An Essential Change to the Training of Computer Science Teachers: The Need to Learn Graphics

Vadim Grinshkun [a], Esen Bidaibekov [b, *], Svetlana Koneva [b], Gulnaz Baidrakhmanova [b]

[a] Moscow City Pedagogical University, Moscow, Russian Federation
[b] Abay Kazakh National Pedagogical University, Almaty, Republic of Kazakhstan

Abstract
The higher education system of the Republic of Kazakhstan emphasizes the need for the development of institutional science, including the field of information technology, with a priority on fundamental research. Professional training of future computer science teachers requires them to know the range of tasks related to computer graphics, and the ability to accomplish them. The training of a computer science teacher goes beyond the task solution and demands that the teacher have the basic knowledge needed in the field of computer science and information technology, and especially in computer graphics. In addition to the traditional and fundamental tasks of teaching computer graphics, the teacher is faced with the use of computer graphics elements and their tools in their teaching. It should be noted that present trends in the digitalization of education have made such tasks fundamental to teaching. Based on the above, this article analyzes the need to teach computer graphics related to the activities of future computer science teachers, and possible methods of their solutions, and the field of use.

Keywords: professional training, a computer science teacher, fundamentalization in education, computer graphics, methods of solution.

1. Introduction
The system of higher education of the Republic of Kazakhstan emphasizes the need to develop institutional science, including the field of information technology, with a priority on fundamental research (Nazarbayev, 2018). In this regard, the educational system cannot bypass the methods of training future computer science teachers in the subject field of computer science and information technology, which is characterized by a fairly high rate of updating of both the tools and technologies. The raining of computer science teachers should be done in a way that provides

---
* Corresponding author
E-mail addresses: esen_bidaibekov@mail.ru (E. Bidaibekov), grinshkun@yandex.ru (V. Grinshkun),
konevasveta@mail.ru (S. Koneva), gulnaztai83@mail.ru (G. Baidrakhmanova)

them with the opportunity to flexibly vary the direction and content of their work in response to the development and change of technologies and/or market requirements. In other words, computer science teachers should not only acquire knowledge but also develop the skills and ability to work and educate themselves by using the latest technology. In addition future computer science teachers will need to be able to teach schoolchildren the fundamentals of computer science, to help students understand that computer science is a basic science, to make them use their knowledge and skills in the field of computer science to learn other disciplines, and to aid them in their future work (Bidaibekov et al., 2010). In the Republic of Kazakhstan and many other countries, it is proposed that the government should ensure the fundamental training of teachers in computer science through the integration of information disciplines that include the use of both mathematical and information models.

The required training should consist of the introduction of the fundamental principles in most disciplines, which are in some way related to computer science. One of these disciplines, which should be studied by all future teachers of computer science in a university, are the academic subjects related to different types of computer graphics. The fundamental training of future computer science teachers in the field of computer graphics should be characterized by an emphasis on the essential foundations and links in computer graphics, and the training itself should be focused on the use of the natural internal links between computer graphics courses, computer science and mathematics, as well as other interdisciplinary connections (Bidaibekov et al., 2005). Knowledge and understanding of computer science and information technologies are essential components of the modern teacher in higher education. In this regard, the training of future teachers of computer science in the field of computer graphics should be based on the relationship between the fundamental foundations of computer science and various applied fields; the training should also promote the development of relevant professional competencies in future computer science teachers (Bidaibekov et al., 2017).

2. Classification of computer graphics tasks
The study of computer graphics during university training is an indispensable component of the professional training of any specialist in the field of computer science; it is a necessary component for the development of applied skills and the students' fundamental knowledge of computing students. In the last decade, the study of the practice and theory of training computer science teachers in universities in the Republic of Kazakhstan has revealed the need to revise the content of this training with a focus on international experience (Nazarbayev, 2018). The professional training of future computer science teachers should include a system of teaching practical skills and foster the ability to perform any educational and pedagogical task.

Teachers of computer science should be required to master a broader and deeper set of tools related to the fundamentals of computer studies. In addition to the traditional tasks of understanding computer graphics, the teacher is now faced with the use of computer graphics elements and their tools as part of their teaching.

Since the scope of application of computer graphics elements in the context of education is broad enough, the set of computer graphics tasks can be classified as a certain system. In order to apply the whole range of methodological principles for constructing a system of exercises and tasks related to computer graphics, it is necessary to identify each task's main feature(s) in order to classify them. In addition, there is a need to use the capabilities of this system to solve pedagogical problems, which are used in the professional activities of future computer studies teachers.

All of the above allows us to highlight the following requirements regarding computer graphics for the training of future computer science teachers. These include:
• the high rate of updating of means and technologies, including the field of information technologies;
• requirements of the market to form not only knowledge but also the abilities and skills needed for work and ongoing self-education using the latest means of information and digitalization;
• the need to instill students' understanding of computer studies as a fundamental science;

- exposure to the systematic use of knowledge and skills used in computer science in the studies of other disciplines, as well as developing the ability to apply them in a subsequent career;
- updating the content of the programs of the schools' computer studies classes;
- the need to use the methods and tools of computer graphics to complete pedagogical tasks; and
- the need to develop institutional science with a priority on basic research, including in the field of information technology.

Guided by the above principles, we offer a classification of the tasks that covers the training of teachers of computer studies. These include:

1. geometric tasks,
2. tasks of two-dimensional and three-dimensional graphics, and
3. tasks of study aids, i.e. pedagogical tools.

In this classification, the order of consideration of the task types is important, as each subsequent type of task implies knowledge of the previous ones. Within a single type, the tasks can be divided into:

- easy, medium and complex;
- school and university higher education course tasks; and
- standard and non-standard tasks.

Thus, the main goal is to classify computer graphics tasks while staying focused on the future training of computer studies teachers, and to offer common algorithms for their solutions.

Methods for solving computer graphics tasks depend on many factors: the type of graphics (raster or vector graphics), the dimension of the task being solved (two-dimensional or three-dimensional graphics), and the tools of the graphic system. The content and the level of task complexity are influenced by the student's level of knowledge, skills and the student's ability to master the tools of the graphic system. The fundamental basis for the successful solution of computer graphics tasks is the good mathematical understanding and ability of the future computer studies teacher.

Shvetsky (1994) believes that the fundamental element required to effectively teach computer science can be achieved through a combination of training theory, abstraction, and implementation. The study of mathematical algorithms and special data structures in a specific programming language allows students to achieve fundamental knowledge in the field of computer science (Shvetsky, 1994). This clearly implies the need to study the mathematical algorithms of raster and vector graphics, two-dimensional and three-dimensional graphics.

An analysis of the existing methods and procedures for solving a particular graphical task with the help of various tools allows us to single out the general steps needed to accomplish the task. For ease of description of the stages, we use the concept of "graphic object," which, depending on the complexity of the task, can be either a graphic primitive or a group of graphic objects.

Let us highlight the following stages of solving tasks in computer graphics, taking into account the fundamental training of future computer science teachers:

1. the study of the mathematical foundations of building a graphic object (or in the case of group objects, each individual object),
2. the study of the algorithm of computer graphics construction of the object (or group),
3. the study of tools of visualization of the graphical object (the language and programming environment, a graphical editor, and special math software),
4. computer implementation of the algorithm for building a graphical object in the selected visualization environment, and
5. an analysis of the graphical object created (an analysis of the selected method, testing the program, and saving the results).

