Задание по программированию: Набор фигур

Условие

Вам дана функция main, которая считывает из стандартного ввода команды по работе с набором геометрических фигур:

int main() {

  vector<shared\_ptr<Figure>> figures;

  for (string line; getline(cin, line); ) {

    istringstream is(line);

    string command;

    is >> command;

    if (command == "ADD") {

      // Пропускаем "лишние" ведущие пробелы.

      // Подробнее об std::ws можно узнать здесь:

      // https://en.cppreference.com/w/cpp/io/manip/ws

      is >> ws;

      figures.push\_back(CreateFigure(is));

    } else if (command == "PRINT") {

      for (const auto& current\_figure : figures) {

        cout << fixed << setprecision(3)

             << current\_figure->Name() << " "

             << current\_figure->Perimeter() << " "

             << current\_figure->Area() << endl;

      }

    }

  }

  return 0;

}

Как видно из кода, есть два вида команд:

* **ADD** — добавить фигуру в набор;
* **PRINT** — для каждой фигуры в наборе распечатать название, периметр и площадь.

Программа поддерживает три вида фигур: прямоугольник, треугольник и круг. Их добавление описывается так:

1. **ADD RECT *width height*** — добавить прямоугольник с размерами ***width*** и ***height*** (например, **ADD RECT 2 3**).
2. **ADD TRIANGLE *a b c*** — добавить треугольник со сторонами ***a***, ***b*** и ***c*** (например, **ADD TRIANGLE 3 4 5**).
3. **ADD CIRCLE *r*** — добавить круг радиуса ***r*** (например, **ADD CIRCLE 5**).

Пример работы программы

Ввод

ADD RECT 2 3

ADD TRIANGLE 3 4 5

ADD RECT 10 20

ADD CIRCLE 5

PRINT

Вывод

RECT 10.000 6.000

TRIANGLE 12.000 6.000

RECT 60.000 200.000

CIRCLE 31.400 78.500

**Не меняя функцию main**, реализуйте недостающие функции и классы:

* базовый класс Figure с чисто виртуальными методами Name, Perimeter и Area;
* классы Triangle, Rect и Circle, которые являются наследниками класса Figure и переопределяют его виртуальные методы;
* функцию CreateFigure, которая в зависимости от содержимого входного потока создаёт shared\_ptr<Rect>, shared\_ptr<Triangle> или shared\_ptr<Circle>.

Гарантируется, что размеры всех фигур — это натуральные числа не больше 1000. В качестве значения PI используйте 3,14.

Как будет тестироваться ваш код

В рамках тестирования вашего кода будет два этапа.

На первом этапе будет проверена корректность работы функции CreateFigure: функция возвращает корректный объект, созданный с помощью make\_shared<Rect>; если во входной поток функции передана команда на создание прямоугольника (RECT width height); корректный объект, созданный с помощью make\_shared<Circle>, если во входной поток функции передана команда на создание круга (CIRCLE r), и т. д. При этом гарантируется, что все команды на создание объектов во входном наборе тестов имеют корректный формат без дополнительных пробелов в начале и конце команды.

На втором этапе автоматическая проверяющая система заменит в вашей программе функцию main на main из условия задачи, скомпилирует получившийся файл и прогонит на наборе тестов, проверив таким образом корректность вашего решения на исходной функции main. При этом гарантируется, что все команды ADD в тестах также корректны.