

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет»

Институт высокоточных систем им. В.П. Грязева

Кафедра «Ракетное вооружение»

Утверждено на заседании кафедры  
«Ракетное вооружение»  
«\_19\_»\_\_01\_\_2022 г., протокол № 5

/И.о. зав. кафедрой

 А.В. Смирнов

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**по выполнению лабораторных работ**  
**по дисциплине (модулю)**

**«CALS-технологии и CAD-, CAE-системы»**

**основной профессиональной образовательной программы**  
**высшего образования – программы специалитета**

по специальности  
**24.05.02 Проектирование авиационных и ракетных двигателей**

со специализацией  
**Проектирование ракетных двигателей твердого топлива**

Форма обучения: очная

Идентификационный номер образовательной программы: 240502-01-22

Тула 2022 год

**Разработчик методических указаний**

Дунаева И.В., доцент, к.т.н., доцент

---

(ФИО, должность, ученая степень, ученое звание)



---

(подпись)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАБОТ .....	5
Лабораторная работа №1 .....	5
Лабораторная работа №2 .....	9
Лабораторная работа №3 .....	16
Лабораторная работа №4 .....	26
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	27

## **ВВЕДЕНИЕ**

На современном этапе в промышленности и научно-исследовательской практике используются компьютерные технологии проектирования изделий, разработки технологических процессов и подготовки документации. При этом роль и значение графического представления результатов вычислений непрерывно возрастает, вследствие наиболее емкого и наглядного представления больших объемов данных, которое обеспечивает графическая форма. CALS-технологии и CAD-, CAE-системы дают возможность провести визуальную оценку результатов вычислений, внести необходимые коррективы, отобрать из представленного материала данные для последующей машинной обработки, позволяет существенно увеличить пропускную способность информационного канала, через который осуществляется двухсторонняя связь пользователя и компьютера.

CALS-технологии обеспечивают не только повышение наглядности полученных результатов, но и возможность решения принципиально новых задач, как, например, геометрическое моделирование, дизайн, применение мультипликации при моделировании процессов.

Лабораторные опыты, предлагаемые в данном сборнике, выполняются на этапе общей базовой подготовки. Они предназначены для закрепления на практике знаний, даваемых в курсе лекций и выработки у студентов практических навыков в применении методов компьютерной графики, наиболее широко применяемых в практике, углубление навыков составления алгоритмов и программирования для задач средней сложности, а также развитие навыков оформления разработанных программ. Каждая из лабораторных работ представляет собой небольшое, но самостоятельное и законченное исследование, включающее выбор численного метода, составление алгоритма решения задачи, составление и отладку программы, анализ результатов.

В сборнике содержатся основные сведения теоретического характера, план проведения работ, рекомендации по оформлению отчета, что в комплексе с указанной литературой и курсом лекций позволяет студентам самостоятельно выполнять предлагаемые работы.

## **ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАБОТ**

Каждая лабораторная работа должна выполняться после изучения студентами соответствующих разделов теоретического курса лекций.

К работе на ЭВМ допускаются только те студенты, которые показали знание теоретических вопросов и методики данной работы. Работа считается выполненной после того, как преподаватель, проверив результаты расчета, и ответы студентов на контрольные вопросы, подписывает предоставленный студентом отчет по установленной форме.

Лабораторные работы выполняются с использованием вычислительной техники. Перед занятием преподаватель должен сообщить студентам инструкцию по технике безопасности при работе на ЭВМ.

### **Лабораторная работа №1**

Реализация возможности поворота и сдвига простейших  
геометрических фигур

#### **1. ЦЕЛЬ РАБОТЫ.**

Целью данной работы является:

- практика в разработке графических программ, использующих объектно-ориентированный принцип построения;
- получение навыков использования библиотек стандартных программ и личных библиотек;
- закрепление лекционного материала;
- углубление навыков отладки программ и практической работы на ЭВМ.

#### **2. ОСНОВЫ ТЕОРИИ.**

Графические изображения формируются с использованием примитивов. Примитивами в машинной графике принято считать наименьшие, неделимые с точки зрения прикладных программ, графические элементы, которые используются в качестве базовых для построения более сложных изображений. В аппаратных средствах машинной графики генерация графических примитивов осуществляется специальными блоками, которые попиксельно формируют изображение графического элемента.

Графическими примитивами считают такие элементы, для генерации которых в аппаратные или программные средства вводят команды, которые их идентифицируют.

В зависимости от области применения определяют набор примитивов, которые наиболее целесообразно применить для формирования графических изображений. Так, например, в машиностроительных чертежах наиболее распространены отрезки прямых, дуги окружностей и алфавитно-цифровые символы.

В некоторых приложениях в качестве графических примитивов применяют целое множество связанных между собой графических элементов, которые определяют объект для идентификации.

Примитивы разделяются на геометрические (точки, отрезки прямых, ломаные, дуги кривых, части поверхностей); текстовые и символьные (маркеры).

Различают аппаратный, программный и программно-аппаратный уровни формирования примитивов.

Физическим примитивом называют графический элемент, для генерации которого в графическом устройстве есть соответствующий аппаратный блок. В большинстве систем машинной графики аппаратно реализуют такие примитивы, как точка, отрезок прямой, ломаная линия, строка текста, дуга и пр.

Логическим примитивом называют графический элемент, который является элементарным объектом конкретной программы. Примерами логических примитивов могут служить простейшие геометрические фигуры: треугольник, квадрат, многоугольник, параллелепипед, конус, шар, цилиндр, часть произвольной поверхности и пр.

При программно-аппаратной реализации примитивов на программном уровне выполняются подготовительные расчеты, после чего передается управление соответствующему аппаратному узлу, который завершает генерацию необходимого графического элемента. Так, например, при заполнении области, ограниченной полигоном, программным путем определяют границы контура, а аппаратным – заполнения точек между ними.

Примитив задается:

- параметрами, которые определяют его форму, размеры и местоположение;
- визуальными свойствами, которые определяют его видимость, цвет, яркость, динамические свойства, тип и толщину линий;
- статусом, который определяет отношение к различным операциям, таким как изъятие, преобразование, указание;
- режимом занесения в видеопамять (режимы замещения, наложения, обратного чтения и пр.).

Графическая система включает средства для объединения примитивов. Наличие структурированных данных разрешает получить из ограниченного набора базовых элементов достаточно большое количество видов и проекций изображений. При этом упрощается замена, поворот и перемещение как отдельных фрагментов, так и всего изображения. Структурные связи между графическими данными значительно облегчают процесс поиска и идентификации информации, а также формирование динамических изображений. Последнее объясняется тем, что при формировании изображений, которые воссоздают движение, на экране появляются только несколько новых элементов, а полное изменение производится через некоторый интервал времени, который определяется скоростью движения изображения и размерами поля вывода. Поэтому имитацию движения

осуществляют путем формирования новых фрагментов, при сохранении статических фрагментов без перемен.