From the point of view of the computer graphics training, the selected stages are invariants relative to the type of graphics and the dimension of the task. The tools for solving the task are variable.

Next, we consider in more detail the application of the selected steps in order to solve different types of tasks in computer graphics.

3. Methods for solving geometric tasks

It is noted that the achievement of a high level of mathematical training of students is observed in solving mathematical tasks, especially geometric tasks (Baranova, 2000). Geometry, because of its use in computer graphics is a propaedeutic course. The basic concepts of geometry – point and line – are associated with the concepts of pixel and segment in computer graphics, and the basic geometry axioms and theorems are necessary knowledge for solving computer graphics tasks. In turn, the solution of geometric tasks with the help of computer graphics tools allows the teacher to increase the students' interest in geometry, to repeat and deepen the students' knowledge and skills of solving geometry tasks, and to visualize the results of solving a geometry task.

There are a number of conditional classifications of geometric tasks: purely geometric and practical; standard and non-standard; the use of individual theorems and a combination of theorems and formulas; training, searching and problematic, etc. The most common classifications of geometry tasks are the following: to calculate, to prove, and to build (constructive problems). Geometry tasks during school are standard, use ready-made algorithms, or the algorithms are derived from definitions or theorems, are studied before applying them to computer science lessons, and do not require a lot of time to study the mathematical solution of the task during the study of computer science.

Geometrical construction tasks are those tasks that are achieved (solved) with the help of a compass and a ruler, and at the modern stage of development of information technologies, in the corresponding graphic editor. Geometrical construction tasks are a component of elementary geometry and have great potential for teaching students to choose a method for solving them and applying them in practice. Geometrical construction tasks also have close inter-subject connections with drafting courses, algebra and physics. The formation of skills and abilities to solve geometric construction tasks is important in school geometry courses in the Commonwealth of Independent States (CIS) countries, including drawing courses. A number of studies during the mid-20th century are devoted to methods for solving these tasks, among which are the works of Alexandrova (1954) and Olifer (1955).

Over the past sixty years, the importance of geometric tasks in the school courses in the CIS countries has gradually been decreasing (a decrease in the number of hours is observed, although their role is quite large). It is possible to compensate for this deficiency by making interdisciplinary connections between geometry and computer studies. Olifer (1955) proposes using construction tasks as a means of establishing relationships between geometry and practice, and proposes the following classification of these tasks: tasks for the construction of polygons, tasks for the construction of circles or arcs, tasks for the construction of lines or segments, and tasks for the construction of points. The analysis of the programs and textbooks of the school courses of computer studies in Russia and the Republic of Kazakhstan shows that such a range of tasks are offered to students to solve them (or rather, build) in a graphic editor or using a programming language.

In the course of "Computer Studies" for grades 6-7 (Makarova, 2000) the use of a raster graphics editor has been proposed to solve geometric tasks. The authors propose to study algorithms and technologies for working with graphical objects, including cyclic algorithms. The following tasks are suggested: with the help of graphic primitives (tools) to draw a house, a snowman, a plan of the terrain, the construction of a symmetrical ornament, finding the middle of a segment, the division of a square into four equal squares, and the construction of a set of inscribed squares and circles, etc.

Next, we consider the example of finding the middle of a segment.

Task 3.1. Find the middle of the horizontal segment.

Solution: This method is based on the properties of an isosceles triangle: the height dropped from the vertex of an isosceles triangle to the base is the bisector, the height, and the median. It is known that the median of a triangle divides the opposite side into two equal parts, i.e. in half.

The method of dividing a segment into two equal parts is reduced to the following actions:

1. Build a horizontal line.
2. For any end of the segment, use the Line tool to build a line at an angle of 45 degrees.
3. Similarly, from the other end, build a line.
4. From the point of intersection of the lines, lower the line to the intersection with the original segment.

Geometric constructions and their solutions were known to al-Farabi, one of the greatest thinkers and encyclopedists of the early Middle Ages. Among the numerous mathematical works of al-Farabi, a special place is occupied by the "*Book of spiritual clever tricks and natural secrets about the intricacies of geometric figures,*" the only copy of this manuscript, which is kept in the library of Uppsala University in Sweden (Kubesov, 1972). It offers unique algorithms for solving a variety of geometric tasks for construction using a compass and a ruler, which are important in the land surveying, architecture, engineering, and geodesy.

Interest in such tasks for many centuries is not only due to their beauty and the originality of the methods of their solutions, but is mainly due to their great practical value. At present, the geometric tasks of construction are also of great interest since the design of construction objects, architecture, the construction of various techniques and many other applied problems are based on geometric constructions. The huge role of such tasks in mathematical development is in the hands of the future computer science teacher. As one of the meaningful lines of the school geometry course, they are a very important element in the teaching of computer graphics, one of its integral parts (Bidaibekov et al., 2017). Among them there are a lot of tasks of different levels for the construction of regular polygons, including tasks that are unsolvable with the help of a compass and a ruler.

The members of the Department of Informatics and Education Informatization of Abay Kazakh National Pedagogical University propose to enrich the content of the curriculum of geometry and computer science of the Nazarbayev Intellectual School by studying geometric task construction with a compass and a ruler according to the methods and algorithms described in the works of al-Farabi.

Modern methods and procedures for solving graphic tasks allow us to build models of geometric shapes with given properties (equilateral triangle, hexagon, isosceles triangle, etc.), using the axioms and theorems of geometry, the possibility of computer graphics, and the dynamic geometry environment GeoGebra (Bidaibekov et al., 2017).

Next, we consider examples of the construction of a regular octagon according to the algorithms of al-Farabi.

Task 3.2. Modeling objects with the given geometric properties. The construction of a regular octagon.

Solution: One of the simplest methods for constructing a regular octagon is based on the construction using a regular quadrilateral and straight line segments. The method of constructing a regular octagon needed to construct two regular quadrangles and connect their vertices. Thus, it turns out to be a regular octagon (Figure 1).
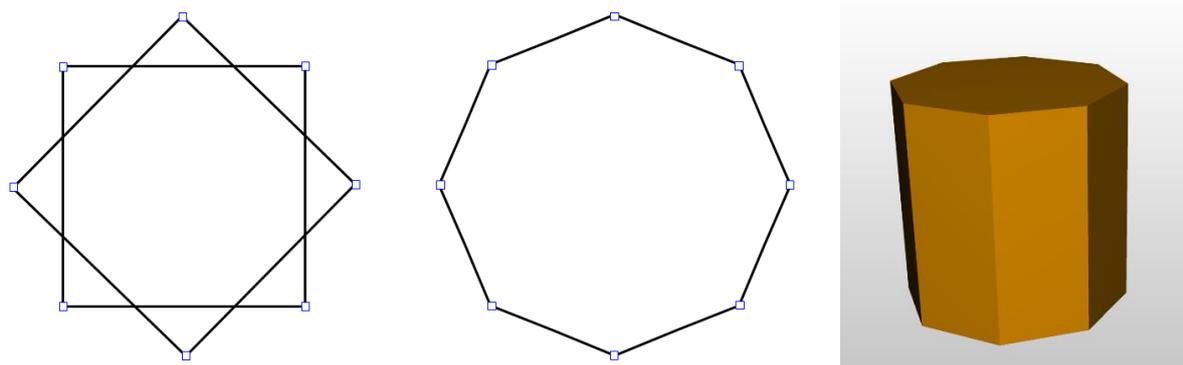
(a) Overlapping two quadrilaterals and connecting vertices.　　(b) A regular octagon.　　(c) An octagonal prism.

