

Краснодарский край
Муниципальное образование город Новороссийск
Муниципальное автономное общеобразовательное учреждение
Средняя общеобразовательная школа № 34 имени Н.К.Божененко

УТВЕРЖДЕНО

решением педагогического совета

от 08.08.2023 года протокол № 1

Председатель А.А.Спичакова



РАБОЧАЯ ПРОГРАММА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

тематическая

(тип программы: комплексная/тематическая)

Программирование на языке Pascal

(наименование)

1 год

(срок реализации программы)

13-14 лет

(возраст обучающихся)

Гайнутдинов Равиль Ринатович

(Ф.И.О. учителя, составителя)

КОПИЯ
ВЕРНА

Директор

МАОУ СОШ № 34

А.А.Артемько



Пояснительная записка

Примерная рабочая программа курса внеурочной деятельности «Программирование на языке Pascal» (далее — курс) для 8 класса составлена на основе требований Федерального государственного образовательного стандарта основного общего образования с учётом:

- Указа Президента Российской Федерации от 21.07.2020 № 474 "О национальных целях развития Российской Федерации на период до 2030 года";
- Федерального закона от 31.07.2020 № 304-ФЗ "О внесении изменений в Федеральный закон "Об образовании в Российской Федерации" по вопросам воспитания обучающихся";
- Приказа Министерства просвещения Российской Федерации от 31.05.2021 № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования»;
- Примерной программы воспитания - протокол Федерального учебно-методического объединения по общему образованию № 3/22 от 23.06.2022;
- Примерной основной образовательной программы основного общего образования - протокол Федерального учебно-методического объединения по общему образованию № 1/22 от 18.03.2022.

Внеурочная деятельность является неотъемлемой частью учебно-воспитательной работы в школе. Она способствует углублению знаний учащихся, развитию логического мышления, расширяет кругозор. Кроме того, данный курс по информатике имеет большое воспитательное значение, так как его цель не только в том, чтобы осветить какой-либо узкий вопрос, но и в том, чтобы заинтересовать учащихся предметом, вовлечь их в серьёзную самостоятельную работу.

Основная функция данного курса по информатике – выявление средствами предмета информатики нравственности личности, ее

профессиональных интересов. Содержание не дублирует базовый курс. Это курс, расширяющий базовый курс информатики, дающий возможность познакомиться учащимся с интересными нестандартными вопросами.

Изучение основ программирования связано с развитием целого ряда таких умений и навыков, которые носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач современной школы. Изучение программирования развивает мышление школьников, способствует формированию у них многих приемов умственной деятельности. Здесь роль информатики сродни роли математики в школьном образовании. Поэтому не использовать действительно большие возможности программирования для развития мышления школьников, формирования многих общеучебных, общеинтеллектуальных умений и навыков было бы, наверное, неправильно.

Изучая программирование на Паскале, учащиеся прочнее усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

Он расширяет базовый курс по информатике и информационным технологиям, является практико и предметно-ориентированным и дает учащимся возможность познакомиться с интересными, нестандартными вопросами информатики, проверить способности к информатике.

Вопросы, рассматриваемые в курсе, выходят за рамки обязательного содержания. Вместе с тем, они тесно примыкают к основному курсу. Поэтому данный элективный курс будет способствовать совершенствованию и развитию важнейших знаний и умений в области информатики, предусмотренных школьной программой, поможет оценить свои возможности по информатике и более осознанно выбрать профиль дальнейшего обучения.

Программа рассчитана на один год обучения. Образование осуществляется в виде теоретических и практических занятий для учащихся – 1 час в неделю, всего – 34 часа за год.

Личностные, метапредметные и предметные

результаты освоения учебного курса

Личностные образовательные результаты

готовность и способность к выполнению норм, требований, прав и обязанностей ученика;

умение вести диалог, работать в команде, оказывать посильную помощь одноклассникам при разработке и создании проектов;

готовность и способность к выполнению моральных норм в отношении взрослых и сверстников в школе, дома, во время внеурочных занятий;

готовность к выбору профильного образования, связанного с профессией программиста или частично связанной с ней;

развитие познавательных интересов и мотивов, направленных на изучение алгоритмических структур и команд среды программирования;

получение навыков прогнозирования своей деятельности в ходе создания проектов;

воспитание бережного отношения к техническим средствам обучения: компьютер, микрофон, наушники, проектор.

Личностные результаты освоения элективного курса по информатике для основного общего образования должны отражать готовность обучающихся руководствоваться системой позитивных ценностных ориентаций и расширение опыта деятельности на её основе и в процессе реализации основных направлений воспитательной деятельности, в том числе в части:

Гражданского воспитания: готовность к выполнению обязанностей гражданина и реализации его прав, уважение прав, свобод и законных интересов других людей; активное участие в жизни семьи, образовательной организации, местного сообщества, родного края, страны, в том числе в сопоставлении с ситуациями, отражёнными в литературных произведениях, написанных на русском языке; неприятие любых форм экстремизма, дискриминации; понимание роли различных социальных институтов в жизни

человека; представление об основных правах, свободах и обязанностях гражданина, социальных нормах и правилах межличностных отношений в поликультурном и многоконфессиональном обществе, формируемое в том числе на основе примеров из литературных произведений, написанных на русском языке; готовность к разнообразной совместной деятельности, стремление к взаимопониманию и взаимопомощи; активное участие в школьном самоуправлении; готовность к участию в гуманитарной деятельности (помощь людям, нуждающимся в ней; волонтерство).

Патриотического воспитания: осознание российской гражданской идентичности в поликультурном и многоконфессиональном обществе, понимание роли русского языка как государственного языка Российской Федерации и языка межнационального общения народов России; проявление интереса к познанию русского языка, к истории и культуре Российской Федерации, культуре своего края, народов России в контексте учебного предмета «Русский язык»; ценностное отношение к русскому языку, к достижениям своей Родины — России, к науке, искусству, боевым подвигам и трудовым достижениям народа, в том числе отражённым в художественных произведениях; уважение к символам России, государственным праздникам, историческому и природному наследию и памятникам, традициям разных народов, проживающих в родной стране.

Духовно-нравственного воспитания: ориентация на моральные ценности и нормы в ситуациях нравственного выбора; готовность оценивать своё поведение, в том числе речевое, и поступки, а также поведение и поступки других людей с позиции нравственных и правовых норм с учётом осознания последствий поступков; активное неприятие асоциальных поступков; свобода и ответственность личности в условиях индивидуального и общественного пространства.