Графическим элементом называется упорядоченное множество примитивов одного и того же типа. Понятно, что графический элемент может состоять всего из одного примитива.

Графическим объектом называется множество примитивов, которые имеют одинаковые визуальные свойства и статус, и идентифицируются одним именем. При формировании графического объекта отпадает необходимость в изменении режимов работы графического контроллера (аппаратных блоков), которые отвечают за визуальные свойства.

Поскольку изображение может состояться из примитивов различного типа, то вводят понятия сегмента, который определяется как совокупность графических элементов. Уровню сегмента в языках программирования отвечает такая конструкция, как процедура.

Выполняя группирование примитивов в поименованные сегменты, программист может селективно изменять отдельные части полного изображения путем изъятия или модификации сегмента.

Графические элементы являются синтаксическими объектами (описание графических данных), в то время, как сегменты – семантическими (связь графических данных), например, здание, черчение.

Для работы с элементами графических изображений удобнее применять подходы объектно-ориентированного программирования (ООП). ООП представляет собой технологию программирования, в основе которой лежит тот же подход, что позволяет людям формировать модели объектов реального мира. Каждому реальному объекту сопоставляется объект данных (часть текста программы) объединяющий воедино код и данные. Благодаря данному подходу в значительной мере облегчается сотрудничество рабочих групп в работе над крупными проектами, исключается несанкционированное использование данных. Объект может содержать закрытые массивы данных, работа с которыми осуществляется только функциями, описанными в этом же объекте, сторонний же пользователь, получающий данный объект во владение, имеет возможность обращения только к открытым данным и функциям этого объекта. ООП позволяет экономить время за счет создания иерархических типов данных, порождая новые объекты от существующих и наследуя (полностью или частично) имеющиеся в нем данные и функции.

### **3. ВАРИАНТЫ ЗАДАНИЙ.**

Необходимо описать 2 графических объекта отображающих графические примитивы и элементы (табл.1) по выбору преподавателя. Графический объект должен содержать характеристиками примитива (свойства), соответствующие геометрическим параметрам, видимости, цвету, и функции, обеспечивающие отображение данного объекта (методы).

После создания объектов для примитивов требуется написать небольшую программу построения изображения из данных примитивов, иллюстрирующую использование этих объектов.

Табл.1.

№ варианта	Используемые примитивы и элементы	
1	Отрезок	правильный многоугольник
2	окружность	звезда
3	Эллипс	стрелка
4	окружность	правильный многоугольник
5	дуга	звезда
6	эллипс	правильный многоугольник
7	прямоугольник	звезда
8	окружность	текст
9	отрезок	текст
10	дуга	стрелка

#### **4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ.**

- Определить все свойства примитива, выделить уникальные свойства.
- Описать методы, необходимые для отображения объектов.
- Составить и отладить программу построения изображения.
- Создать диалоговый модуль, для варьирования параметрами (свойствами) графических объектов.
- Ответить на контрольные вопросы.

#### **5. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА.**

Каждый студент оформляет отчет, который должен содержать:

- постановку задачи;
- алгоритмы решения задачи;
- тексты распечаток программ;
- перечень и характеристику ошибок, допущенных в процессе прохождения задания;
- ответы на контрольные вопросы.

К составленным программам следует дать таблицу использованных имен переменных.

#### **6. КОНТРОЛЬНЫЕ ВОПРОСЫ.**

1. Какие структурные элементы изображений считают графическими примитивами?
2. Чем отличаются логические примитивы от физических?
3. Какими атрибутами задаются графические примитивы?
4. Чем обусловлено введение структурных единиц типа графический объект, графический элемент?



## Лабораторная работа №2

Построение простейших геометрических фигур  
с использованием различных растровых алгоритмов.  
Реализация на увеличенной растровой сетке.

### 1. ЦЕЛЬ РАБОТЫ.

Целью данной работы является:

- практика в разработке графических программ;
- закрепление лекционного материала;
- получение навыков использования библиотеки стандартных программ;
- практика в разработке алгоритмов и программ, имеющих модульную структуру;
- углубление навыков отладки.

### 2. ОСНОВЫ ТЕОРИИ.

Достаточно важным понятием для растровой сетки является связность - возможность соединения двух пикселей растровой линией, то есть последовательным набором пикселей. При этом возникает вопрос, когда пиксели  $(x_1, y_1)$  и  $(x_2, y_2)$  можно считать соседними.

Вводится два понятия связности:

- 4-связность, когда пиксели считаются соседними, если либо их  $x$ -координаты, либо  $y$ -координаты отличаются на единицу, то есть  $|x_1 - x_2| + |y_1 - y_2| \leq 1$ ,
- 8-связность, когда пиксели считаются соседними, если их  $x$ - и  $y$ -координаты отличаются не более чем на единицу, то есть  $|x_1 - x_2| < 1, |y_1 - y_2| \leq 1$ .

Понятие 4-связности является более сильным, чем 8-связность: любые два 4-связных пикселя являются и 8-связными, но не наоборот (рис.1).

Так как понятие линии базируется на понятии связности, то естественным образом возникает понятие 4- и 8-связных линий. При этом нужно иметь в виду, что растровое представление объекта не является единственным и возможны различные способы построения.

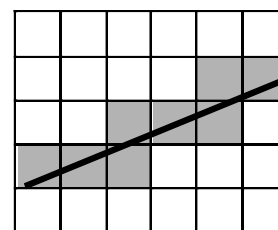


Рис.1.

### Растровое представление отрезка

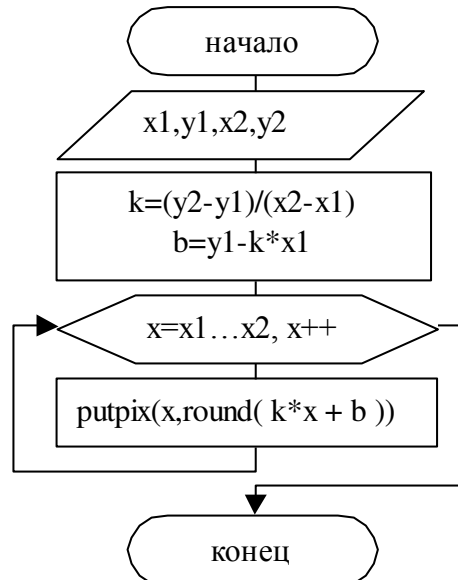
Рассмотрим задачу построения растрового изображения отрезка, соединяющего точки  $(x_1, y_1)$  и  $(x_2, y_2)$ . Для простоты будем считать, что  $0 \leq y_2 - y_1 \leq x_2 - x_1$ .