Fig. 1. An example of constructing a regular octagon and an octagonal prism

Consider another task involving the constructing a regular octagon.

Task 3.3. Construction of a regular octagon inscribed in a circle.

Solution: al-Farabi's solution to this problem is given as follows (Kubesov, 1972). Draw a circle, build an equilateral line and equilateral quadrilateral lines in it, then split each semicircle in half, and connect the division points with straight lines with new points. We obtain an equilateral and equiangular octagon (Figure 2).

Next, we present an algorithm for solving the task of constructing a regular octagon in the GeoGebra program environment:

1. Draw a circle.
2. Fit an equilateral and equiangular quadrilateral in the circle.
3. Divide the semicircles AB, BC, CD and DA in half.
4. Divide the semicircle AB in half, mark the point E.
5. Divide the semicircle BC in half, mark point G.
6. Divide the CD semicircle in half and mark the point H.
7. Divide the semicircle DA in half, mark point I.
8. Connect the division points with straight lines with new points: A, E, B, G, C, H, D, I.

We obtain an equilateral and equiangular octagon inscribed in a circle (Figure 2).
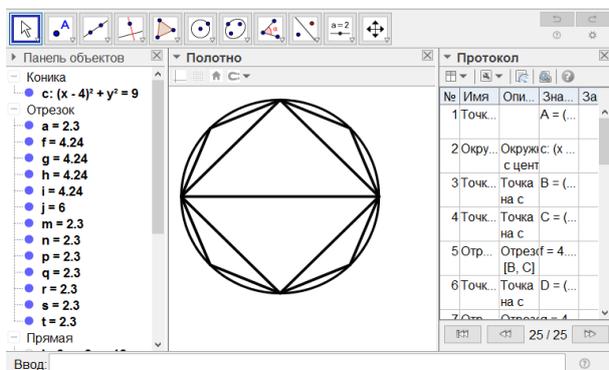
(a) Construction of a regular octagon inscribed in a circle.
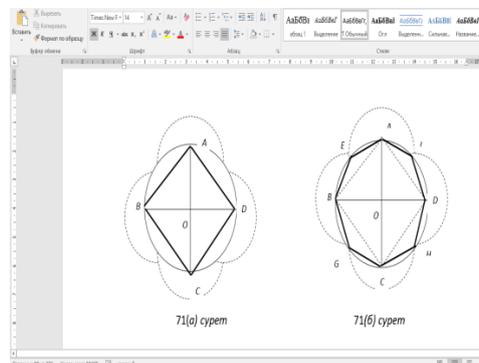
(b) External construction of a regular octagon inscribed in a circle.

Fig. 2. Construction of a regular octagon inscribed in a circle in the GeoGebra program environment.

The computer model created as a result of the constructions of the above algorithm in the software environment of GeoGebra is mathematically accurate. Computer science teachers are required to build skills using the al-Farabi algorithm in this software environment and substantiate the correctness of this algorithm based on fundamental knowledge in the field of planimetry.

Thus, the ability to solve geometric tasks, especially tasks of construction, with the help of computer graphics tools by future computer science teachers is important both for the mathematical preparation of students and for fundamental teacher training in the field of computer science and computer graphics. Geometric tasks are fundamental tasks proven essential over time, and therefore they are not subject to the development of science and technology, but only reinforce the basic content of teaching computer science, including computer graphics. Algorithms for solving geometric tasks are studied in more detail in the computational geometry section of computer science.

4. Methods for solving tasks of two-dimensional and three-dimensional graphics

Two main versions of graphic data descriptions are used in computer graphics, differing in the number of independent coordinates needed to determine the position of graphic objects and their elements – two-dimensional and three-dimensional graphics.

These kinds of computer graphics tasks are often identified with computer graphics algorithms and separated into an independent training course, which is studied in universities. In terms of the scope of the tasks, two-dimensional and three-dimensional graphics are the most extensive and range from displaying a point on the screen using a programming environment to

the three-dimensional modeling of research results. The mathematical foundations of solving two-dimensional and three-dimensional graphics are laid in school when studying of geometry, planimetry and stereometry at various levels of education.

The geometric tasks we considered earlier, especially the construction tasks, are both two-dimensional and three-dimensional. These tasks can be solved both in a graphical editor and by using a high-level programming language using graphic primitives. Visualization of a graphic object in a graphic editor is a simpler and more understandable method and requires only knowledge of the algorithm for solving the task and the ability to correctly use the tools of a graphic editor. Visualization with the help of a programming language is more difficult, since it requires an additional knowledge of the functions and procedures of the particular programming language in the environment of which the algorithm will be visualized. Note that in the case of a two-dimensional construction it occurs by points (pixels), which significantly increases the time needed to execute the program implementing a particular method and therefore requires large resources of computer technology.

In the CIS countries, in the programs of the beginning of the twenty-first century of professional training of computer science teachers in universities, the purpose of teaching computer graphics was to systematically familiarize students with the mathematical and algorithmic foundations of computer graphics, the consideration of fundamental mathematical methods, principles of rotation, the transfer and change of scale of geometric shapes, methods of constructing images using axonometric and perspective methods, methods of describing curves in explicit and parametric form on the plane and in space, algorithms for constructing segments and circles, completing solid areas, truncation, the removal of hidden lines and surfaces, and the construction of realistic images (Koneva, 2010).

As a result of the study of computer graphics, future teachers of computer science should be able to formulate solutions for solving graphic tasks, choose the method of implementation, make algorithm solutions, and write a program for the formulated algorithm.

Line algorithmization and programming is an invariant component of courses of computer science, regardless of the development of information technology, and is studied in middle school, and in some cases in primary school. In primary schools a system of performers is used, which, as a rule, allows solving algorithmic tasks by graphical methods. Learning high-level programming languages is impossible without studying graphics. Essentially, each high-level programming language has the ability to support a graphics mode, built-in procedures and functions, modules and libraries, etc. For a deeper understanding of the principles of working in a graphical mode, knowledge of algorithms and methods of two-dimensional and three-dimensional computer graphics is necessary. In turn, the study of computer graphics algorithms requires the implementation of the algorithm in one of the programming languages. When studying the graphics section of high-level programming languages, such tasks as demonstrating the movement of a graphical object (point, line, ball, etc.) on the screen, text output in graphical mode, plotting a function, building visible and invisible lines and others are usually considered.

An analysis of graphic objects formulation and applying the practical skills needed for solving a range of tasks of two-dimensional and three-dimensional graphics allows the authors of this article to highlight the general stages of solving these tasks using programming in high-level computer languages:

1. Initialization of the graphics mode.
2. Defining the input and output parameters.
3. Constructing the coordinate axes (the x axis and the y axis, and in the case of the three-dimensional task, the z axis).
4. Formulating a description of the algorithm for the graphical object construction using the functions and procedures of the computer programming language.
5. Debugging the program.
6. Viewing the resulting graphic object.
7. Testing the program with various parameters (including taking "critical" points).
8. Disabling the graphics mode.

It should be noted that the solution of this range of tasks, algorithms and methods of computer graphics has an invariant theoretical fundamental nature regarding the algorithm and

method for solving two-dimensional or three-dimensional graphics; the implementation of the algorithm or method in a particular graphical editor has a practical dynamic nature.

