Министерство науки и образования РФ

Федеральное государственное автономное образовательное

учреждение высшего профессионального образования

«Санкт-Петербургский государственный электротехнический

университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

(СПбГЭТУ «ЛЭТИ»)

Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

**Отчёт по заданию № 1**

**на тему: “Работа с иерархией объектов: наследование и полиморфизм”**

**по дисциплине**

**“Алгоритмы и структуры данных”**

**Вариант 26**

|  |  |
| --- | --- |
| Выполнил студент гр.9308: | Яловега Н.В. |
| Проверил: | Колинько П.Г. |

Санкт-Петербург, 2020 г.

Оглавление

[Введение 4](#_Toc65366441)

[1. Задание 4](#_Toc65366442)

[2. Формализация задания 4](#_Toc65366443)

[3. Контрольные примеры 5](#_Toc65366449)

[4. Работа с наследованием объектов 5](#_Toc65366451)

[5. Результаты эксперимента 6](#_Toc65366452)

[Вывод 8](#_Toc65366455)

[Список используемых источников 9](#_Toc65366456)

[Приложение 1 (Исходный текст программы) 10](#_Toc65366457)

## Введение

Целью задания является ознакомление с принципами ООП, в частности, с принципами наследования и полиморфизма, использование библиотеки «из пакета» и последующего расширения его функционала.

## 1. Задание

Доработать модуль shape.cpp, добавив в коллекцию еще одну фигуру — трапецию с крестом. Для этой фигуры нужно определить подходящее место в иерархии классови написать недостающие функции-члены.

Разработанной фигурой нужно дополнить картинку в указанных

в варианте позициях. Необходимо написать функции, обеспечивающие примыкание очередной фигуры p с ужной стороны по отношению к уже размещенной q.

## 2. Формализация задания

## Имеется схема объекта с пометками возможных точек присоединения (рис.1), в нашему случае такими точками будут: 12, 2, 3

## 

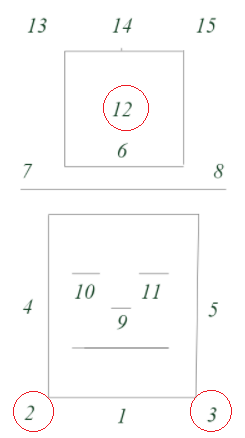
## *Рис.1. Возможные точки стыковки*

## Фигурой для присоединения будет использоваться окружность – примитивная фигура, свойства отражаемости и поворота не имею смысла из-за симметрии фигуры. Поэтому будем использовать производный класс circle от класса-родителя shape, не используя классы rotatable и reflectable.

## Также для присоединения нашей фигуры к точке 12 потребуется функция центровки двух фигур относительно друг друга, а для точек 2 и 3 функции присоединения по северо-восточной – юго-западной и северо-западной – юго-восточной точкам соответственно.

## 3. Контрольные примеры

В качестве контрольного примера выберем ожидаемый итоговый рисунок, не учитывая конкретные размеры окружности. Добавленные атрибуты выделены красным цветом на Рис. 2: эмблема на шляпе(т.12), бакенбарды(т.2 и т.3):



## *Рис.2. Ожидаемый результат(контрольный пример)*

## 4. Работа с наследованием объектов

**4.1. Добавленные классы**

Для реализации примитивной фигуры окружности был создан класс circle, являющийся производным от класса shape – абстрактного представления фигур. Наследование от классов rotatable и reflectableявлялось бы избыточным т.к. из-за симметричности окружности смысла в повороте или отражении не имеется.

**4.2. Переопределение функций-членов у добавленного класса**

По причине того, что класс-родитель shape является первым и единственным в иерархии наследования для производного класса circle – нам следует переопределить ВСЕ изначально заданные виртуальные(абстрактные) функции-члены в классе shape, т.к. ранее они не могли быть переопределены промежуточными классами.

**4.3. Реализация недоступности необязательных функций-членов класса**

Было решено сделать недоступными четыре функции-члена: конструктор копирования, конструктор перемещения, оператор присваивания копированием и оператор присваивания перемещением.

Способом реализации было выбрано изменение модификаторов доступа функций-членов на приватный. Таким образом, при попытке воспользоваться этими функциями возникнет ошибка на этапе компиляции программы.

## 5. Результаты эксперимента

Для начала убедимся в корректности создания и отрисовки добавленной фигуры :

*Рис.3. Отрисовка исходных фигур*

Сформируем бакенбарды из двух наибольших окружностей путём их уменьшения в 4 и 5 раз соответственно:

## *Рис.4. Преобразование фигур*

Уже подготовленные фигуры состыкуем на соответствующие позиции:

## *Рис.5. Преобразование фигур*

При полученнии результата нужно учитывать неидеальность принятой системы отображений фигур на экране, а потому может сложится ситуация, когда, например, высота или(и) длина имеют четные значения, то точку «идеального центра» выделить не удастся и появится двойственность в выборе этой точке, которая в свою очередь будет решаться округлением арифметических действий в языке С++.

## Вывод

Добавить

## Список используемых источников

1. Колинько П.Г. Пользовательские контейнеры / Методические указания по дисциплине «Алгоритмы и структуры данных» - Санкт-Петербург: СПбГЭТУ «ЛЭТИ», 2020.
2. Понимание lvalue и rvalue в C и С++. URL: <https://habr.com/ru/post/348198/> [дата обращения 25.02]
3. Урок №163. Виртуальные функции и Полиморфизм. URL: <https://ravesli.com/urok-163-virtualnye-funktsii-i-polimorfizm/> [дата обращения 26.02]
4. Урок №114. Спецификаторы доступа public и private. <https://ravesli.com/urok-114-spetsifikatory-dostupa-public-i-private/>

## Приложение 1 (Исходный текст программы)