Тогда отрезок описывается следующим уравнением:

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1} (x - x_1), \quad x \in [x_1, x_2],$$

или  $y = kx + b$ .

Простейший алгоритм растрового представления отрезка имеет вид:



### Алгоритм Брезенхейма

Используя рекуррентное соотношение для вычисления  $y$ , можно упростить функцию, однако это не устраняет *основного недостатка алгоритма* – использования вещественных вычислений для работы на целочисленной решетке.

В 1965 году Брезенхеймом был предложен простой целочисленный алгоритм для растрового построения отрезка, первоначально предназначенный для использования в графопостроителях.

При построении растрового изображения отрезка всегда выбирается ближайший по вертикали пиксел.

При этом из двух точек  $A$  и  $B$  (рис.2) выбирается та, которая ближе к исходной прямой (в данном случае выбирается точка  $A$ , так как  $a < b$ ). Для этого вводится число  $d = (x_2 - x_1)(b - a)$ .

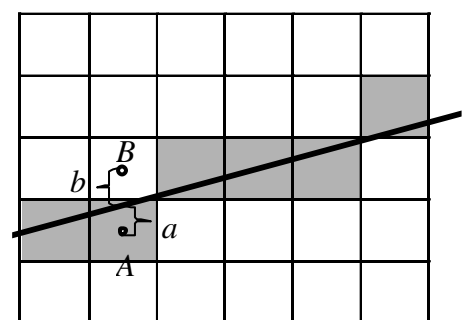
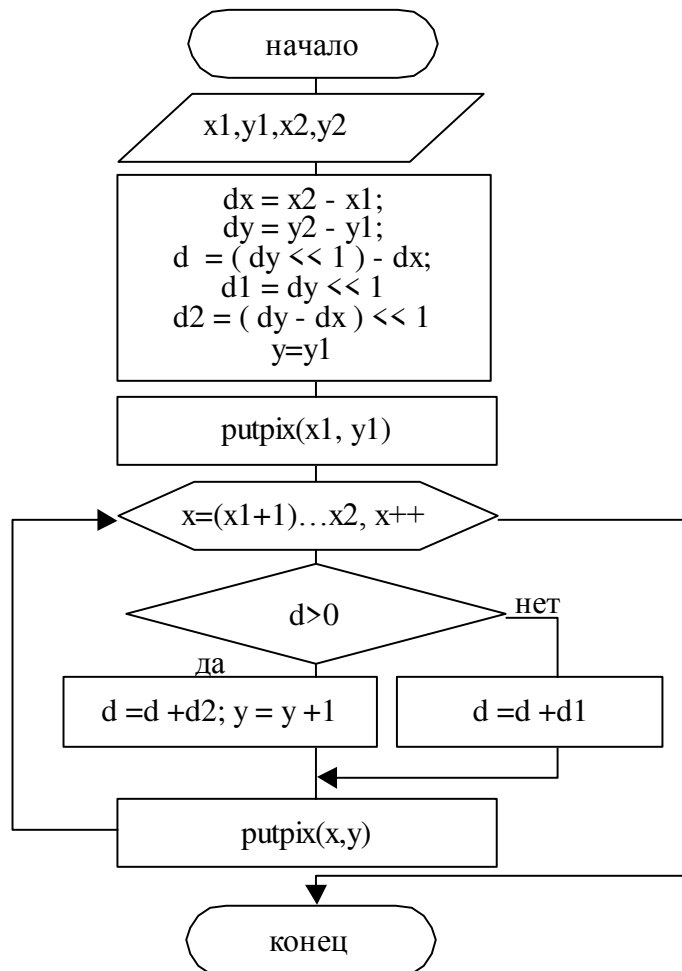


Рис. 2.

В случае  $d > 0$  значение  $y$  от предыдущей точки увеличивается на 1, а  $d$  на  $2(\Delta y - \Delta x)$ . В противном случае значение  $y$  не изменяется, а значение  $d$  заменяется на  $2\Delta y$ .

Блок-схема приведенного алгоритма представлена ниже.

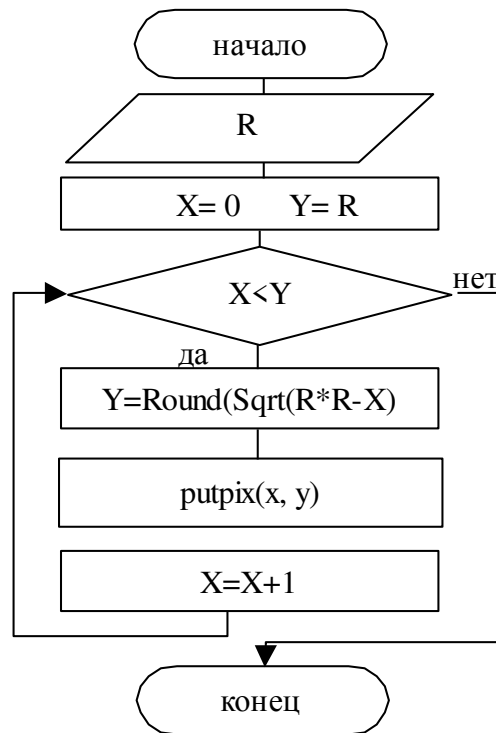


Приведенная схема работает при выполнении условий:  $0 \leq y_2 - y_1 \leq x_2 - x_1$ .

Общий случай произвольного отрезка легко сводится к рассмотренному выше, следует только иметь в виду, что при выполнении неравенства  $|\Delta y| \geq |\Delta x|$  необходимо поменять местами  $x$  и  $y$ .

### Построение окружности.

Построение окружности не сложно. Основное уравнение  $Y^2 = R^2 - X^2$ . Алгоритм имеет вид:



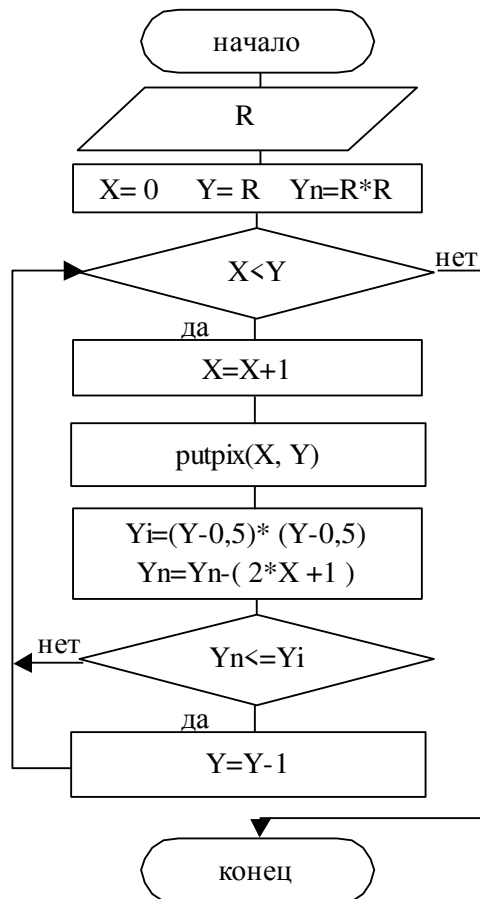
### Построение окружности методом Харденбурга.