4.1. Solving two-dimensional tasks

A two-dimensional scene can be formed from objects located both in the coordinate plane and from the projections of all visible objects on the image plane. Computer graphics are replete with examples of flat scenes, such as digital photos, scanned drawings and paintings, and images created by raster or vector graphics editors. Examples of vector graphics are drawings, motion patterns, geographical maps, function graphs, and construction of cross sections of geometric figures. Studying the same material in different of the course enriches the subjects and fills them with a specific meaning.

As mentioned above, knowledge of algorithms and methods of two-dimensional computer graphics is necessary for a deeper understanding of the principles of work in graphic mode. Also, the study of algorithms and methods of creating machine graphics involves knowledge of at least one of the computer programming languages. In the programming environment, we operate "blindly" with certain functions of the graphics mode, without delving into their mathematical essence.

Let's consider an example of solving a two-dimensional task in the programming environment.

Task 4.1. Simulation in the graphic screen of a ball's movement.

Solution: To simplify the solution of the task, we will assume that the ball is initially located in the center of the screen, and after receiving a push it will move in a given direction (Burin, 2000). We assume that the ball moves at a constant speed, and when it hits the border of the graphic screen it is reflected, and the angle of reflection is equal to the angle of incidence on the border. The contact of the ball with the boundary occurs without friction. The position of the ball will be determined by the coordinates of the center of the ball. That is, the movement of the ball is fully described by the movement of its center in the rectangle of the graphic screen. The movements of the ball will be infinite unless the "Enter" key is pressed.

The algorithm for modeling the movement of the ball is as follows:

1. Initialize the graphics mode.
2. Define the input and output parameters.
3. Build a rectangular area to limit the movement of the ball.
4. Set an infinite loop before pressing the "Enter" key. Erase the image of the ball in the current position. Calculate the coordinates of the new position of the center of the ball. Bring the ball in the direction of reflection. Hold the screen to view the results of the algorithm.
5. Debug the task.
6. View the obtained graphic object.
7. Test the program with various parameters (including taking "critical" points).
8. Disable the graphics mode.

High-level computer languages display images on the graphics screen only in the form of raster graphics and to complete these tasks, the programmer must have a mathematical foundation. Understanding how to construct a program to the geometric task of construction with the help of programming language on the graphic screen allows us to look at the possibilities of using the graphics editor in a different way.

A graphic editor includes a set of tools, each of which is either a graphic primitive or an automation of some kind of computer graphics algorithm. For example, there are algorithms for filling an area and deleting invisible lines. To construct the same graphic object in different graphic editors, regardless of how the image is specified (raster or vector), it is enough to study the environment of the graphic editor and know the computer graphics' algorithm for the construction of the type of object desired. The implementation method of the algorithm will depend on the type of graphic editor (raster or vector) and on the visualization environment.

Applying the practical skills of solving a range of tasks of two-dimensional graphics in various graphic editors allows the authors of this article to highlight the general stages of the solution:

1. Studying the interface of the graphic editor.
2. Analysis of the original graphic object of the task solution (decomposition of the graphic object, selection of graphic primitives).

3. Creating a visual description of the algorithm need to construct a graphic object using the tools and features of the graphic editor.

4. Construction of a graphical object.

5. Comparison of the graphic object created with the original specifications.

6. Saving the results of the task.

Next, we will consider an example of solving a graphical task in a graphical editor.

Task 4.2. Constructing a picture of a house in a graphic editor

Solution: To simplify the solution of the task, we will assume that the real object "house" is one-story, has two windows, a roof, a door, and a porch. This task is the simplest task available for students of any level, does not require knowledge of special algorithms, and is often used as the simplest example by the authors of both school textbooks of computer studies and the educational and methodical literature on graphic editors. To study this task, it is enough for the student to draw the desired image of the house on a piece of paper. The graphic object "house" consists of the following graphic primitives: a rectangle for the base of the house, a triangle for the roof, two small rectangles for the windows, and a curve to represent the porch.

The algorithm for building a "house" is as follows.

1. Run the graphics software.

2. Study the interface of the graphic editor.

3. Break up the original graphic object into the graphic primitives.

4. Construct a picture of the "house." Select the Rectangle tool, then the Triangle, then double Rectangle, and finally the Curve tool (Bézier curve).

5. Compare the image created with the original drawing.

6. Save the image.

This kind of approach to solving tasks with the help of a graphics editor allows learning that does not to depend on the installation and study of all graphics editors; it is sufficient to install any raster editor and any vector editor. This algorithm for constructing a two-dimensional object can also be implemented in a three-dimensional modeling environment.

Within the framework of the computer graphics course, the future teachers of computer science will have previously studied the Bresenham algorithm, the floating horizon algorithm, and the Roberts algorithm. The Bresenham algorithm is one of the oldest algorithms used to construct lines on a computer screen. This algorithm is applicable to the construction of a segment, a line, a circle, and an ellipse.

An analysis of the content of modern computer graphics courses reveals that a number of modern algorithms and methods of raster graphics are not represented in this content and, accordingly, a range of tasks of computer graphics remains unexplored. Therefore, it is necessary to study the following algorithms and methods of raster two-dimensional graphics:

• output algorithms of lines: the Bresenham algorithm, Castle-Pitway algorithm, and DDA-lines algorithm;

• methods for the rasterization of Bézier curves: the direct method and splitting method;

• algorithms for clipping segments: the Cohen-Sutherland algorithm, midpoint algorithm, Cyrus-Beck algorithm, and Liang-Barsky algorithm;

• polygon clipping algorithms;

• polygon rasterization algorithms: an algorithm that partitions polygons by frequency of occurrences, a median section algorithm, and algorithms using clustering methods;

• pseudo-toning algorithms: a simple halftone approximation algorithm, an ordered blur algorithm, a Floyd-Steinberg error dispersion algorithm;

• a quantization algorithm;

• an algorithm of equal separation of color space,

• an algorithm for image compression without a loss: a length coding algorithm, dictionary algorithms, the Hoffman statistical coding algorithm, an arithmetic coding algorithm;

• an algorithm of image compression with a loss: a JPEG compression algorithm, and a fractal compression algorithm.

In this paper, we will not dwell on the description of the mathematical component of each of the above methods due to the fact that the description of the algorithms of raster two-dimensional graphics and their software implementation are widely presented in educational and methodical

literature. We make a point that the knowledge and ability to apply the algorithms and methods listed above is a prerequisite for the implementation of the fundamental training of future computer science teachers in computer graphics. The result of achieving a good mastery of these methods is to obtain the desired graphic image in one of the programming environments. The stages of solving these tasks using high-level programming languages are similar to those listed above.

Task 4.3. Visualization of geometric objects (parabola, hyperbola, ellipse, circle) using the tool "Spline"

Solution: When working with the Bézier curve tool, it is necessary to set the control points of the curve. When constructing a spline Bézier curve, the nodes and tangent vectors at the nodes are specified. In some graphics systems, a Bézier spline is also called a Bézier curve. The construction of a curve using the Bézier curve tool and a non-uniform rational basis spline (NURBS) is a sequence of work cycles affecting each of the curve nodes. The cycle begins by positioning the instrument pointer at a specific point and the subsequent location of the future node.

Briefly, the procedure for constructing a curvilinear segment includes the following steps:

1. Determine the position of the nodes and control vertices using the mouse cursor and by pressing the left mouse button.

