

**Work program on the module “Programming”
(34 hours for 8-10 years)**

Pavlodar, 2018

APPROVED BY

Vice-Rector for Academic work

c.e.s., docent _____A.S. Kadyrova

“ ____ ” _____ 2018 y.

Compiled by : Head of the department of Information technology at INEU,

c.p.s., docent Asainova A.Z. _____,

Head of the department of organization and support of the educational process of the INEU,

doctor PhD Abykenova D.B. _____

Teacher of information technology of the first qualification category of school №25

Shakirova A.S. _____

Training program for the module “Programming” for students (8-10 years)

For the discipline “Programming in the Scratch”

according “Programming” module

for the basic course for students (8-10years)

Developed on the basis of the “IT-classes for children” project

for 2018-2020 of the “Nur Otan” party.

EXPLANATORY NOTE

Today science and technology are developing so rapidly that education often does not keep up with them. One of the most famous informal way of organizing extracurricular educational activities is the method of project. The most suitable tool for organizing such activities is the Scratch environment. Scratch – is the newest environment, which allow children to create their own interactive and animated stories, presentations, models, games and another compositions. It is easier to be successful in such surrounding. Scratch could be considered as a tool for creativity, leaving programming in the background. Pupils could compose stories, paint and liven up their invented characters on the screen, study to work with graphics and sound. There are many applications to the possibilities of Scratch: in this medium animated postcards, mini-games, cartoons. As a result of the execution of simple commands, a complex model can be formed, in which a set of objects with different properties will interact.

When students create projects in Scratch, they learn many 21st century skills that will be necessary for success:

- creative thinking,
- clear communication
- system analysis,
- fluent use of technology
- effective interaction,
- design,
- the ability to learn and self learn,
- independent decision making.

Studying Scratch can seriously help students learn the basics of algorithmization and programming, and the gained knowledge will be useful for further and more serious study of programming. Work in the Scratch environment is carried out in the same way as a means of preparing students for various competitions and exhibitions on this subject, which are currently gaining momentum.

The study of this course is aimed at achieving the following goals:

- formation of general educational skills and abilities based on the tools and methods of informatics and ICT, including mastering the skills to work with various types of information, independently plan and implement individual and collective information activities, present and evaluate its results;
- propaedeutic study of the concepts of the basic course of school informatics;
- fostering a responsible and selective attitude to information; development of cognitive, intellectual and creative abilities of students,
- development of the ability to solve creative problems,
- development of independence.

The key to the program is the concept of “project scientific-cognitive activity of a schoolchild”, both joint (with other subjects) and independent activity using scientific research methods, the leading motive of which is cognitive interest. Project scientific-cognitive activity is not an end in itself, but is considered as an environment in which the student’s personal potential is most naturally revealed.

In this regard, the objectives of the project student's scientific and educational activities are:

- development of intellectual, educational and creative abilities of the student;
- development of interdisciplinary skills (personal, cognitive, communicative, regulatory);
- development of ways of thinking;
- formation of a complete picture of the world and system thinking based on interdisciplinary connections.

It should be borne in mind that the age characteristics of the younger student do not allow to fully implement the full-fledged scientific research. At the same time, the early inclusion in a specially organized project activity of a creative nature allows the student to form cognitive interest and research skills, which at an older age will be useful to them to carry out scientific and educational projects.

The main objective of the course is to contribute to the formation of information and functional competence in students, the development of algorithmic thinking. The purpose of the course is to help children learn the basic capabilities of the computer and learn how to use it in everyday life. And:

- to give students an idea of the modern approach to the study of the real world, the wide use of algorithms and computing technology in scientific research;

- to form students' skills in computer skills as a means of solving practical problems;
- preparing students for an active, full-fledged life and work in a technologically developed society
- creation of conditions for the introduction of new information technologies in the school's educational process;
- realize the most fully increasing interest of students in the in-depth study of programming through the improvement of their algorithmic and logical thinking;
- formation of skills and skills of independent use of a computer as a means for solving practical problems.

The implementation of these tasks will contribute to the further formation of the schoolchildren's view of the world, the disclosure of the role of computer science in shaping the natural science picture of the world, the development of thinking, including the formation of an algorithmic thinking style, and the preparation of students for life in the information society.