В данном методе используется приращение к возводимым в квадрат элементам, чтобы сказать когда пора переходить на следующую точку по неосновной оси. Т.е. отслеживается текущее состояние  $Y^2$  и  $(R^2 - X^2)$  и декрементируется один пиксель по оси  $Y$ , как только  $Y^2$  становится меньше или равной  $R^2 - X^2$ .

Почему это удобно? Зная  $R^2 - X^2$  в начальной точке, в каждой следующей точке это значение вычисляется очень просто

$$R^2 - [X^2 + 2 \cdot X + 1] .$$

Вычитая из последней формулы предпоследнюю, мы получим значение по оси  $Y$  для следующего значения по оси  $X$  зная предыдущее значение по оси  $Y$ .



Алгоритм выполняет построение 1/8 окружности, т.е. до того как X станет равным Y. Далее идет симметричное отражение для полного вывода окружности.

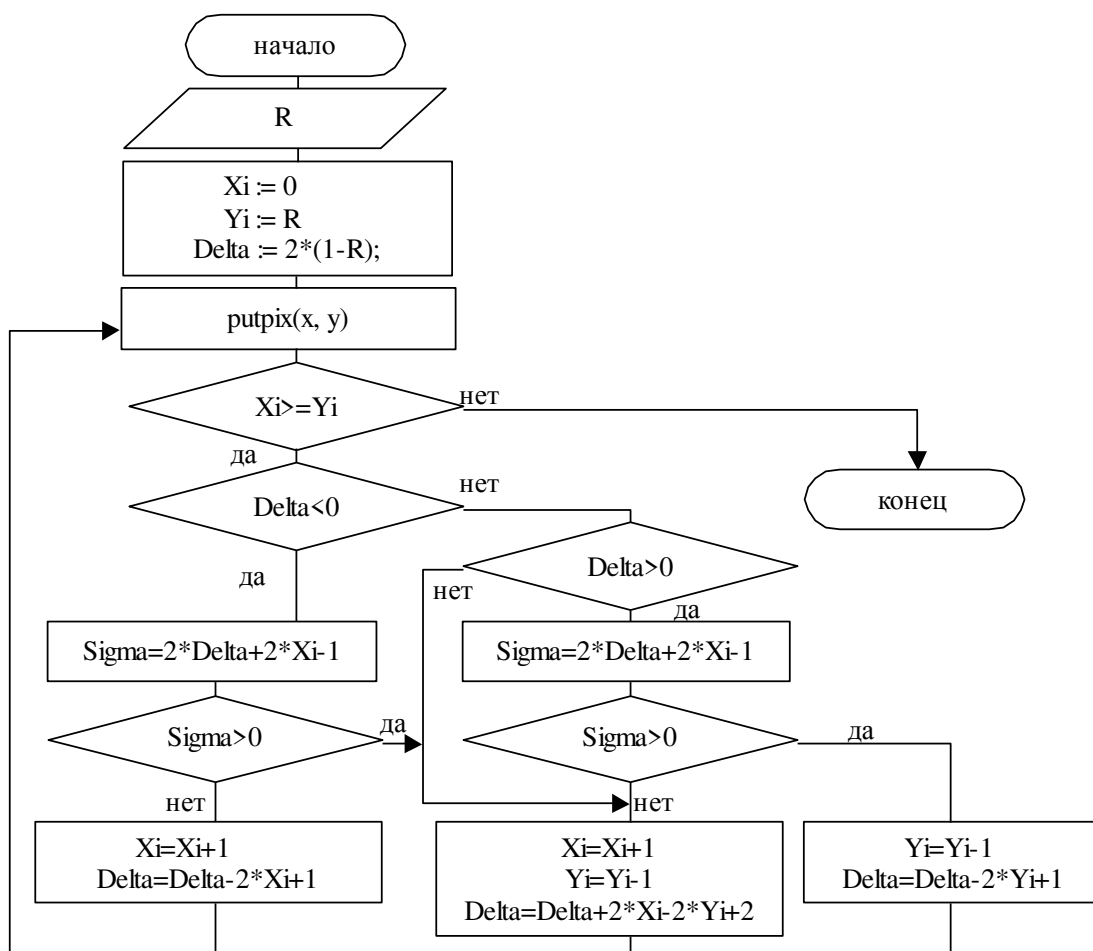
### Построение окружности методом Брезенхема.

Для любой заданной точки на окружности при генерации по часовой стрелке существует только три возможности выбора следующего пиксела, наилучшем способом приближающего окружность: горизонтально вправо, по диагонали вниз и вправо, вертикально вниз. Эти направления:

$$\begin{array}{c}
 |(X+1)^2 + Y^2 - R^2| \\
 |(X+1)^2 + (Y-1)^2 - R^2| \\
 |X^2 + (Y-1)^2 - R^2|
 \end{array}$$

Критерием оценки наиболее подходящего из трех вариантов служит разность между квадратами расстояний от центра окружности до диагонального пиксела (X+1, Y-1) и от центра до точки на окружности R<sup>2</sup>:

$$\Delta = (X+1)^2 + (Y+1)^2 - R^2$$



### 3. ВАРИАНТЫ ЗАДАНИЙ.

Программа, написанная в ходе лабораторной работы должна отображать увеличенную сетку пикселей. Демонстрация работы растрового метода должна производиться на этой сетке. Необходимо предусмотреть возможность пошагового построения растра.

Индивидуальное задание определяется для каждого студента преподавателем.

Табл.3.

№ варианта	Примитив	Метод
1	отрезок A(0,0)-B(10,5)	прямое построение
2	отрезок A(10,1)-B(5,0)	прямое построение
3	отрезок A(0,0)-B(10,5)	Брезенхейма
4	отрезок A(10,1)-B(5,0)	Брезенхейма
5	окружность O(0,0) R=5	прямое построение
6	окружность O(0,0) R=5	Брезенхейма
7	окружность O(0,0) R=5	Харденбурга
8	окружность O(3,3) R=3	прямое построение
9	окружность O(3,3) R=3	Брезенхейма
10	окружность O(3,3) R=3	Харденбурга

#### **4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ.**

- Описать методы формы, отображающие на ней сетку.
- Составить и отладить подпрограмму построения растрового изображения.
- Обеспечить возможность пошагового построения.
- Ответить на контрольные вопросы.