2. Move the nodes or control vertices to set the desired curve shape.

3. Finish the drawing.

4. Save the result.

4.2. Three-dimensional task solution

The study of algorithms and methods of two-dimensional and three-dimensional graphics in the system of higher education in the CIS countries is carried out step by step from two-dimensional graphics to three-dimensional graphics.

An analysis of the content of modern courses related to computer graphics also shows that a number of modern algorithms and three-dimensional graphics methods are not included in this course content and, therefore, the range of tasks for the students to accomplish is rather narrow. To correct this, it is necessary for students to study the following algorithms and methods of creating three-dimensional graphics:

• hidden surface and hidden line removal algorithms: the re-bounded type method, the Z-buffer method, the removal of non-facial boundaries of a polyhedron, the Roberts algorithm, the Warnok algorithm, and the Weiler-Azerton algorithm;

• an algorithm for the line-by-line scanning for curvilinear surfaces, and an algorithm for the binary partitioning of space;

• priority methods: the artist method and the floating horizon method;

• scanline algorithms for curved surfaces;

• a method for the binary partitioning of space;

• an algorithm for determining the visible surfaces using ray tracing;

• lighting algorithms: the Gouraud, Blinn-Phong, and Cook-Torrens shading models; and

• the radiosity method.

These algorithms and methods of three-dimensional graphics are also studied in detail in computer graphics courses and are described in detail in educational and methodical literature; therefore, we will not discuss them here.

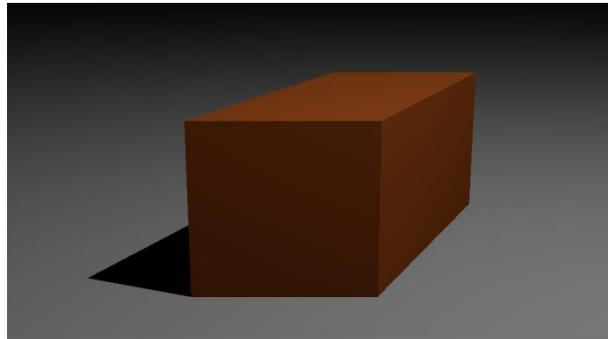Let us consider an example of solving a three-dimensional task.

Task 4.4. Modeling an image of a cube

Solution: there are several algorithms used for removing invisible parts of an object. Among them are the painter's algorithm and the Z-buffer algorithm. We will restrict ourselves to the simple case of image construction of a cube. A cube is useful for analysis because it consists of six square faces. In computer graphics, even complex surfaces are replaced by similar surfaces assembled from flat polygonal faces. Let's assume that each face of the cube contains five "reference" points, four vertices and the center. At the central point, we will construct a normal line which is perpendicular to the surface. If we assume that the normal line (in our case) is directed from the object, then in the view coordinate system the projection of the normal line is on the *z* axis, which we will consider is going on the plane of the screen and the observer; it will be positive in the case when the face is visible to the observer, and negative if it is not visible. So we get a fairly

simple algorithm for determining the visible and invisible faces. This algorithm can be applied in a general case, the case of a surface assembled from polygons. In order to construct a normal line it is necessary to make an assumption: if we assume that two sides of the polygon coming from one common vertex are vectors, then their vector product can be taken as a normal line to the surface of this polygon. The direction of the normal line will be uniquely determined if the order of vector factors is given.
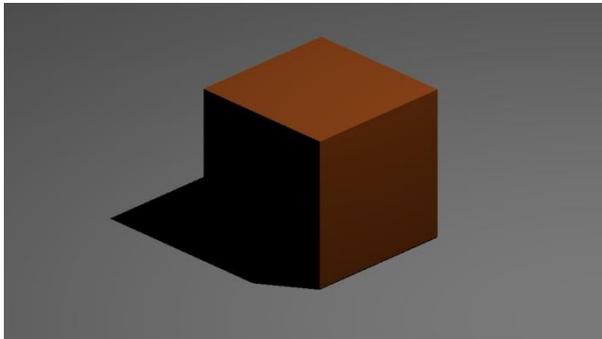
(a) Isometric projection.   (b) Oblique projection.

Fig. 3. Three-dimensional modeling of a cube using a visual programming environment

Computer graphics, by its definition, are used in important aspects of modern society such as 3D printers, 3D printing and 3D modeling, and it would be natural that these concepts and basic skills are formed in this area, and also that future computer science teachers' knowledge is deepened and their skills developed at a highly professional level (Nazarbayev, 2018). Therefore, speaking of content revision of the educational program in computer graphics in schools and in institutions of higher education, it is necessary to pay attention to the study these issues, which are becoming fundamental. In some schools there are pilot projects teaching three-dimensional graphics; the latest changes in the content of teaching computer science in the Nazarbayev Intellectual School now touches on the teaching of 3D modeling.

3D modeling is a process of developing a mathematical representation of any three-dimensional surface of an object using specialized software. The modern software market offers a wide range of graphics programs for drawing and modeling.

Today, the following 3D modeling tools are widely accepted (Farin, 2002):
- spline modeling: the Bézier curve, *B*-spline and NURBS (Piegl et al., 2012);
- polygon modeling: polygon mesh, tessellation method;
- using surface subdivisions; and
- procedural modeling.

Three-dimensional graphics include a set of methods and tools that allow the creation of three-dimensional objects using free-form surfaces, polygon meshes and primitive solids. Examples of such objects include the objects of the construction industry, the gaming industry, and printing. Building a geometrical projection of a three-dimensional model is carried out with the help of specialized graphics programs. The model created in these environments simulates the corresponding objects in the real world. Three-dimensional graphics deal with the representation of a three-dimensional object on a two-dimensional plane, hence a number of difficult mathematical transformations are needed to build a realistic model. Here are the main stages of building a 3D model:
1. Creating a reference (drawing the sketch of an original modeling object).
2. Geometric modeling (based on the reference).
3. Create a sweep.
4. Render the texture.
5. Adjust the texture, refraction, reflection and transparency.
6. Compare the resulting model with the original reference.

3D modeling implies the presence of several 3D models on a three-dimensional scene, which significantly complicates the process of modeling in a particular task. Next, we highlight the main stages of 3D modeling:

1. Modeling (creating a three-dimensional mathematical model of the scene, creating 3D-models).
2. Texturing (defining surface texture models).
3. Lighting (choice of light source).
4. Animation (movement of models).
5. Dynamic simulation (calculation of model interaction).
6. Rendering (visualizing models, building a model projection).
7. Compositing (model layout, refinement).

Determining the surfaces of a 3D texture model is not an easy task. The following algorithms are used to accomplish it:

- texture filtering algorithm;
- mipmapping;
- algorithm of point texturing;
- bilinear filtering algorithm;
- trilinear filtering algorithm; and
- anisotropic filtering algorithm.

Quite a complex mathematical calculation requires the stage of visualization (rendering), as at this stage there is a transformation of three-dimensional vector model to two-dimensional raster model. The following rendering algorithms are used:

- Z-buffer algorithm;
- scanline algorithm;
- ray tracing algorithm; and
- global illumination algorithm.

Static 3D models are of little interest; they become valuable in interaction with other models in the process of animation. Animation (in this case computer animation) is one of the most important stages of 3D modeling. Computer animation is produced by computer graphics methods and algorithms of raster and vector graphics, and two-dimensional and three-dimensional graphics.

2D animation allows the creation of moving objects or two-dimensional scenes. The previously reviewed Task 4.1 "Simulating a ball movement in a graphic screen" is a kind of example of programmable two-dimensional animation.