Эстетического воспитания: восприимчивость к разным видам искусства, традициям и творчеству своего и других народов; понимание

эмоционального воздействия искусства; осознание важности художественной культуры как средства коммуникации и самовыражения; осознание важности русского языка как средства коммуникации и самовыражения; понимание ценности отечественного и мирового искусства, роли этнических культурных традиций и народного творчества; стремление к самовыражению в разных видах искусства.

Физического воспитания, формирования культуры здоровья и эмоционального благополучия: осознание ценности жизни с опорой на собственный жизненный и читательский опыт; ответственное отношение к своему здоровью и установка на здоровый образ жизни (здоровое питание, соблюдение гигиенических правил, сбалансированный режим занятий и отдыха, регулярная физическая активность); осознание последствий и неприятие вредных привычек (употребление алкоголя, наркотиков, курение) и иных форм вреда для физического и психического здоровья; соблюдение правил безопасности, в том числе навыки безопасного поведения в интернет-среде в процессе школьного языкового образования; способность адаптироваться к стрессовым ситуациям и меняющимся социальным, информационным и природным условиям, в том числе осмысляя собственный опыт и выстраивая дальнейшие цели; умение принимать себя и других, не осуждая; умение осознавать своё эмоциональное состояние и эмоциональное состояние других, использовать адекватные языковые средства для выражения своего состояния, в том числе опираясь на примеры из литературных произведений, написанных на русском языке; сформированность навыков рефлексии, признание своего права на ошибку и такого же права другого человека.

Трудового воспитания: установка на активное участие в решении практических задач (в рамках семьи, школы, города, края) технологической и социальной направленности, способность инициировать, планировать и самостоятельно выполнять такого рода деятельность; интерес к практическому

изучению профессий и труда раз личного рода, в том числе на основе применения изучаемого предметного знания и ознакомления с деятельностью филологов, журналистов, писателей; уважение к труду и результатам трудовой деятельности; осознанный выбор и построение индивидуальной траектории образования и жизненных планов с учётом личных и общественных интересов и потребностей; умение рассказать о своих планах на будущее.

Экологического воспитания: ориентация на применение знаний из области социальных и естественных наук для решения задач в области окружающей среды, планирования поступков и оценки их возможных последствий для окружающей среды; умение точно, логично выражать свою точку зрения на экологические проблемы; повышение уровня экологической культуры, осознание глобального характера экологических проблем и путей их решения; активное неприятие действий, приносящих вред окружающей среде, в том числе сформированное при знакомстве с литературными произведениями, поднимающими экологические проблемы; активное неприятие действий, приносящих вред окружающей среде; осознание своей роли как гражданина и потребителя в условиях взаимосвязи природной, технологической и социальной сред; готовность к участию в практической деятельности экологической направленности.

Ценности научного познания: ориентация в деятельности на современную систему научных представлений об основных закономерностях развития чело-века, природы и общества, взаимосвязях человека с природной и социальной средой; закономерностях развития языка; овладение языковой и читательской культурой, навыками чтения как средства познания мира; овладение основными навыками исследовательской деятельности с учётом специфики школьного языкового образования; установка на осмысление опыта, наблюдений, поступков и стремление совершенствовать пути достижения индивидуального и коллективного благополучия.

Личностные результаты – это сформировавшаяся в образовательном процессе система ценностных отношений учащихся к себе, другим участникам образовательного процесса, самому образовательному процессу, объектам познания, результатам образовательной деятельности. Основными личностными результатами, формируемыми при изучении информатики в основной школе, являются:

- наличие представлений об информации как важнейшем стратегическом ресурсе развития личности, государства, общества;
- понимание роли информационных процессов в современном мире;
- владение первичными навыками анализа и критичной оценки получаемой информации;
- ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- развитие чувства личной ответственности за качество окружающей информационной среды;
- способность увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
- готовность к повышению своего образовательного уровня, обучение с использованием средств программирования;
- способность и готовность к общению и сотрудничеству со сверстниками и взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой деятельности;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.

Метапредметные образовательные результаты

- создание и преобразование моделей и схем для решения задач;

осуществление выбора наиболее эффективных способов решения задач на компьютере в зависимости от конкретных условий;

построение логического рассуждения, включающее установление причинно-следственных связей;

овладение составляющими проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат своей деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы;

умение работать с разными источниками информации: находить информацию в различных источниках (тексте учебника, научной литературе, в словарях и справочниках, интернете), анализировать и оценивать информацию, преобразовывать информацию из одной формы в другую; создавать Лого-проекты;

способность выбирать целевые и смысловые установки в своих действиях и поступках при решении алгоритмических задач;

умение оформлять свои мысли письменно; слушать и понимать; адекватно использовать речевые средства для дискуссии и аргументации своей позиции, сравнивать разные точки зрения, отстаивать свою точку зрения, приводить аргументы, подтверждая их фактами.

умение строить логическое рассуждение, включающее установление причинно-следственных связей;

умение правильно включать и выключать устройства ИКТ, входить в операционную систему и завершать работу с ней, выполнять базовые действия с экранными объектами (перемещение курсора, выделение, прямое перемещение, запоминание и вырезание);

осуществление информационного подключения к локальной сети и глобальной сети Интернет;

умение входить в информационную среду образовательного учреждения, в том числе через Интернет, размещать в информационной среде различные информационные объекты;

· соблюдение требований техники безопасности, гигиены, эргономики и ресурсосбережения при работе с устройствами ИКТ, в частности учитывающие специфику работы с различными экранами.

формирование собственного информационного пространства: создавать системы папок и размещать в них нужные информационные источники;

моделирование с использованием средств программирования;

проектирование и организация своей индивидуальной и групповой деятельности, организация своего времени с использованием ИКТ.

Предметные образовательные результаты:

понимание терминов «исполнитель», «система команд»;

понимание термина «алгоритм»; знание основных свойств алгоритмов (фиксированная система команд, пошаговое выполнение, детерминированность, возможность возникновения отказа при выполнении команды);

составление неветвящихся (линейные) алгоритмов управления исполнителями и запись их на языке программирования;

понимание (формально выполнять) алгоритмов, описанных с использованием конструкций повторения (циклы), вспомогательных алгоритмов;

создание алгоритмов для решения несложных задач, используя конструкции повторения (циклы) и вспомогательные алгоритмы;

создавать и выполнять программы для решения несложных алгоритмических задач в среде программирования

Цели и задачи курса

- Формирование у учащихся интереса к профессиям, связанным с программированием.