#### **5. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА.**

Каждый студент оформляет отчет, который должен содержать:

- постановку задачи;
- алгоритмы решения задачи;
- тексты распечаток программ;
- перечень и характеристику ошибок, допущенных в процессе прохождения задания;
- ответы на контрольные вопросы.

К составленным программам следует дать таблицу использованных имен переменных.

#### **6. КОНТРОЛЬНЫЕ ВОПРОСЫ.**

1. Что называется растром?
2. Почему возникает необходимость растровых алгоритмов?
3. Приведите известные Вам алгоритмы растровой графики.
4. Какой уровень реализации графики предпочтительнее аппаратный или программный? Почему?

## Лабораторная работа №3

Создание трёхмерных моделей для дальнейшего проведения инженерного анализа изделия.

### Добавление размеров.


В SolidWorks *необязательно* вводить размеры на эскизе до того, как они будут использованы для создания элементов. Однако в данном примере сейчас добавляются размеры для полного определения эскиза. После добавления размеров в эскиз состояние эскиза появится в строке состояния. Все эскизы SolidWorks могут быть представлены в одном из трех состояний. Каждое состояние обозначается другим цветом:

♦ В *полностью определенном* эскизе положения всех объектов полностью описаны с помощью размеров или взаимосвязей, или тех и других одновременно. В таком эскизе все объекты показаны *черным* цветом.

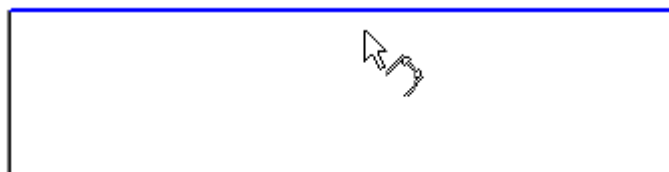
♦ В *недоопределенном* эскизе для полного указания геометрии требуются дополнительные размеры или взаимосвязи. В этом состоянии можно перетаскивать недоопределенные объекты эскиза для изменения эскиза. Недоопределенные объекты эскиза показываются *синим* цветом.

♦ В *переопределенном* эскизе объект содержит противоречивые размеры или взаимосвязи, или те и другие одновременно. Переопределенные объекты эскиза показываются *красным* цветом.

1. Выберите **Инструменты, Параметры**. На вкладке **Настройки пользователя** выберите **Общие**, затем отключите параметр **Ввести значение размера**. Нажмите ОК.

2. Нажмите кнопку  **Размер** на панели инструментов “Взаимосвязи эскиза” или выберите **Инструменты, Размеры, Параллельные**.

Указатель принимает следующую форму



3. Нажмите на верхнюю кромку прямоугольника, затем нажмите в том месте, где требуется нанести размер. Цвет вертикальной линии справа изменился с синего на черный. Указанием размера длины верхней стороны прямоугольника было полностью определено положение



самого правого сегмента. Тем не менее, можно перетаскивать верхний сегмент вверх и вниз. Его синий цвет означает, что он недоопределен.


4. Нажмите на правую сторону прямоугольника, затем нажмите, чтобы нанести ее размер. Верхний сегмент и оставшиеся вершины становятся черными. В строке состояния в правом нижнем углу окна указывается, что эскиз полностью определен.


### **Изменение значений размеров.**

Для изменения размеров используется инструмент **Размеры**.

- ♦ Дважды нажмите на один из размеров. Появится диалоговое окно **Изменить**. Текущий размер выделен.



- ♦ Введите 120мм, затем нажмите . Размер на эскизе изменится в соответствии с новым размером. Значение размера теперь составляет 120 мм.

- ♦ Нажмите кнопку  **Изменить в размер экрана** на панели инструментов “Вид” или нажмите клавишу f, или выберите **Вид, Изменить, Изменить в размер экрана**, чтобы отобразить весь прямоугольник в полный размер и разместить его по центру в графической области.


- ♦ Дважды нажмите на другой размер и введите значение 120мм.

- ♦ Нажмите кнопку  **Изменить в размер экрана** еще раз для размещения эскиза центру.


### **Вытяжка основания.**

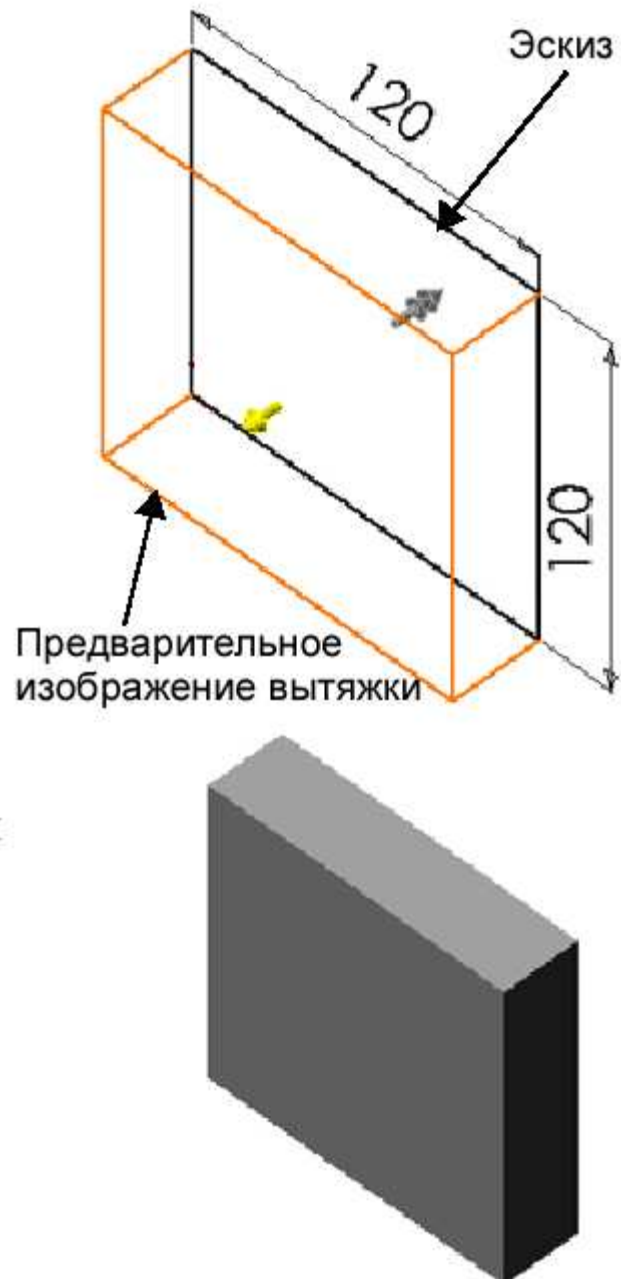
Первый элемент в любой детали называется *основанием*. Этот элемент создается путем вытяжки нарисованного прямоугольника.