Below, we list the main steps of creating a computer animation:

- animation on key frames;
- recording (capturing) the movement;
- procedural animation;
- shape animation; and
- programmable animation.

From the point of view of the defining the essential training needed in computer graphics, the most interesting part of the course is the programmable animation, which is based on the algorithms of raster and vector graphics. The stages of creating a programmable animation are similar to the stages of creating two-dimensional and three-dimensional graphics using programming in high-level computer languages.

Let us now consider an example of a programmable animation.

Task 4.5. Animation of a rotating sphere

Solution: The illusion of a moving image is created by viewing a sequence of frames showing different phases of motion. To display a dynamic image, one must program the output of a sequence of frames. The frame rate should be fast enough to look realistic. This, in fact, is the main task in the programming of dynamic images. It is not easy to solve this task if each frame contains a complex image consisting of a large number of graphic elements, and even its own attributes, such as lighting conditions, the nature of the reflecting surface, etc.

The easiest way to include animation in a program is to simply redraw the graphic screen. This is the method used to solve the task. It displays images of a rotating a wireframe model of the sphere on the screen. For this purpose, two types of data are entered, the first one is for the storage

of the world coordinates of a point, and the second is for the screen coordinates. The array contains the world coordinates of the points intersection of the "meridians" and "parallels" of the sphere. In addition, a supportive procedure for converting spherical coordinates into rectangular coordinates is used. We have already used the spherical coordinate system; in it, the position of a point is set by the length of the radius vector and two angles. The procedures perform a rotation conversion at given angles around each of the x, y, and z axes. In the first case, the rotation is performed for a single point, and in the second case for the sphere as a whole (that is, for all its reference points). The next frame is then displayed on the screen, and at this time the next frame is built on the invisible page. Then the pages switch roles.
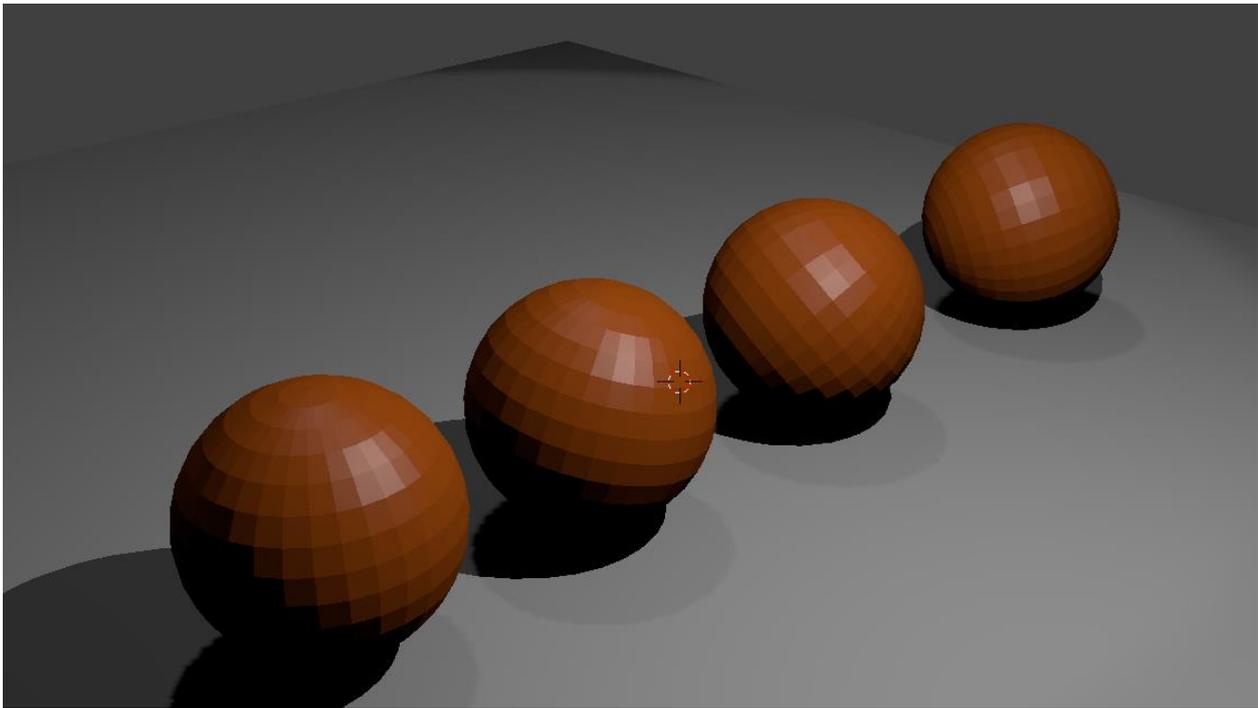

Fig. 4. Rotation of a sphere

3D animation allows the creation of 3D models and scenes. High-quality 3D animation allows viewers to visualize real objects and phenomena, and also nonexistent ones, to visually examine and understand complex processes. The scope of application of 3D modeling and animation covers almost all areas of work and is related to multimedia technology. At first glance, the list of stages of creating 3D animations looks quite simple, but each of them requires creative abilities and ideas from the animator-developer in addition to fundamental knowledge and the skills to program or work with animation software. We will list the stages of creating 3D-animation:

1. Developing creative ideas.
2. Writing a script and storyboard.
3. Creating 3D models.
4. Animating the 3D models.
5. Compositing or mixing frames into a single video clip.
6. Stacking the sound (voice overlay and voice acting).
7. Creating the titles.

The current process of three-dimensional modeling and animation is the culmination of the work in the field of computer graphics. It is quite complex and requires knowledge of the algorithms and methods that were discussed at the beginning of this section.

5. Methods for solving the tasks of developing pedagogical tools

Studying the practical experience of training future computer science teachers, as well as future subject teachers, in the field of information and communication technologies shows that such training is being carried out at all stages of the continuing education system in such

disciplines as computer science, the increasing use of computers in education, digitalization of education, information and communication technologies in education, pedagogical technologies in education, informatization of education and learning problems. Within the framework of this training, the following pedagogical tasks are being completed (Koneva, 2008):

• The development of pedagogical tools using a text editor: using text editor capabilities to create forms, using text editor tables to create forms, preparing printed materials in a text editor, developing graphic elements in a text editor, preparing didactic materials in a text editor, creating printed advertising products in a text editor.

• The development of teaching tools using spreadsheets: the accounting of material and technical equipment in a spreadsheet, the development of an electronic journal in a spreadsheet, the automation of monitoring progress by using a spreadsheet, and the use of graphics in a spreadsheet.

• Creating pedagogical multimedia presentations: presentation of the curriculum, development of educational presentations, presentation of an almanac, and the development of the teacher's electronic portfolio of teaching aids.

• The preparation and development of pedagogical tools using a desktop publishing system: preparation of printed materials for extracurricular activities, the development of electronic educational materials with the help of a desktop publishing system.

Thus, it follows that a number of pedagogical tasks will require future teachers to know and understand the field of computer graphics. These tasks include:

• Tasks focused on the use of graphic capabilities of a text editor: the preparation of printed materials, development of graphic elements, preparation of didactic materials, and creating printed advertising materials.

• Tasks focused on the use of graphic capabilities of the table processor: a graphical representation of the results of monitoring performance, and the use of elements of business graphics.

• Tasks focused on the use of multimedia presentations: the presentation of the curriculum, and the development of educational presentations, a presentation of the almanac, and the development of an electronic portfolio of teaching aids.