- Формирование алгоритмической культуры учащихся.

- Развитие алгоритмического мышления учащихся.
- Освоение учащимися всевозможных методов решения задач, реализуемых на языке Паскаль.
- Формирование у учащихся навыков грамотной разработки программы.
- Углубление у школьников знаний, умений и навыков решения задач по программированию и алгоритмизации.

Планируемые результаты

По окончании обучения, учащиеся научатся:

- знать понятие алгоритма, его свойств, способов записи;
- знать основные базовые алгоритмические конструкции: следование, ветвление и цикл;
- иметь представление о исполнителях Муравей, Робот, Чертежник, Черепаха;
- уметь работать в различных средах исполнителей;
- иметь представление о языках программирования, в частности о языке программирования Паскаль;
- уметь работать в среде программирования ABC Pascal;
- использовать основные базовые алгоритмические конструкции при программировании на языке Паскаль;
- уметь писать программы на языке Паскаль, используя основные базовые алгоритмические конструкции и стандартные алгоритмы.

Содержание

Алгоритмизация (5 часов)

Понятие алгоритма и исполнителя. Система команд исполнителя. Формы записи алгоритмов. Выполнение алгоритмов для исполнителя. Графическое изображение алгоритма. Блок-схемы алгоритмов. Понятие линейного алгоритма. Условие. Разветвленные алгоритмы. Графическое изображение разветвленного алгоритма. Цикл. Способы записи цикла. Исполнители Муравей, Робот, Чертежник, Черепаха.

Программирование (29 часов)

Языки программирования, язык программирования Паскаль. Среда программирования ABC Pascal, элементы интерфейса ABC Pascal.

Данные. Типы данных. Константы. Переменные, присваивание значений. Арифметические выражения, правила записи на языке Паскаль, основные арифметические действия и их запись на языке Паскаль. Формат результата. Процедуры ввода и вывода и их простейшая форма.

Структура программы на языке Паскаль, линейный алгоритм. Программирование линейных алгоритмов.

Ветвление. Полное и неполное ветвление. Условный оператор, оператор выбора.

Циклы, цикл с предусловием, цикл с постусловием, цикл с параметром, итерационные циклы. Организация циклов с помощью блока «ветвление».

Графика на языке Паскаль. Способы решения алгоритмических задач. Программирование алгоритмических задач. Массивы данных. Обработка массива. Строковые переменные. Обработка данных, вводимых в виде символьных строк. Подпрограммы: процедуры и функции, локальные и глобальные переменные.

Исправление ошибок в простой программе с условными операторами. Обработка массива.

Календарно-тематическое планирование

№	Тема урока	Характеристика основных видов деятельности учителя	Характеристики основных видов деятельности обучающихся	Дата	Примечание
1.	Введение. Инструктаж по ТБ Что такое алгоритм. Исполнители вокруг нас.	Проводит инструктаж по ТБ. Объясняет алгоритм, его свойства, примеры алгоритмов. Формальный исполнитель. Система команд исполнителя.	Знать понятие алгоритма как фундаментального понятия информатики, определение алгоритма, его свойства. Уметь приводить примеры алгоритмов. Знать назначение исполнителя. Иметь представление об исполнителях, системе команд конкретного исполнителя, о формальном исполнении алгоритма.		
2.	Формы записи алгоритмов.	Объясняет формы записи алгоритмов. Графическое изображение алгоритма. Блок-схемы	Знать формы и способы записи алгоритмов, понятие блок-схемы, обозначения блоков. Уметь записывать алгоритм известными способами.		

3.	Линейные алгоритмы. Разработка линейных алгоритмов.	Объясняет понятие линейного алгоритма. Исполнитель Муравей.	Иметь представление о линейных алгоритмах. Знать правила записи линейного алгоритма; обозначения блоков. Уметь составлять линейные алгоритмы для исполнителя Муравей и записывать их различными способами. Уметь пошагово выполнять линейные алгоритмы.		
4.	Алгоритмы с ветвлениями.	Учит строить разветвленные алгоритмы. Графическое изображение разветвленного алгоритма. Исполнитель Робот.	Иметь представление о разветвляющихся алгоритмах. Уметь строить разветвляющийся алгоритм для исполнителя Робот.		
5.	Циклические алгоритмы.	Учит способам записи цикла. Исполнитель Чертежник и Черепаха.	Иметь представление о циклических алгоритмах. Уметь строить циклические алгоритмы для исполнителей Робот, Чертежник, Черепаха.		
6.	Решение задач.	Объясняет основные базовые алгоритмические конструкторы. Исполнители Робот, Чертежник, Черепаха.	Уметь решать задачи с помощью исполнителей Робот, Чертежник, Черепаха с использованием базовых алгоритмических конструкций.		

7.	Знакомство со средой исполнителя Scratch.	Знакомит с интерфейсом и возможностями среды исполнителя Scratch.	Иметь представление о возможностях среды исполнителя Scratch. Уметь решать простейшие задачи в Scratch.		
8.	Решение задач в Scratch.	Рассказывает возможности среды Scratch. Основные алгоритмические конструкции в среде Scratch.	Уметь решать задачи с различными базовыми конструкциями в Scratch.		
9.	Величины. Алгоритмы с величинами.	Рассказывает алгоритмы с величинами. Знакомит со средой КуМИР. Учит решению задач в среде КуМИР.	Иметь представление о величинах. Уметь составлять алгоритмы с величинами и записывать их на алгоритмическом языке в среде КуМИР.		
10.	Алгоритмы с величинами.	Учит разрабатывать алгоритмы с величинами в среде КуМИР.	Уметь составлять алгоритмы с величинами в среде КуМИР, используя различные базовые алгоритмические конструкции.		
11.	Среда разработчика ABC Pascal. Элементы интерфейса.	Знакомит со Средой программирования ABC Pascal, элементами интерфейса ABC Pascal.	Иметь представление о среде программирования ABC Pascal. Уметь запускать программу ABC Pascal.		