Student training requirements

Students must:

know:

- individual methods of planning activities;
- drawing up a plan for the upcoming project in the form of a drawing, a diagram;
- drawing up a plan for the upcoming project in the form of a table of objects, their properties and interactions;
- splitting tasks into subtasks;
- on the requirements for the organization of a computer workplace, to comply with the requirements of safety and hygiene in working with ICT tools;
- computer devices;
- examples of information carriers.

be able to:

- give simple life examples of the transfer, storage and processing of information in human activities, in wildlife, society, technology;
- distinguish between computer software and hardware;
- launch programs from the Start menu;
- resize and move windows, respond to dialog boxes;
- enter information into the computer using the keyboard and mouse;
- use a text editor for typing, editing and formatting simple texts;
- use the simplest graphics editor to create and edit images;
- perform calculations using the Calculator application;
- draw up a project plan, including:
- analyze the results and draw conclusions;
- find and correct errors;
- prepare a small report on the work;
- make a public report;
- outline the future paths of the project.

Working on projects in Scratch, students have the opportunity to learn important computational concepts, such as repetitions, conditions, variables, data types, events, processes, and express themselves in computer creativity.

Goals:

- formation of information and algorithmic culture;
- formation of ideas about algorithms and models, their properties;
- Development of algorithmic thinking necessary for professional activity in modern society; development of skills to compose and record an algorithm for a specific artist; the formation of knowledge about algorithmic constructions, logical values and operations; familiarity with the basic algorithmic structures - linear, conditional and cyclic;

- the study of object-oriented and event-based programming;
- Familiarity with parallel programming technologies;
- gaining experience in the use of informational resources of the society and means of communication in educational and practical activities;
- Ability to create and maintain an individual information environment, to ensure the protection of significant information and personal information security;
- development of basic skills and abilities to use computer devices;
- formation of skills of formalization and structuring of information, the ability to choose the way of presenting data in accordance with the task;
- formation of skills and abilities of safe and expedient behavior when working with computer programs and on the Internet, the ability to comply with the norms of information ethics and law
- establishment of interdisciplinary connections in the process of project and scientific-cognitive activity

Terms of the program:

The term of the program realization-

The program is designed for students - 8-10 years old and assumes that students have skills in working with the keyboard, mouse, techniques for working with graphic images, know how to save work, the program does not require initial knowledge in the field of programming

The estimated amount of study time is ___ an hour a week.

Classes are held ___ once a week for 40 minutes. The program is designed for 34 hours.

This program uses individual, group and frontal forms of work.

The content of practical classes is focused not only on mastering the skills of programming by students, but also on preparing them as competent PC users; the formation of skills to participate in distance contests and competitions, skills to successfully use the skills of network interaction.

The current control of mastering the material is planned to be carried out by oral and written questioning in the form of various tests, including in electronic form, independent, practical and creative works; by using the game form of knowledge control in the form of puzzles, crosswords, contests.

Final control - in the form of project activities, protection and presentation of creative works.

MAIN CONTENT

1. Introduction (1 hour)

Safety Instructions. Basics of algorithms.

As a result of studying the section, students should

have an idea:

- about the concepts of safety, the fundamentals of algorithmization;

know:

- basic safety regulations;
- algorithm and its properties;
- principles of algorithm development;
- visual tools to describe the algorithm;

know:

- basic safety regulations;
- algorithm and its properties;
- principles of algorithm development;
- visual tools to describe the algorithm;

be able to:

- create algorithms;

have experience:

- work in the basics of algorithmization;

1. Familiarity with the Scratch programming environment (2 hours)

What is Scratch. Acquaintance with the interface of the program Scratch. Team system of the performer Scratch

As a result of studying the section, students should

have an idea:

- about the Scratch program;

know:

- The basic techniques for working with objects in the Scratch environment window;
- various ways to run a script or several scripts;
- technology of drawing up the script;
- distinguish between blocks;
- Scratch interface;

be able to:

- organize an individual information environment;
- work with objects of the Scratch environment;

have experience:

- work with the Scratch environment interface.

Getting Started in Scratch (2 hours)

Scene. Editing, creating and adding background from file. The concept of sprites. Adding new sprites. Drawing new objects.

As a result of studying the section, students should

have an idea:

- about the concepts of "Stage", "Sprite";

know:

- The basic methods of work creating and adding background from a file;
- The main work of the sprite;

be able to:

- edit and create a background;
- insert a standard background from the library module environment Scratch;
- draw a background from a graphic editor;
- create multiple backgrounds in one **scene**;

- upload sprites to the scene from the standard collection of the Scratch environment;
- own the main ways to create a sprite;
- to simulate actions, processes, phenomena;

have experience:

- work with the background;
- work with sprites;

1. Scratch Scripts (15 hours)

The mechanism for creating a script. Motion commands. Drawing commands Animation using motion commands and costume change. Control. Adding sound. Creating an animation using sound. Use in programs of conditional statements (branching). The functionality of the cycles. Cycle of actions depending on the conditions. Operators. The use of arithmetic and logical blocks together with control blocks. Variables. String constants and variables. String operations. Lists. Sensors. Input - output data. Free design. Creating projects according to your own design.