• Tasks focused on the use of desktop publishing: the development of a teacher's electronic portfolio, the preparation of printed materials for extracurricular activities, and the development of an electronic syllabus.

Thus, at the end of postgraduate education, the future teacher of computer science should have the necessary knowledge, skills and abilities in the field of creating graphic objects in a text editor (images, diagrams, inscriptions), should be able to build charts and graphs in a spreadsheet, to create and use graphic elements for the monitoring and visual presentation of results, to design and create educational multimedia presentations, and prepare printed materials for extracurricular activities (Koneva et al., 2018).

An analysis of the professional activities of not only the future computer studies teacher, but also of teachers in general, in terms of his or her use of methods, technologies and information tools, reveals that the teacher in their daily pedagogical activities encounters visualization tools, and elements of a pedagogical experiment: self-analysis of a lesson, and a lesson on the same topic in different classes or groups conducted in different ways; an analysis of performance carried out by topic, section, class or group, terms or semesters, etc. Thus, the future teacher of computer studies should have the skills needed to conduct and analyze both research and experimental teaching activities, including a pedagogical experiment (Koneva et al., 2012).

The issues of training and organizing a pedagogical experiment are examined in pedagogy courses, information technologies in science and education, organizing and conducting pedagogical research; and, as a result, the knowledge and skills gained manifest themselves in the processes of pedagogical practice, research, and experimental pedagogical activities. But as education becomes more computerized, this knowledge alone is not enough. Knowledge is also required in from the following areas:

• information technologies: skills and abilities to use a text editor, spreadsheet, presentation development tools, etc.;

• the increased use of computers in education: methods of accessing, sharing and creating scientific and methodological research, and methods of computer-based control and measurement of learning outcomes.

But, unfortunately, for a competent presentation of the results of a teacher's research activity, including a computer science teacher, having a general knowledge and understanding of the concepts is not enough. It is also necessary to teach teachers how to accomplish the tasks of research and create experimental-pedagogical content. This includes the following tasks:

• presenting the results of research activities and pedagogical experiments with the help of a text editor using a text editor to prepare a publication, a research report, a dissertation, an abstract, and to work in teams on research using a text editor;

• processing the results of research activities and pedagogical experiments using spreadsheets: the automated monitoring of progress using a spreadsheet, processing the results of research and a pedagogical experiments, using elements of business graphics for a visual presentation of the research results in a spreadsheet.

• simulating the research activities results and pedagogical experiment with the help of multimedia presentations: Simulation of the research work plan in the form of a presentation, development of an electronic report in the form of multimedia presentation, simulation by a research almanac in the form of a presentation, creating an electronic portfolio based on the results of research and pedagogical experiment.

• preparing and simulation of the research work's results and pedagogical experiments with the help of a desktop publishing system: the development of the corporate style of a teacher-researcher, preparation of printed products of scientific content, and the development of an electronic resource based on the results of research activities and a pedagogical experiment.

Computer graphics as a field of computer science and information technology allows users to fully disclose the results of the tasks listed above.

From the computer graphics point of view, a text editor belongs to the desktop publishing system, it allows the user to prepare printed products such as a scientific article, scientific report, dissertation, and abstract. Actually, the desktop publishing system allows a researcher to convey information about their research and teaching activities in the form of a booklet, an abstract in the form of a brochure, or the development of an electronic resource based on the results of research work and pedagogical experiments.

The development of business graphics takes the automation of processing results of experimental and pedagogical work process to a higher level at all stages of the pedagogical experiment. The most optimal form of using graphics in these conditions is to depict the statistical data from ascertaining, testing and forming experiments, which are most easily analyzed, visualized and processed and thus allows viewers to see the dynamics of the experimental and control groups, and to conduct a comparative analysis of the results.

The effectiveness of such processes depends largely on how a future computer science teacher is able to use graphics tools to carry out the functions of statistical processing of experimental results, controls, and monitoring. The use of graphics creates conditions for more efficient analysis of research results using processes related to information processing. Processing the results of an experiment using graphics can significantly reduce the time and labor costs of analysis and pedagogical monitoring, and at the same time using graphics significantly increases the information content of the results. The graphics system allows viewers to observe the dynamics of information indicators. Results of the analysis of an experiment can be presented in the form of tables, graphs and charts. Charts, and in particular graphs, are elements of business graphics. Almost all modern spreadsheets, word processors, and presentation development tools have built-in business graphics. To produce these types of graphics, there is a graphical mode in which you can build charts of different types, which shows numerical dependencies.

It is known that graphs and charts do not just appear by themselves; they should be based on numerical data. In a pedagogical experiment, such data is the data of pedagogical monitoring. The effectiveness of a teacher's assessment and evaluation activities depends largely on how well he or she implements all the basic functions of monitoring each student's learning results, and how well the teacher is able to use information technology tools to perform the functions of control and monitoring.

The information control function is a systematic recording of learning results, which allows one to judge the performance of each student and his or her achievements and shortcomings in their educational work, which is very important at the evaluation stage.

The control adjusting function provides feedback between the teacher and student, which is necessary to for the teacher to make adjustments to the teaching methodology used, and in the further course of pedagogical experimental work, the control function can affect the redistribution of educational time between the various issues of the topic caused by the student's deficiencies in knowledge, which is important at the stage of verifying the experiment.

And finally, it is very important for teachers to be able to competently present the results of their research and teaching activities and to show the results of their experimental work; it must be clear to everyone what the researcher was doing, what was researched, how, and what the results were . Also, no one wants to listen to long speeches or try to memorize numbers given in an oral presentation. Therefore, an electronic report should be presented, which will combine both the elements of an abstract and the results of the pedagogical monitoring and allow everything to be presented clearly and concisely.

Let us now consider in detail an example of automating the monitoring of student progress using graphics tools.

Task 5: Construction of an attendance report and the students' progress on a specific topic on a particular subject using graphics elements (histogram, schedule)

Solution:

1. Make the necessary calculations. For example, you can insert Microsoft Excel's SUM or AVERAGE function, or even develop your own functions.

2. Select the information you need to create a schedule. If you want to select information that is not in adjacent cells, select one group of cells, hold down the Ctrl key, and then select another group of cells.

3. Create a histogram with a series of columns. To create graphs, you need to launch the diagram wizard, which will help you to select the type of graph, give the name of the graph, label the axes, specify the data on the diagram, etc. Use the title of the table as the title of the graph. If desired, enter titles for the x and y axes. Save the graph as a new sheet. Remove excess information. If necessary, refer to the section How to create a bar chart or histogram for a comparison of values and values of the program's reference guide.

4. Copy the sheet with the histogram. Move the copied graph to the end of the document.

5. Return to the second graph in the editing table and convert it to a graph with icons, formatting the columns with the image you copied. If you use OpenOffice Calc, format the columns with pre-rendered graphics.

6. Rename all the tables and graphs and delete unnecessary sheets.

7. Arrange the sheets in such a way that the first is a table, then a histogram, and then a graph with icons.

8. Save your work.

Thus, to organize and conduct research and experimental-pedagogical activity, the teacher-researcher must possess the knowledge, skills and abilities to provide reports in the form of articles, reports, scientific reports, theses and other documentation in a word processor for performing the statistical processing of scientific and pedagogical information and to demonstrate the results of an experiment in the form of a report and/or presentation.