12.	Создание, компиляция, исполнение и отладка программ.	Формирует представление о программе на языке Паскаль, компиляции, исполнении, отладке программы.	Иметь представление о структуре программы на языке Паскаль. Уметь создавать программу в среде ABC Pascal, проводить компиляцию и отладку.		
13.	Синтаксис и семантика языка ABC Pascal 1.	Объясняет синтаксис языка Паскаль, семантика языка Паскаль. Типичные ошибки. Сообщения об ошибках.	Иметь представление о синтаксисе и семантики языка Паскаль. Знать типичные ошибки при составлении программы и уметь их исправлять.		
14.	Данные. Типы данных. Константы.	Рассказывает простые и составные типы данных в языке Паскаль. Переменные и константы.	Знать основные типы данных (простые и составные). Иметь представление о переменных и константах. Уметь записывать их на языке Паскаль.		
15.	Числовой тип данных.	Объясняет числовые типы данных, целые и вещественные типы данных.	Знать основные типы числовых данных в языке Паскаль. Уметь применять их при составлении программ.		

16.	Построение арифметических выражений.	Ставит задачи на арифметические выражения, правила записи на языке Паскаль, основные арифметические действия и их запись на языке Паскаль. Формат результата.	Знать правила записи основных арифметических действий и выражений на языке Паскаль. Уметь применять их в среде ABC Pascal 1.		
17.	Оператор присваивания.	Формирует знания об операторе присваивания, его записи на языке Паскаль.	Иметь представление об операторе присваивания. Уметь выполнять оператор присваивания.		
19.	Процедура ввода и ее формат. Простейший ввод.	Ставит задачу на Ввод и вывод данных на языке Паскаль, простейшие формы процедуры вывода.	Иметь представление о процедуре ввода и ее простейшей форме. Уметь применять её при программировании в среде ABC Pascal 1.		
20.	Вывод информации на экран в текстовом режиме.	Объясняет текстовый режим. Вывод информации на экране в текстовом режиме.	Иметь представление о процедуре вывода информации на экране в текстовом режиме. Уметь применять её при программировании в среде ABC Pascal 1.		

21.	Расчеты по линейному алгоритму.	Знакомит со структурой программы на языке Паскаль, линейном алгоритме.	Знать структуру программы на языке Паскаль. Уметь составлять блок-схему линейного алгоритма.		
22.	Программирование линейных алгоритмов.	Объясняет структуру программы на языке Паскаль, линейные алгоритмы.	Знать структуру программы на языке Паскаль. Уметь составлять блок-схему линейного алгоритма; программировать линейный алгоритм на языке Паскаль.		
23.	Графический режим.	Объясняет графический режим, модуль Graph.	Иметь представление о графическом режиме в среде ABC Pascal 1. Уметь подключать в программе модуль Graph.		
24.	Примитивы в графическом режиме.	Объясняет графический режим, модуль Graph, основные графические примитивы.	Иметь представление о графическом режиме в среде ABC Pascal 1. Уметь подключать в программе модуль Graph. Уметь рисовать основные графические примитивы в среде ABC Pascal 1.		

25.	Рисование с помощью примитивов.	Объясняет графический режим, модуль Graph, основные графические примитивы.	Уметь создавать рисунки с помощью основных графических примитивов в среде ABC Pascal I.		
26.	Данные логического типа и логические выражения.	Рассказывает логический тип данных, логические выражения.	Иметь представление о логическом типе данных. Уметь составлять логические выражения.		
27.	Организация программ разветвляющейся структуры. Условный оператор	Формирует представление о операторе ветвление, условном операторе, полном и неполном ветвлении.	Иметь представление об условном операторе. Уметь составлять блок-схему условного оператора и программировать на языке Паскаль.		
28.	Составной оператор. Многообразие способов записи ветвления.	Объяснить что такое оператор выбора.	Иметь представление об операторе выбора. Уметь составлять блок-схему оператора выбора и программировать на языке Паскаль.		
29.	Программирование циклов с предусловием и постусловием.	Рассказать об общей записи цикла, тело цикла, выход из цикла	Знать понятие цикла с предусловием и постусловием. Уметь описывать их на языке Паскаль. Иметь навыки создания программы для решения учебных задач.		

30.	Программирование циклов с параметром.	Рассказать что такое общая запись цикла с параметром, тело цикла, счетчик, выход из цикла	Знать понятие цикла с параметром. Уметь составлять блок-схемы основных алгоритмических структур и описывать их на языке Паскаль. Иметь навыки создания программы для решения учебных задач.		
31.	Одномерные массивы целых чисел. Описание, заполнение, вывод массива.	Формирует представление: регулярные типы данных, одномерные массивы, работа с элементами массива.	Иметь представление о понятии регулярного типа данных. Уметь описывать одномерные массивы на языке программирования и выводить их элементы на экран. Иметь навыки создания программ для решения учебных задач.		
32.	Вычисление суммы элементов массива.	Формирует представление: регулярные типы данных, одномерные массивы, работа с элементами массива.	Иметь представление об одномерных массивах. Уметь описывать одномерные массивы на языке программирования, выводить их элементы на экран; вычислять сумму элементов массива.		

33.	Творческая работа «Составление алгоритмов для исполнителей».	Ставит задачу на составление Алгоритма, исполнителя алгоритма, СКИ, основных базовых алгоритмических конструкции.	Составлять задачи для различных исполнителей на использование различных алгоритмических конструкций. Разрабатывать алгоритм для решения поставленной задачи.		
34.	Творческая работа «Составление программ на языке Паскаль».	Ставит задачу по созданию Программы, основные базовые алгоритмические конструкции на языке Паскаль, простые и составные типы данных.	Составлять задачи на использование различных алгоритмических конструкций на языке Паскаль. Разрабатывать алгоритм для решения поставленной задачи.		

Список литературы

1. Валов А.М. Алгоритмы и исполнители в учебном курсе информатике. НиПКипРО, 2014г.
2. Николаева Т.В. Информатика 7 класс. Методическое пособие для учителя. Кострома, 2005.
3. Крылов С.С., Лещинер В.Р., Якушкин П.А. ЕГЭ-2009. Информатика. Универсальные материалы для подготовки учащихся / под ред. В.Р. Лещинера / ФИПИ. — М: Интеллект-центр, 2009.
4. Культин Н.Б. Программирование в TurboPascal 7.0 и Delphi. СПб.: БХВ – Санкт-Петербург, 2008.
5. Алгоритмы и исполнители *К.Ю.Поляков, 2014г.*
6. <https://kpolyakov.spb.ru/>
7. Методическое пособие для учителей 7-9 класс. БИНОМ. Лаборатория знаний, 2018г.
8. УМК Босова Л.Л./Босова А.Ю. Информатика, 7-9 класс, БИНОМ. Лаборатория знаний, 2016г.

