp <= 0)

{

cout << "Введите длину корректно: ";

cin >> tmp;

}

return tmp;

}

Check()

{

double first = 0;

double second = 0;

}

Check(double first, double second)

{

this->first = first;

this->second = second;

}

void fill()

{

cout << "Введите катет А: ";

first = GetKatet();

cout << "Введите катет В: ";

second = GetKatet();

}

void hipotenuse()

{

cout << "Гипотенуза равна: " << sqrt(first \* first + second \* second) << endl;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

Check hipot;

hipot.fill();

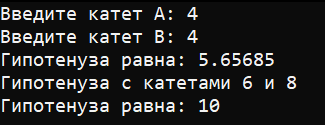
hipot.hipotenuse();

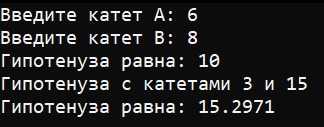
Check A(6, 8);

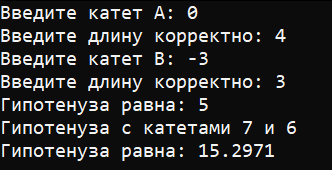
A.hipotenuse();

}

**Скриншоты**

****

****

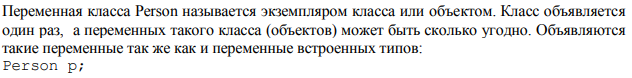
****

**Контрольные вопросы**



1. 



2.



3. Полями/атрибутами называются данные класса.



4. Функции класса называются методами.



5. Спецификаторы доступа управляют видимостью элементов класса.



6. Спецификатор public используется, чтобы задать общедоступную видимость.



7. Спецификатор private используется, чтобы задать видимость только внутри класса для полей или методов под этим спецификатором.



8. Для спецификатора class по умолчанию используется private.



9. Для спецификатора struct по умолчанию используется public.



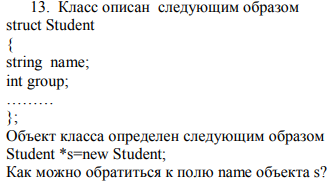
10. Интерфейс класса описывается после спецификатора public, чтобы к методам класса можно было обратится из других функций.



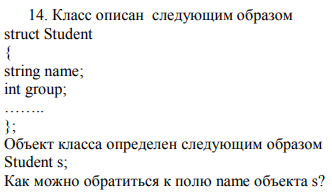
11. Значения атрибутов экземпляра класса можно изменить с помощью методов класса: модификаторов.



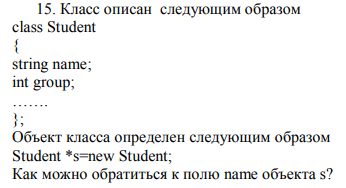
12. Значения атрибутов экземпляра класса можно получить с помощью методов класса: селекторов.



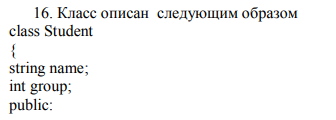
13. Через «->» можно обратиться к полю name, так как оно имеет открытый доступ public.

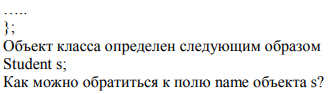


14. Через «.» можно обратиться к полю name, так как оно имеет открытый доступ public.

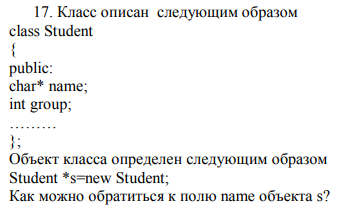


15. Необходимо прописать модификатор для класса Student.





16. Необходимо прописать модификатор для класса Student.



17. Необходимо прописать селектор и модификатор для класса Student.