The main requirements for a future computer science teacher to organize and conduct a pedagogical experiment are the knowledge and skills to work with printed materials, use of graphics for pedagogical monitoring and proof of the pedagogical research hypothesis, and to be able to create pedagogical multimedia presentations.

Therefore, the priority areas for teaching computer graphics to future computer science teachers are the creation of printed materials, graphics, and multimedia presentations.

6. Conclusion

In the context of teaching fundamental computer graphics to future teachers, the algorithms and procedures needed for solving graphic tasks are not limited to the examples listed above. It is clear that computer science teachers must have the requisite knowledge and understanding of fundamental approaches to the methods of solving computer graphics tasks. This will contribute

not only to the teacher's professional ability and efficacy despite the constant improvements to computer technology, but also to the teacher's ability to teach using the latest technical tools and techniques. This knowledge will further develop a teacher's interest and desire to prepare young members of society who will live and work in constantly changing conditions.

By increasing computer science teachers' fundamental knowledge and understanding of computer graphics, the students' related studies of computational geometry, drawing, and computer graphics will allow students to create and evaluate images of various graphic objects.

After analyzing different methods of solving tasks in computer graphics and defining the principles of their classification, we can conclude the following: the systematization of tasks in computer graphics makes it possible to effectively and correctly use different methods of solving tasks at different stages of the educational process, and to develop the students' abilities to work with computer graphics tools of different levels, and to create conditions that motivate and sustain the students' interest in computer graphics, thus enhancing their independent computer activities and improving the quality of graphics training in the context of fundamental computer science education. The future computer science teacher needs to be competent in the field of information and communication technology, in working with graphic editors, in demonstrating algorithms of raster and vector graphics, in using the technical language of computer science, and in working out the methodology for choosing and using knowledge and skills in their future professional activities.

The proposals described to improve fundamental training of computer science teachers are not limited to the improvement of the disciplines mentioned above. Of course, the main aspects of improving the fundamental training of teachers in higher education institutions are reflected in other courses of the curriculum for training future teachers of computer science.

References

Abishev et al., 2016 – *Abishev, N., Bidaibekov, E., Dalinger, V., Knyazyev, O.* (2016). Higher education in Russia and Kazakhstan in modern conditions. *Rupkatha Journal on Interdisciplinary Studies in Humanities,* 8(2), 117-127.

Alexandrov, 1954 – *Alexandrov, I.I.* (1954). Collection of geometric construction problems with solutions, 12th edition. Moscow: Uchpedgiz, 176 p. [in Russian]

Baranova, 2000 – *Baranova, L.N.* (2000). Geometric problems of construction in the middle school, Ph.D. thesis, 190 p. [in Russian]

Bidaibekov et al., 2017 – *Bidaibekov, E., Kamalova, G., Bostanov, B., Salgozha,* I. (2017). Development of information technology competency in students during their training in Al-Farabi's geometric heritage within the framework of supplementary school education. *European Journal of Contemporary Education*, 6(3), 479-496.

Bidaibekov et al., 2017 – *Bidaibekov, E.I., Grinshkun, V.V., Koneva, S.N., Baidrakhmanova, G.A.* (2017). Aspects of teaching computer graphics teachers regarding the increased use of computers in of education, modern information technologies and IT education, 13(2), 229. [in Russian]

Bidaibekov, Kamalova, 2010 – *Bidaibekov, E.I., Kamalova, G.B.* (2010). On the issue of fundamental teacher training in the field of computer science and increased use of computers in education, Bulletin of KazNPU named after Abay. Physics and Mathematics Series, 80-87. [Electronic resource]. URL: http://kaznpu.kz/docs/ vestnik/fizika_matematika/129_2010.pdf (accessed 25.09.2018). [in Russian]

Bidaibekov et al., 2017 – *Bidaibekov, E.I., Koneva, S.N., Baidrakhmanova, G.A.* (2017). On a classification problem of computer graphics in the fundamental conditions of education. The computerization of Education: Theory and Practice, an International Conference. (Omsk, 17-18 November 2017), Ed. M.P. Lapchik, Omsk: OmGPU Publishing House, 420 p. [in Russian]

Burin, 2000 – *Burin, E.A.* (2000). Programming in the Turbo-Pascal language, AGU, Almaty, 398 p. [in Russian]

Farin, 2002 – *Farin, G.* (2002). Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann.

Koneva, 2008 – *Koneva, S.N.* (2008). Computer graphics as a component of training research teachers. *Bulletin of KazNPU. Physics and Mathematics Series,* 4(24), 69-75. [in Russian]

Koneva, 2010 – *Koneva, S.N.* (2010). Aspects of computer graphics training of students with physical and mathematical majors training as teachers, Kasipkeri-Professional of Kazakhstan, 6(85), 24-25. [in Russian]

Koneva, 2010 – *Koneva, S.N.* (2010). The peculiarities of teaching computer graphics to physical and mathematical specialists taking Education majors in Kazakhstan, Kasipkeri-Professional of Kazakhstan, 6(85), 24-25. [in Russian]

Koneva, 2011 – *Koneva, S.N.* (2011). Creating educational tools using information technology tools. Almaty: KazNPU named after Abay, 40 p. [in Russian]

Koneva, Baidrakhmanova, 2018 – *Koneva S.N., Baidrakhmanova G.A.* (2018). Teaching computer graphics as a component of the system of training teacher-researchers: *Proceedings of the scientific-practical conference dedicated to the 30th anniversary of teaching computer science in school, "From computer science at school to the technosphere of education",* Moscow: MGPU [Electronic resource]. URL: http://www.it-school.ucoz.org/publ/doklady/obuchenie_ kompjuternoj_grafike_kak _/1-1-0-237 (accessed 25.09.2018). [in Russian]

Koneva, Nurpeisova 2012 – *Koneva S.N., Nurpeisova D.M.* (2012). Requirements for the teacher-researcher of computer graphics, Higher School of Kazakhstan, 3, 232-235. [in Russian]

Koneva, Nurpeisova, 2012 – *Koneva S.N., Nurpeisova, D.M.* (2012). Requirements for a research teacher in the field of computer graphics, Poisk, 35-40. [in Russian]

Kubesov, Al-Farabi, 1972 – *Kubesov A.* (1972). Al-Farabi. Mathematical treatise, Publishing House "Science" of the Kazakh SSR, Alma-Ata. [in Russian]

Makarova, 2010 – *Makarova, N.V.* (2000). Informatics for grades 6-7, St. Petersburg: Piter Publishers, 256 p. [in Russian]

Nazarbayev, 2018 – *Nazarbayev, N.* (2018). The message of the president of the Republic of Kazakhstan, Nursultan Nazarbayev, to the people of the Republic of Kazakhstan [Electronic resource]. URL: https://www.mod.gov.kz/rus/poslaniya_prezidenta?cid=0&rid=4451 (accessed 25.09.2018). [in Russian]

Olifer, 1955 – *Olifer, G.M.* (1955). Basic principles of teaching methods for building of tasks in the middle school in the light of the tasks of polytechnic training, PhD thesis. [in Russian]

Piegl, Tiller, 2012 – *Piegl, L., Tiller, W.* (2012). The NURBS book. Springer Science & Business Media.

Shvetsky, 1994 – *Shvetsky M.V.* (1994). Methodological system for the fundamental training of future teachers of computer science in higher education institutions undergoing a two-stage education, D.Sci. thesis, St. Petersburg. 480 p. [in Russian]