Сборник практических работ элективного курса по информатике Программирование на языке Pascal

Практическая работа № 1

Тема: Знакомство с системой ABC Паскаль. Ввод программ.

Основные приемы работы.

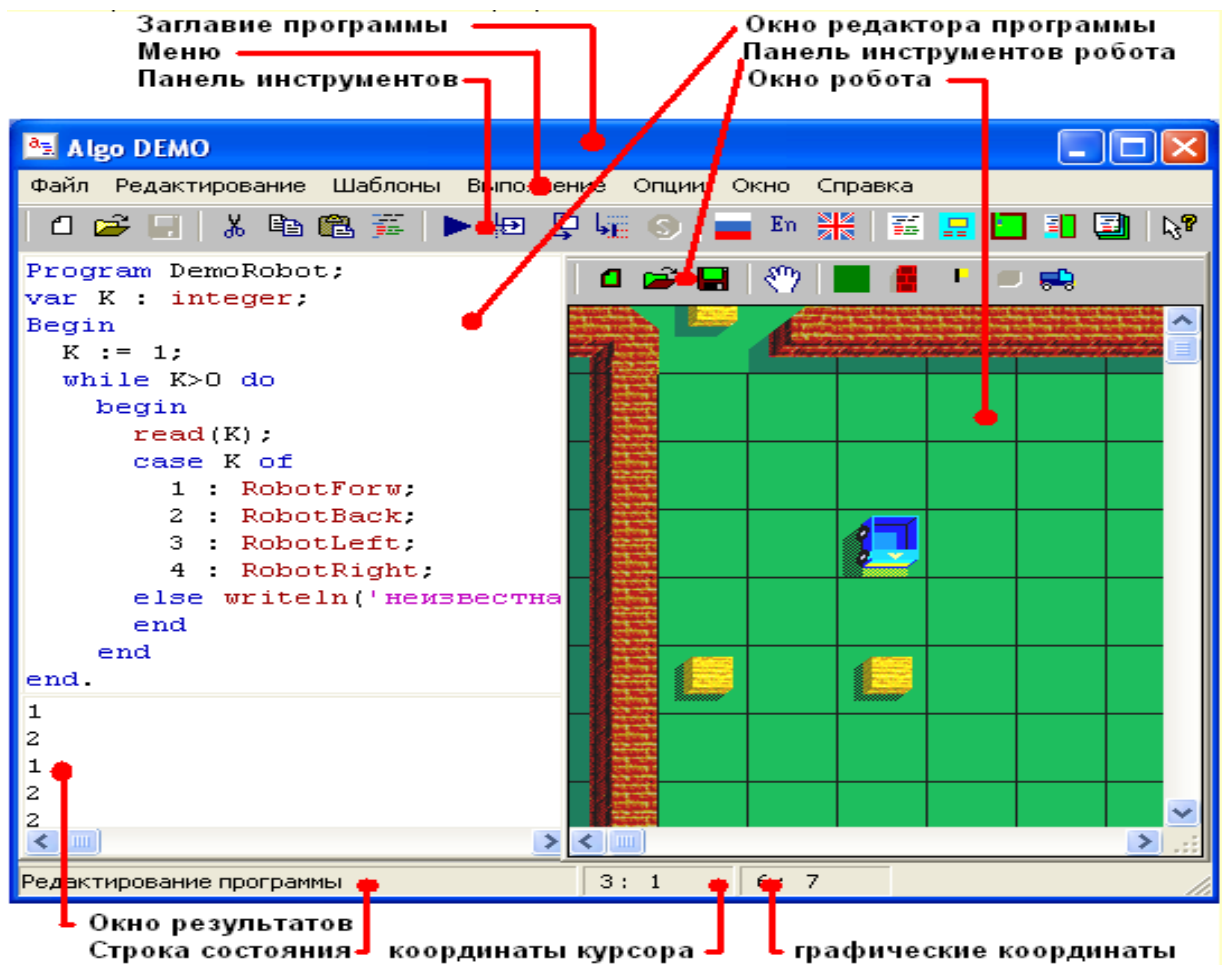
Простейший ввод и вывод данных различных типов.

Цель работы:

- Формирование и отработка умений и навыков работы в среде ABC Паскаль.
- Научиться вводить, редактировать и запускать простейшие программы.
- Развивать познавательный интерес к программированию, предмету.

Краткая теория

ABC Паскаль, как и любая программа, имеет свой интерфейс:



Оболочка ABC Паскаля (АЛГО) выполнена двумя языками - английским и русским. Для перехода на английский язык надо нажать кнопку с английским флажком, а на русский - с российским (Ctrl+E; Ctrl+U) или через меню. Следствием переключения является перевод всех ключевых слов подготовленной программы на выбранный язык.

Аналогично можно выбрать язык среды (меню, сообщений об ошибках, диалоговых окон).

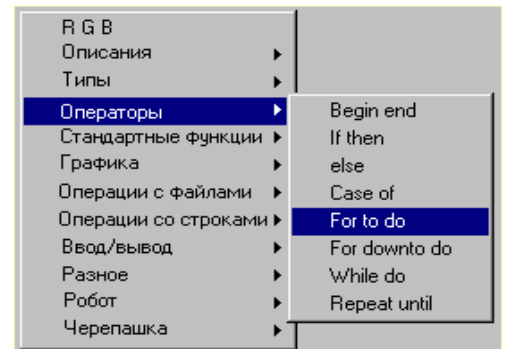
Язык оболочки и язык записи программы никак не связаны с режимом работы клавиатуры. Для переключения клавиатуры из латинских букв на кириллические и наоборот, как правило используют варианты Ctrl + Shift или Alt + Shift.

Рекомендуется записывать программу после набора каждых 10-20 строк текста и перед первым выполнением. Иначе вследствие ошибочных действий или сбоя программы подготовленный текст программы может быть утрачен.

Группа "Шаблоны"

Шаблоны используют для вставки в текст программы операторов, описаний и обращений к стандартным функциям и процедурам. При вставке оператора в текст программы, имеющийся текст раздвигается.

Во время вставки шаблонов список фактических параметров содержит соответствующее количество запятых, которые указывают на количество параметров. Исключение представляют шаблоны группы Графика, использующие цвета, составляющие R,G,B цвета. Для ввода чисел, которые задают нужный цвет, можно воспользоваться командой Шаблоны / RGB, которая позволяет с помощью системного диалога выбора цвета установить



нужный цвет и записывает R,G,B этого цвета в текст программы.