1. Нажмите кнопку  **Вытянутая бобышка /основание** на панели инструментов “Элементы” или выберите **Вставка, Основание, Вытянуть**. Появится диалоговое окно **Основание-Вытянуть** PropertyManager (Менеджера свойств) на левой панели, а вид эскиза будет показан в изометрии.
2. В окне группы **Направление 1** выполните следующие операции:

♦ Установите для параметра **Граничное условие** значение **На заданное расстояние**.

♦ Установите **Глубину** , равную 30мм. Чтобы увеличить значение используйте стрелки или введите значение. При использовании стрелок в графической области появляется предварительный вид полученного результата.



3. Нажмите  **ОК** для создания вытяжки. Новый элемент **Основание-Вытянуть** появится в дереве конструирования FeatureManager.
4. Если требуется изменить масштаб, чтобы отобразить модель целиком, нажмите **Z** для уменьшения или **Shift+Z** для увеличения.

5. Нажмите на знак плюса рядом с записью **Base-Extrude (Основание-Вытянуть)** в дереве конструирования FeatureManager. **Sketch1 (Эскиз 1)**, который использовался для вытяжки элемента, теперь отображен под данным элементом.



### **Сохранение детали.**

1. Нажмите кнопку **Сохранить** на панели инструментов “Стандартная ” или выберите **Файл, Сохранить**. Появится диалоговое окно **Сохранить как**.
2. Введите **Tutor1** и нажмите кнопку **Сохранить**. Расширение **.sldprt** добавится к имени файла, и файл будет сохранен в текущем каталоге. Чтобы сохранить файл в другом каталоге, используйте кнопку обзора Windows для указания каталога, затем сохраните файл. В именах файлов не учитывается состояние регистра.







### **6. КОНТРОЛЬНЫЕ ВОПРОСЫ.**

1. Что происходит с моделью детали при изменении размеров в процессе работы?
2. Приведите пример элементов, использующихся при построении 3-мерной модели?
3. Какой эскиз является переопределённым?

. Сборка – это узел, состоящий из двух или более деталей, называемых также компонентами, в одном документе SolidWorks. Расположение и ориентация компонентов задается с помощью сопряжений. Сопряжения образуют взаимосвязи между гранями и кромками компонентов.


### **Создание основания**

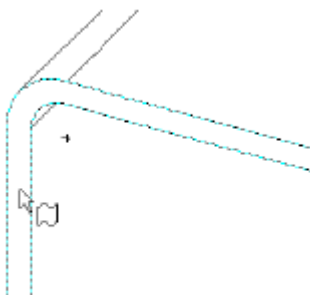
Для того чтобы создать основание для новой детали, можно воспользоваться методами, использовавшимися в предыдущей работе.

1. Откройте новую деталь.
2. Нажмите кнопку **Эскиз**  и нарисуйте прямоугольник, начиная с исходной точки.
3. Нажмите кнопку **Размер**  и укажите размер прямоугольника, равный 120 x 120 мм.
4. Нажмите кнопку **Вытянутая бобышка/основание**  и вытяните прямоугольник, установив в параметре **Граничное условие** значение **На заданное расстояние**, на **Глубину** 90 мм .
5. Нажмите кнопку **Скругление**  для скругления четырех показанных кромок с радиусом 10 мм.
6. Нажмите кнопку **Оболочка** . В качестве грани для удаления выберите *лицевую* грань модели и установите **Толщину** 4 мм.
7. Сохраните деталь как **Tutor2**. (К имени файла будет добавлено расширение .sldprt.)



### **Создание выступа на детали**

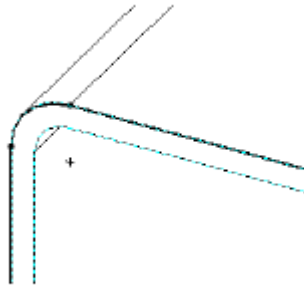
В этом разделе для создания геометрии эскиза используются инструменты **Преобразование объектов** и **Смещение объектов**. Затем с помощью выреза создается выступ для присоединения детали.

1. Нажмите кнопку **Увеличить элемент вида**  или выберите **Вид, Изменить, Увеличить элемент вида**, а затем с помощью мыши выберите угол детали, как показано. Нажмите кнопку **Увеличить элемент вида** еще раз, чтобы выключить инструмент.

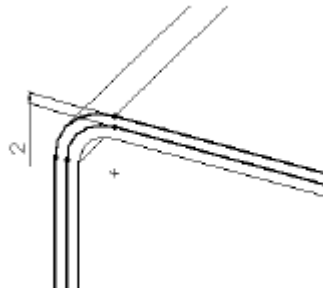


2. Выберите тонкую стенку на лицевой грани детали и нажмите кнопку **Эскиз** для открытия эскиза. Кромки грани детали высвечиваются.

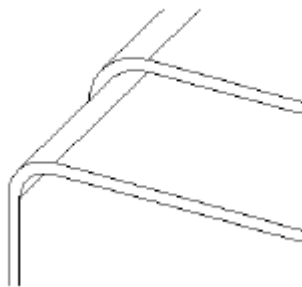
3. Нажмите кнопку **Преобразование объектов**  на панели инструментов – “Инструменты эскиза” или выберите **Инструменты, Инструменты эскиза, Преобразование объектов**. Наружные кромки выбранной грани проецируются (копируются) на плоскость эскиза в виде **линий и дуг**.
4. Нажмите снова на лицевую грань.
5. Нажмите кнопку **Смещение объектов**  на панели инструментов – “Инструменты эскиза” или выберите **Инструменты, Инструменты эскиза, Смещение объектов**.



6. Установите **Расстояние смещения**, равное 2 мм. Предварительный просмотр показывает смещение, выдвинутое наружу.



7. Выберите параметр **Реверс**, чтобы изменить направление смещения.
8. Нажмите ОК .  
В эскиз добавляется ряд линий, смещенных от наружной кромки выбранной грани на 2 мм. Это соотношение сохраняется, если изменяются исходные кромки.
9. Нажмите кнопку **Вытянутый вырез** или выберите **Вставка, Вырез, Вытянуть**.
10. В окне группы **Направление 1** установите **Глубину**, равную 30 мм, и нажмите **ОК**. Материал между двумя линиями вырезается, образуя выступ.