As a result of studying the section, students should

have an idea:

- on the concepts of "Script", animation, conditional operators, cycle, logical blocks, variables, sensors, etc.
- the concept of storyboard movement

know:

- basic drawing movement commands;
- the sequence of scripts;

be able to:

- create programs for the movement of sprites around the scene;
- create programs for drawing various shapes;
- create programs to control the appearance of the object;
- create Scratch-stories with imitation of walking and movement of objects;
- create programs with a change in the sequential execution of scripts in the presence of conditions
- create programs using cycles;
- create programs for processing user data with the output on the screen of the final result;
- create test programs on the basis of comparing data from several lists;
- work with control units

have experience:

- work with control units;
- work with conditional operators;
- work with sensors;
- create scripts.

1. Work with multiple objects. Synchronization of their work (4 hours)

Sequence and parallel execution of scripts. Interaction between sprites.

As a result of studying the section, students should

have an idea:

- about the concepts of working with multiple objects;

know:

- basic work with several objects;
- synchronize work;

be able to:

- create Scratch-stories with simultaneous and alternate work of several performers;
- create Scratch-stories with the interaction of several performers and fixed objects;
- create Scratch-stories with the interaction of several artists;
- create Scratch-stories with the interaction of several performers and fixed objects;
- create Scratch-stories with the interaction of several artists;

have experience:

- work with multiple objects.

1. Using Scratch to create mini-games (4 hours)

Free design. Creating projects according to your own design. Join the Scratch Community. Development of basic sprites for the game. Formation of basic scripts. Synchronization of scripts for different sprites. Moving from one scene to another. Creating a game interface.

As a result of studying the section, students should

have an idea:

- about developing base sprites.
- about synchronization of scripts for different sprites

know:

- how to register in the Scratch community

be able to:

- create projects according to your own plan;
- move from one scene to another;
- create interfaces for games;
- develop base sprites;

have experience:

- Work in the Scratch community;
- the creation of projects.

1. Creating a personal project (5 hours)

Project activity. Demonstration and protection of the project

As a result of studying the section, students should

have an idea:

- about project activities;

know:

- the main constructions of the programming language Scratch;
- working methods in the Scratch programming environment, text, graphic, sound editors, browsers;
- copyright;

be able to:

plan, predict, adjust their activities;

- set goals, determine the final result of the activity;
- draw up a plan of activities;
- identify the main types of information that arise in the process of solving the problem;
- select all objects of the upcoming project, their properties and interactions;
- select individual subtasks and the sequence of their implementation;
- establish causal relationships;
- to formulate the problem and independently create ways to solve it;
- Express your thought;

have experience:

- Work in the Scratch community;
- the creation of projects.

CALENDAR AND THEMATIC PLANNING

№	Name of the theme	hours	date
1. Introduction (2 h)			
1	Safety Instructions. Basics of algorithms.	1	
2	Basics of algorithms.		
2. Introduction to Scratch (2 h)			
1	What is Scratch. Acquaintance with the interface of the program Scratch.	1	
2	Team system of the performer Scratch	1	
3. Getting started in the Scratch environment (2 h)			
1	Scene. Editing, creating and adding background from file.	1	
2	The concept of sprites. Adding new sprites. Drawing new objects.	1	
4. Scripts of Scratch program (15 h)			
1	The mechanism for creating a script.	1	
2	Motion commands.	1	
3	Drawing commands	1	
4	Animation using motion commands and costume change.	1	
5	Control. Adding sound.	1	
6	Creating an animation using sound.	1	
7	Creating an animation using sound.	1	
8	Use in programs of conditional statements (branching).	1	
9	The functionality of the cycles. Cycle of actions depending on the conditions.	1	
10	Operators. The use of arithmetic and logical blocks together with control blocks.	1	
11	Variables.	1	
12	String constants and variables. String operations.	1	
13	Lists.	1	

14	Sensors. Input - output data.	1	
15	Free design. Creating projects according to your own design.	1	
5. Work with multiple objects. Synchronize their work (4 h)			
1	Sequence and parallel execution of scripts.	1	
2	Sequence and parallel execution of scripts.	1	
3	Interaction between sprites.	1	
4	Interaction between sprites.	1	
6. Using Scratch to create mini-games (4 h)			
1	Free design. Creating projects according to your own design. Join the Scratch Community.	1	
2	Development of basic sprites for the game. Formation of basic scripts.	1	
3	Synchronization of scripts for different sprites.	1	
4	Moving from one scene to another Creating a game interface.	1	
7. Creating a personal project (5 h)			
1	Project activity	1	
2	Project activity	1	
3	Project activity	1	
4	Project activity	1	
5	Demonstration and protection of the project	1	