Операторы в Паскале отделяются друг от друга точкой с запятой и могут располагаться в одну строчку или начинаться с новой строки.

Наберем простейшую программу на Паскале с соблюдением правил.

Текст задачи: Ввести три целых числа (A,B,C) и вывести их среднее арифметическое (D)

Program Primer;	Заголовок программы с именем
Var A,B,C : Integer;	Объявлены три переменных целого типа
D : Real ;	Объявлена одна переменная дробного типа
Begin	Начало твоей программы
Write(' Укажи целые A, B, C - ');	Вывод произвольного текста на экран. ReadLn(A,B,C);
	Ввод пользователем значений переменных A,B,C
WriteLn('-----');	Вывод черты (текста) на экран.
D := (A+B+C)/3;	Вычисление значения переменной D по формуле
Writeln(' D = ',D:6:2);	Вывод текста и значения переменной
DEnd.	Конец программы

Ваши программы могут содержать следующие символы:

Латинские буквы	- A,B,C, ... a, b, ... x, y, z.
Русские буквы	- А,Б,В, ... а, б, ... э, ю, я.
Цифры	- 0, 1, 2, ...9.
Специальные символы	- +, -, /, =, >=, <=, <>, [], (), ;, :, #, ...

В качестве **имен переменных** и других объектов можно использовать **латинские** буквы, буквы с цифрой, латинские слова без пробелов (высота букв значения не имеют).

Для решения задач в любой программе выполняется обработка **данных**.

Данные могут быть самых различных типов: целые и дробные (вещественные) числа, символы (отдельные буквы), строки (текст), массивы (таблицы).

Все данные в языке Паскаль должны быть обязательно описаны в начале программы в разделе Var.

Различные типы данных занимают различные объемы памяти. Переменные могут менять свои значения в процессе выполнения программы.

Пример описания переменных для чисел:

Var

A, B : Real; - Дробное число

C : Integer; - Целое число от -2147483648 до 2147483647

Пример описания переменных для текста:

Var

A : Char; - Текст длиной в один любой символ .

C : String; - Длинный текст (до 255 любых символов)

X : String[15]; - Текст с фиксированным количеством символов (15)

Пример описания переменных логического (булевского) типа:

Var

A : Boolean; - Логический тип данных принимает одно из двух значений – **True**(Истина) или **False**(Ложь).

Наберем простейшую программу на Паскале с соблюдением правил.

Текст задачи: Ввести целое число (**X**) и вывести таблицу умножения на это число из 3 строк, начиная от 1 до 3, по образцу:

Укажи твое число - 8

-----1 x

8 = 8

2 x 8 = 16

3 x 8 = 24

Программу набирал – Иванов В.

Program ab;

Var X,Y : Integer;

Объявлены две переменных целого типа

Z : String;	Объявлена переменная типа текст
Begin	Начало программы
Write(' Укажи число X - ');	Вывод текста на экран.
ReadLn(X);	Ввод значения переменной X (Целого числа –X)
Write(' Укажи свое имя - ');	Вывод текста на экран.
ReadLn(Z);	Ввод значения Z (Текста твоего имени)
WriteLn('-----');	Вывод черты (текста) на экран.
Y := 1;	Переменной Y присвоили значение 1
WriteLn(' ',Y,' x ',X,' = ',X*Y);	Вывели на экран 1 x 8 = 8
Y := Y + 1;	Переменной Y присвоили значение на 1 больше
WriteLn(' ',Y,' x ',X,' = ',X*Y);	Вывели на экран 2 x 8 = 16
Y := Y + 1;	Переменной Y присвоили значение на 1 больше
WriteLn(' ',Y+1,' x ',X,' = ',X*Y);	Вывели на экран 3 x 8 = 24
WriteLn('----- -----');	Вывод черты (текста) на экран.
WriteLn(' Программу набирал -',Z);	Вывели на экран текст и твое имя
End.	Конец программы

Задания к работе

1. Загрузите ABC Паскаль.
1. Наберите предыдущую программу-код.
2. Запустите набранную программу несколько раз с различными значениями переменных A, B, C, поменяйте цифры в предпоследней строке на **6:3**, проанализируйте результат.
3. Переключите программу на русский вариант написания команд, запустите код.
4. Верните английский вариант написания команд.
5. Сохраните свою программу в **D:/Паскаль** с именем **Primer_Иванов**
6. Закройте программу Паскаль
7. Загрузите ABC Паскаль снова.
8. Откройте свою программу.
9. Введите первоначальное целое число и выведите числа по схеме:Укажите число - 9

9
10
11
12
10. Введите 2 целых числа и выведите числа по схеме:

Укажите 2 целых числа - 15 27

```

-----
15             27
16             26
17             25
18            24
  
```

11. Введите первоначальное целое число и выведите числа по схеме:

Укажите число - 15

```

-----
15             15
14             14
13             13
12            12
  
```

12. Наберите программу формирования таблицы Пифагора для 3-4 строк, по образцу:

```

    1  2  3  4
-----
1 | 1  2  3  4
2 | 2  4  6  8
3 | 3  6  9  12
4 | 4  8  12 16
-----
  
```

Практическая работа № 2

Тема: Стандартные операции и функции Паскаля.

Форматированный вывод информации.

Цель работы:

- Формирование и отработка умений и навыков работы по вводу и выводу информации разного типа в среде ABC Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

В Паскале пользователь может вычислять значения сложных выражений, состоящих из многих переменных и функций, с использованием скобок. Любые выражения записываются в одну строку с использованием скобок, что позволяет менять очередность вычислений в сложных выражениях. Пример выражения:

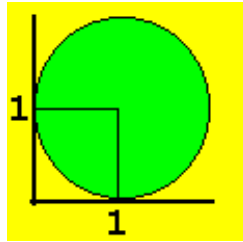
$Y := ((A+B)/(C+D*\text{Cos}(X))*2)/4$ - Переменной Y присвоено значение

выражения. Пример выражения.

$$\frac{\sin X + \cos Y}{2ab}$$

`(sin(x)+cos(y))/(2*a*b)`

Пример из математики



`(sqr(x-1)+sqr(y-1))<1`

Выражение принимает значение TRUE, когда точка (X,Y) находится внутри круга

В Паскале в распоряжении пользователя имеются множество операций и функций. Вот некоторые из них:

