A. Targets for laboratory work 1 semester

Alexey Martynov 27

February 2018

version 1.5

Additional tasks are performed by those who have to date not a job, or is known to be incomplete or incorrect, and the work nekompiliruyuschuyusya [1].

1 Geometric Shapes

All numerical data in this paper should be submitted floating point values.

1. Create a base-types.hpp file containing the definitions of the following structures:

   - point_t, is a point on the plane, the coordinates must be stored in the fields x and y.
   - rectangle_t, The frame width width and height height centered at pos.

2. Create a file shape.hpp, containing the definition of the abstract class Shape. This class must provides the following methods:

   getArea area calculation

   getFrameRect getting the bounding box for the shape (see. the types of predydusche- first paragraph), the sides of the bounding box is always parallel to the axes

   move moving the center of the figure 2 embodiment: a specific point in a displacement of the axes the x and the y

3. Implement classes Rectangle and Circle in rectangle.hpp files, rectangle.cpp, circle.hpp and circle.cpp respectively.

4. Demonstrate correct operation of classes simple program. The demonstration should included the use of chat polymorphic classes.

additional task

Add support triangles. As the center of the triangle and is followed by The use Vat center of gravity, as its calculation of the most simple.

Scaling 2 figures

1. *As an exception! All further work should follow the rules of registration of works and contain the copied code, except for the test suite.* Copy source za- Denmark 1.

2. Drag the shape classes in a separate namespace. Name this space should be chosen to coincide with the student's name in lowercase (accordingly, it coincides with a part of the directory name with the works up to a point), for example, Ivan Petrov directory will petrov.ivan vatsya-called, respectively, the name of the namespace - petrov. This namespace must be maintained for all the remaining works in this semester.

3. Add in the figure class method scale (), performing an isotropic scaling figures relative to its center with a specified coefficient.

   If the first operation was carried triangle, scaling must be implemented, and for him.

4. Write tests that check:

   - invariability width and height, and area of the figure while moving;
   - squared change area of the figure when zooming;
   - availability and processing of incorrect parameters in functions;

To write tests you need to create test-main.cpp file in which to implement the tests. Launched earlier demonstration program should be developed to demonstrate the new features.

[1] If being late is evident from the very beginning, it is possible to present the work at once with additional tasks.

Compound 3 figures

Extend implementation work 2 by adding the class CompositeShape, storing a list of of arbitrary shapes within an array. In this job *not allowed* use of the standard of the containers, it is your responsibility to implement the storage set of figures on the basis of a dynamic array. The class must be placed in the files composite-shape.hpp and composite-shape.cpp.

Write a test suite that verifies the correct operation of the generated class. It must be remembered that the implementation may require more specialized techniques in the classroom to ensure correct operation.

For CompositeShape zooming and moving the work from the center of the object for which the receiving center of the bounding box.

Launched earlier demonstration program should be developed to demonstrate the new features.

4 Processing of figures

To expand the implementation of the work 3:

1. Add the figures support the rotation by a predetermined angle in degrees, positive direction - counterclockwise rotation center coincides with the center of the **figure. It must be remembered that the bounding rectangle retains the parallel sides of the coordinate axes and rotation CompositeShape**

   also comprises displacement elements. The method should be called rotate ().

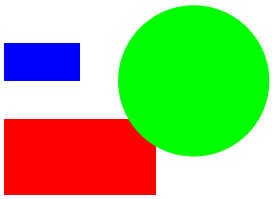
   To those who implemented the first work supporting triangles, it is necessary to maintain the rotation of triangles too.

2. **Write a list of the partition pieces (Rectangle, Circle, CompositeShape and if has been implemented,** Triangle), **presented as a single figure, the layers in the order of adding:**

   - shapes that do not overlap limiting pryamogolnikami are on the same layer;
   - figure overlapped on the other, located on the next layer if it contains a list of the following figures, which it covers;
   - compound shapes added to the figure, subjected to decomposition, should be considered as a whole and to handle its bounding rectangle, rather than its components. For example, figures are given in the following order:

     - red box
     - green circle
     - blue rectangle

   Accordingly, they are drawn to each other:

   Splitting the layers is as follows:

   first layer 2 rectangles, as they do not overlap, and the circle closes direct red rectangle

   second layer green circle, because it is set after the red rectangle and overlap with them.

   Results partition represented in a matrix, where the rows represent layers in koto- ryh specified shape. Matrix implemented independently on the basis of dynamic memory, the data must be stored in block 1 (not allowed to create an array of pointers to the layers). The number of rows corresponds to the number of layers, the number of columns - the maximum number of pieces in a single layer.

3. Implement tests that demonstrate the correct operation of the partition function. Launched earlier demonstration program should be developed to demonstrate the new features.

additional task

Carried out by those who did not submit in time at least one job.

Add support polygon as a set of points on the coordinate plane. Preparation center of said figure should work correctly, the polygon should be tested for non-zero area and convexity and inappropriate polygons must be rejected.