### **Изменение цвета детали**

Цвет и внешний вид детали или ее элементов можно изменять.

1. Нажмите значок Tutor2 в верхней части дерева конструирования **FeatureManager**.

2. Нажмите кнопку **Закрасить** .

3. Нажмите кнопку Редактировать цвет на панели инструментов

“Стандартная”. Появится диалоговое окно **Редактировать цвет** .

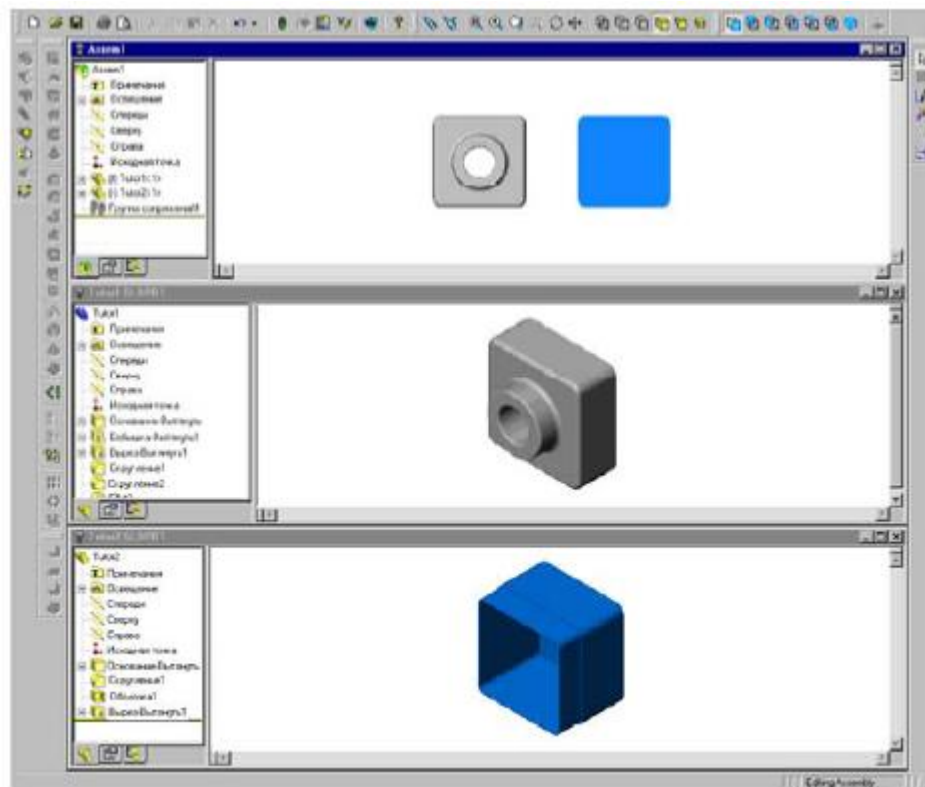
4. Выберите в палитре нужный цвет и нажмите **ОК**.

5. Сохраните деталь.

### **Создание сборки**

Теперь можно создать сборку, используя две детали.





1. Если файл Tutor1.sldprt еще не открыт, нажмите кнопку **Открыть** на панели инструментов “Стандартная” и откройте его.
2. Нажмите кнопку **Создать** на панели инструментов “Стандартная”. Появится диалоговое окно **Новый документ SolidWorks**.
3. Выберите вкладку Учебное пособие, нажмите на значок сборки и нажмите **ОК**.
4. Выберите **Окно, Отобразить окна сверху вниз**, чтобы вывести на экран все три окна. Закройте лишние окна.
5. Перетащите значок Tutor1 с вершины дерева конструирования **FeatureManager** Tutor1.sldprt в дерево конструирования **FeatureManager** в окне сборки (Assem1). При добавлении детали в сборку подобным образом в детали автоматически формируется исходная точка сборки. Когда деталь формирует исходную точку сборки:
  - исходная точка детали совпадает с исходной точкой сборки;
  - плоскости детали и сборка выровнены по отношению друг к другу.
6. Перетащите значок Tutor2 из Tutor2.sldprt в графическую область окна сборки, поместив его рядом с деталью Tutor1.



7. Сохраните сборку как Tutor. (К имени файла будет добавлено расширение .sldasm.) При появлении сообщения о сохранении соответствующих документов нажмите **Да**.
8. Потяните за угол окна со сборкой, чтобы увеличить его, или нажмите на значок **Развернуть** в правом верхнем углу, чтобы развернуть окно во весь экран. Далее нет необходимости сохранять на виду окна Tutor1.sldprt и Tutor2.sldprt.
9. Нажмите кнопку **Изменить** в размер экрана.


### **Сопряжение компонентов**

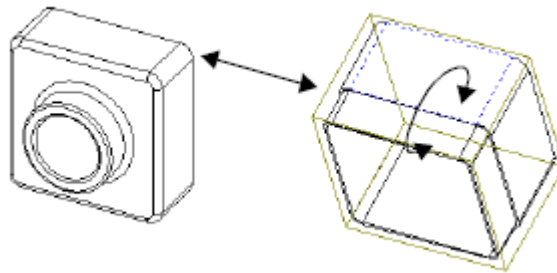
В этом разделе определяются взаимосвязи сопряжения компонентов в сборке, благодаря которым они ровно и точно подгоняются друг к другу.

1. Нажмите кнопку **Изометрия**  на панели инструментов “Стандартные виды”.
2. Нажмите кнопку **Сопряжение**  на панели инструментов “Сборка” или выберите **Вставка, Сопряжение**.
3. Нажмите на верхнюю кромку Tutor1, затем нажмите на верхнюю наружную кромку выступа Tutor2. Кромки появляются в списке **Объекты для сопряжения** .
4. В окне группы **Настройки** сопряжения выполните следующие операции:
  - В качестве типа сопряжения выберите **Совпадение** .
  - Выберите **Максимально близко** для параметра **Выровнять сопряжения**.
5. Нажмите **Предварительный просмотр** для просмотра сопряжения. Выбранные кромки двух компонентов совпадают.

6. Нажмите **ОК**.

Положение компонента Tutor2 в сборке полностью еще не определено, как показывает префикс (-) в дереве конструирования **FeatureManager**. Tutor2 все еще имеет некоторую степень свободы перемещения по направлениям, которые пока не подчинены взаимосвязям сопряжения.

1. Нажмите кнопку **Переместить компонент** .
2. Выберите компонент Tutor2, нажмите и удерживайте левую кнопку мыши.