Операции целочисленной арифметики:

Целочисленное деление **Div** - (при целочисленном делении операция **Div** возвращает целую часть частного (дробная часть отбрасывается))

$$11 \text{ div } 4 = 2$$

$$7 \text{ div } 3 = 2$$

$$26 \text{ div } 5 = 5$$

Остаток от деления **Mod** - (остаток от деления (целая часть отбрасывается))

$$11 \text{ mod } 4 = 3$$

$$7 \text{ mod } 3 = 1$$

$$26 \text{ mod } 5 = 1$$

Обратите внимание: если $X \text{ Mod } Y = 0$, то X кратно Y

если $X \bmod 2 = 0$, то X четное число
если $X \bmod 2 = 1$, то X нечетное число

Стандартные функции Паскаля:

Abs(X)	-	Модуль числа
Sin(X)	-	Функция Синус
Cos(X)	-	Функция Косинус
Sqrt(X)	-	Корень
квадратный Round(X)	-	Округление
числа Trunc(X)	-	Целая часть дроби
Random	-	Случайное дробное число от 0 до 1
Random(N)	-	Случайное целое число от 0 до N

При использовании операций и функций пользователю необходимо помнить о типах данных исходных и конечных значений, например частное от деления двух целых чисел может быть дробным числом, что естественно соответствующим образом должно быть объявлено в разделе переменных **Var**.

Форматированный вывод информации:

Операторы **Write** и **Writeln** выводят значения переменных и тексты на экран. Вещественные (дробные) числа выводятся в формате с плавающей точкой, например: **1.748000000E+02**, что не совсем удобно и наглядно.

Решаем проблему с помощью форматированного вывода информации, где имеется возможность указать ширину поля вывода для переменной или выражения, например:

Var

```
A : string[15];  
B,C : integer; D,Y  
: Real;
```

Begin

```
ReadLn(B,C);  
ReadLn(A);
```

```
D := B Mod C;
```

```
Y := 25.178;
```

```
Y := Int(Y);
```

```
WriteLn(Y:6:1);
```

```
WriteLn(' A = ', A:22);
```

```
WriteLn(B:6, C:4);
```

```
WriteLn(D:8:2);
```

Вывод значения текста A шириной в 22 символа

Вывод значений чисел B и C шириной 6 и 4 символа

Вывод дробного числа D шириной 8 с 2 знаками

End.

после запятой

Форматированный вывод удобно использовать при выводе **чисел** и **текста** для выравнивания столбцов выводимой информации.

Задания к работе

1. Загрузите ABC Паскаль.
2. Наберите программу ввода одного или двух чисел и вывода в отформатированном виде всех вышеперечисленных значений функций Паскаля для введенных чисел. Сохраните программу **D:/Паскаль/R_02_Иванов**
3. Введите дробное число. Выведите отдельно на своей строке дробную и целую части.
4. Запиши программу вычисления выражения. Значения **X** и **Y** вводятся.
$$a = \frac{|x - y|}{10x + |xy + x + y|}$$
5. Запиши программу вычисления выражения. Значения **S**, **A**, вводятся.
$$f = s^2 + \frac{a}{a + s} \sqrt{s^4 - s^3} - 176$$
6. Запиши программу вычисления выражения. Значения **X**, **Y**, **Z** вводятся.
$$a = \frac{\sqrt{|x-1|+y} - \sqrt{|y+z|}}{1 + \frac{x^2}{2+z} + \frac{y^2}{4+z}} + zxy$$
7. Запиши программу вычисления суммы квадратов Синуса и Косинуса для двух-трех значений аргументов. $(\sin(X))^2 + (\cos(X))^2$
8. Напиши программу вычисления площади треугольника, если известны его стороны
9. Напиши программу ввода A, B. Увеличить A в 20 раз, уменьшить B в 30 раз. Во сколько раз отличается первоначальная сумма A + B от конечной суммы A + B ?
10. Найдите площадь прямоугольной рамки, внутренняя сторона которой равна 5, внешняя заданному числу R >= 5.

Практическая работа № 3

Тема: Условный оператор и оператор выбора Паскаля.

Цель работы:

- Формирование и отработка умений и навыков работы по использованию ветвлений и выбора в среде программирования ABC Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Решение большинства задач **редко** сводится к простому последовательному расчету (линейным алгоритмам). Чаще порядок вычислений зависит от определенных условий, например от исходных данных или от промежуточных результатов, полученных несколько ранее. Для организации вычислений в зависимости от какого-либо условия в Паскале используется условный оператор.

Для формулировки условия используются следующие логические операции:

- | | | | |
|---|----------|----|---------------------|
| > | - больше | >= | - больше либо равно |
| < | - меньше | <= | - меньше либо равно |
| = | - равно | <> | - не равно |

Кроме того, для записи сложного условия, когда необходимо сразу проверить несколько условий, используются операции их объединения:

And (оба условия должны выполняться - **И**)

Or (хотя бы одно из условий должно выполняться - **Или**)

Not (Логическое отрицание - **Не**)

При выполнении программы компьютером или человеком при проверке, любое условие может иметь только два значения: либо **истина (True)**, когда оно выполняется, либо **ложь (False)**, когда оно не выполняется.

Пример условного оператора на Паскале:

A := 2; B := 4;

If A > B Then

WriteLn(' A > B ');

Else

WriteLn(' A <= B ');

Если условие **A > B** истинно, то выполняются команды между **Then** и **Else**. Но если условие ложно, то будут выполняться команды после **Else**.

По ветке **Then**, и по ветке **Else** должен выполняться единственный оператор.

Если по смыслу задачи необходимо выполнить несколько команд, тогда следует использовать скобки (составной оператор), например:

A := 2; B := 4; C := 0;

If A > B ThenBegin

WriteLn(' A > B ');C :=

A-B;

End

Else

WriteLn(' A <= B ');

If A > B ThenBegin

WriteLn(' A > B ');End

C := A-B;

Альтернативная часть **Else** может вообще отсутствовать, если в ней нет необходимости.