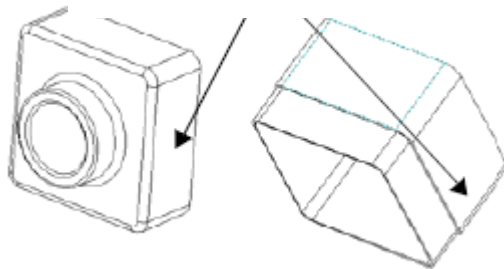


3. Попробуйте переместить компонент в разные стороны, чтобы определить имеющиеся степени свободы, а затем отпустите левую кнопку мыши.
4. Нажмите кнопку **Переместить компонент** еще раз, чтобы выйти из режима перемещения.

### **Добавление сопряжений**

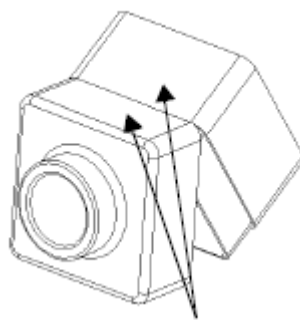
1. Выберите крайнюю правую грань компонента, затем нажмите клавишу **Ctrl** и выберите соответствующую грань в другом компоненте.

Выберите эти грани



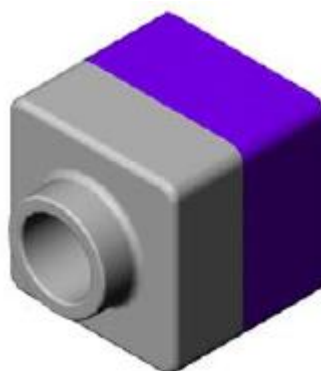
2. Нажмите кнопку **Сопряжение**.
3. Выберите **Совпадение** и **Максимально близко**.
4. Нажмите **Предварительный просмотр** для просмотра сопряжения.
5. Нажмите **ОК**.
6. Повторите шаги с 1 по 5, выбирая верхние грани обоих компонентов, для добавления другого сопряжения **Совпадение**.





Выберите эти грани

7. Сохраните сборку.



### **3. ВАРИАНТЫ ЗАДАНИЙ.**

В результате выполнения данной работы должна быть создана трехмерная модель детали. Индивидуальный вариант детали определяется преподавателем для каждого студента.

### **4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ И МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ.**

- Описать алгоритм создания 3-мерной сборки;
- Указать элементы, составляющие сборку, в порядке их создания.
- Перечислить операции, которые необходимо осуществить для создания сборки.
- Ответить на контрольные вопросы.

### **5. УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА.**

Каждый студент оформляет отчет, который должен содержать:

- индивидуальное задание;
- Алгоритм, отражающий последовательные этапы создания 3-мерной сборки;
- перечень и характеристику ошибок, допущенных в процессе прохождения задания;
- ответы на контрольные вопросы.

### **6. КОНТРОЛЬНЫЕ ВОПРОСЫ.**

1. Что происходит с трехмерной сборкой при изменении размеров входящих в неё деталей в процессе работы?
2. Какие виды сопряжения могут быть выбраны при создании сборки?

## Лабораторная работа №4

Работа с элементами и деталями. Работа со сборками. Создание и оформление чертежей.

### 1. ЦЕЛЬ РАБОТЫ.

Целью данной работы является:

- приобретение навыков инженерного расчёта средствами SolidWorks;
- закрепление навыков в работе в системе SolidWorks;

Примером может служить задача разработки и испытания в сжатые сроки нового гироскоординатора (ГК) увеличенной добротности, который представляет собой карданный подвес с разгоном ротора пружинным гиromотором разобщенного типа. Работа была разбита на последовательные этапы:

- создание деталей и сборка ГК;
- проверка собираемости изделия;
- разработка необходимой конструкторской документации;
- передача электронного макета изделия в смежное отделение.

При этом рассматривалось два варианта конструкции гироскоординатора:

- ротор кардана подвеса выполнен составным;
- ротор карданного подвеса выполнен цельнометаллическим.



Конструкция ГК с составным подвесом.



Модель ГК с цельнометаллическим карданным подвесом.

По результатам 3D-моделирования основных деталей подвеса (ротора, внутренней и наружной рамки) были получены данные (моменты инерции и массовые характеристики), необходимые для математического моделирования работы гироскопа. При работе с 3D-моделью оценивались динамические зазоры между деталями карданного подвеса. С точки зрения собираемости наибольший интерес представляли вопросы обеспечения установки ротора во внутреннюю рамку, внутренней рамки в наружную рамку. При проверке на модели были выявлены ошибки, что позволило на ранних этапах провести необходимые изменения в конструкции. В гироскопическом приборе большое внимание уделяется вопросу балансировки карданова подвеса и входящих в его состав сборочных единиц. Результаты проектирования в системе SolidWorks позволили облегчить процедуру отработки конструкции и балансировку данного узла и повысить уровень разработки приборов. С этой целью в SolidWorks был произведен расчет расположения центров масс:

- ротора;
- сборочной единицы: ротор во внутренней рамке;
- сборочной единицы: карданов подвес.

Для сведения дисбаланса к минимуму детали гироскопа (ротор, внутренняя рамка, элементы крепления) были выполнены симметричными. Однако, при рассмотрении сборочной единицы подвес, вследствие несимметрии от узла арретира на наружной рамке и наличия паза на наружной рамке, для прохождения к ротору ленты со стороны пружинного двигателя, было выявлено значительное смещение центра масс от точки пересечения осей вращения. На твердотельной модели наружной рамки была проведена выборка металла с целью совмещения центра масс карданного подвеса с точкой пересечения осей вращения.

## **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

### **Основная литература**

- 1.Соллогуб А.В. SolidWorks 2007 технология трёхмерного моделирования. – СПб.: БХВ – Петербург, 2007. – 352 с.: ил.
- 2.Бобровский С. И. Технологии C++Builder. Разработка приложений для бизнеса. Учебный курс. — СПб: Питер, 2007.- 560 с.

### **Дополнительная литература**

- 1.Компьютерные чертежно-графические системы для разработки конструкторской и технологической документации в машиностроении : Учеб.пособие для нач.проф.образования / А.В.Быков,В.Н.Гаврилов,Л.М.Рыжкова и др.;Под ред.Л.А.Чемпинского .—

- М. : Академия, 2002 .— 224с. : ил. — (Профобразование) .— Библиогр.в конце кн. — ISBN 5-7695-0903-1 /в пер./ : 267.00.
- 2.Крылов, О.В. Метод конечных элементов и его применение в инженерных расчетах : Учеб.пособие для вузов / О.В.Крылов .— М. : Радио и связь, 2002 .— 104с. : ил. — Библиогр.в конце кн. — ISBN 5-256-01627-X : 60.00.
- 3.Порев В.Н. Компьютерная графика. – СПб.: БХВ – Петербург, 2004, – 432 с.: ил.