A := 2; B := 4; C := 0;

If A > B ThenBegin

If A = C Thenbegin

WriteLn(' A = C ');C

:= A-B;

Условные операторы могут быть вложены друг в друга. При использовании вложенных ветвлений во избежание недоразумений нужно также использовать скобки - **Begin** и **End**

```

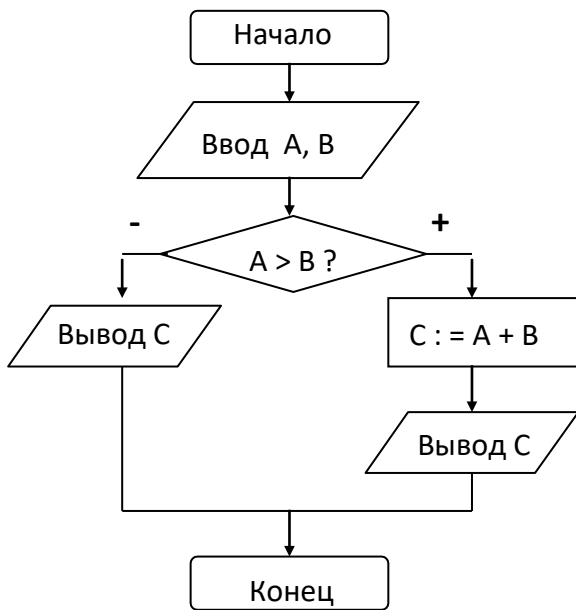
end
else
begin
C := A + B;
WriteLn(' A <> C');end
End
Else
WriteLn(' A <= B ');

```

Условие может состоять из нескольких простых условий. В этих случаях каждое простое условие заключается в круглые скобки и используются логические связки (логические операции) -

And, Or, Not

Конструкцию ветвления (условный оператор) можно изобразить блок-схемой. Вот пример простой блок-схемы, содержащей ветвление.



Использование оператора варианта:

Оператор варианта **Case** необходим в тех случаях, когда в зависимости от значений какой-либо переменной надо выполнить те или иные операторы.

```

Case управляющая переменная of
Набор значений 1 : оператор 1;
Набор значений 2 : оператор 2;
Набор значений 3 : оператор 3;
.....
Набор значений N : оператор N;
Else
Альтернативный оператор;
End;

```

Если управляющая переменная принимает значение из набора значений 1, то выполняется оператор 1.

Если управляющая переменная принимает значение из набора значений 2, то выполняется оператор 2 и т.д.

Примеры:

Program ab;

Var D : Integer;

Begin

**WriteLn(' Укажи число от 1 до 31 – '); ReadLn(D);Case
D mod 7 of**

**1 : WriteLn(' Понедельник ');2 :
WriteLn(' Вторник ');**

**3 : WriteLn(' Среда ');
4 : WriteLn(' Понедельник ');5 :
WriteLn(' Вторник ');
6 : WriteLn(' Среда ');**

Else

WriteLn(' Воскресенье ');

End;

Program ab;

Var D : Integer;

Begin

**WriteLn(' Укажи число от 1 до 12 – '); ReadLn(D);If (D
>= 1) and (D <= 12) Then**

Case D of

12, 1, 2 : WriteLn(' Зима ');

3.. 5 : WriteLn(' Весна ');

6.. 8 : WriteLn(' Лето ');

9.. 11 : WriteLn(' Осень ');

End

Else

WriteLn(' Введено ошибочное число ! ');

End.

Задания к работе

1. Введите три числа. Найдите Минимальное из них. Покажи его.
2. Введите три числа. Найдите разность между Максимальным и Минимальным из них. Покажи разность.
3. Введите четыре числа. Найдите разность между Максимальным и Минимальным из них. Покажи разность.
4. Введите натуральное число и вопрос о характере вывода чисел. Выведите 4-5 четных или нечетных числа через запятую между ними.
5. Получи несколько чисел (4-5 чисел) через запятую, если известно, что каждое очередное число получается суммой двух предыдущих. Первые два числа ввести.
6. Ввести целое двузначное число. Проверить кратно ли оно - 7.
7. Ввести четыре различных числа.
Программа не должна разрешать ввод одинаковых чисел.
8. Ввести число. Проверить делится ли оно на 3 и 11 нацело одновременно.
9. Ввести целое однозначное число. Проверить является ли оно простым.
(Число считается простым, если делится само на себя и единицу).

Практическая работа № 4

Тема: Циклические структуры Паскаля.

Цель работы:

- Формирование и отработка умений и навыков работы по использованию циклических структур в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Циклом в программировании называют повторение одних и тех же действий (команд). Последовательность действий, которые повторяются в цикле, называют - **телом цикла**.

Существует несколько типов циклических структур:

1. цикл со счетчиком
2. цикл с предусловием
3. цикл с постусловием.

Цикл со счетчиком:

Данный цикл обычно используют тогда, когда тело цикла должно быть выполнено заданное количество раз. Примеры:

```
Var I, C : integer;  
Begin  
  For I:= 3 to 50 doBegin  
    WriteLn(' I = ', I);  
    C:= I*I;  
  End
```

Выполнение цикла начинается с присвоения параметру стартового значения. Затем следует проверка, не превосходит ли параметр конечного значения.

Если результат проверки утвердительный, то цикл не выполняется ни разу и управление передается к первой команде после цикла. В противном случае выполняется тело цикла и параметр автоматически меняет свое значение на следующее (на 1-у

```
For I:= 30 downto - 5 do  
  WriteLn(' I = ', I);
```

```
For N:= 'A' to 'R' do  
  WriteLn(NI);
```

Оператор цикла с предусловием - (Пока ... Делать).

Пример:

```
X := - 120;  
While X < 0 do  
  Begin  
    WriteLn(X);X  
    := X +4.5;  
  End;
```

Условие выполнения тела цикла While проверяется **до начала** работы, при входе в цикл. Поэтому, если условие сразу не выполняется, то тело цикла игнорируется и управление передается первому оператору после цикла.

Тело цикла будет выполняться до тех пор, пока логическое условие истинно.

В цикле данного типа (с предусловием)

Оператор цикла с постусловием - (Repeat ... Until).

Пример:

```
X := - 120;
```

```
Repeat
```

```
  WriteLn(X);X  
  := X +4.5;
```

```
Until X > 0
```

Цикл с постусловием всегда выполняется хотя бы один раз.
Тело цикла Repeat выполняется до тех пор, пока не станет истинным условие после Until .
В данном цикле пользователь сам предусматривает изменение всех переменных, здесь также автоматически ничего не ме-

Для всех циклов по команде **Exit** можно покинуть цикл. Пример:

```
X := - 120;
```

```
Repeat
```

```
  WriteLn(X);X  
  := X +4.5;
```

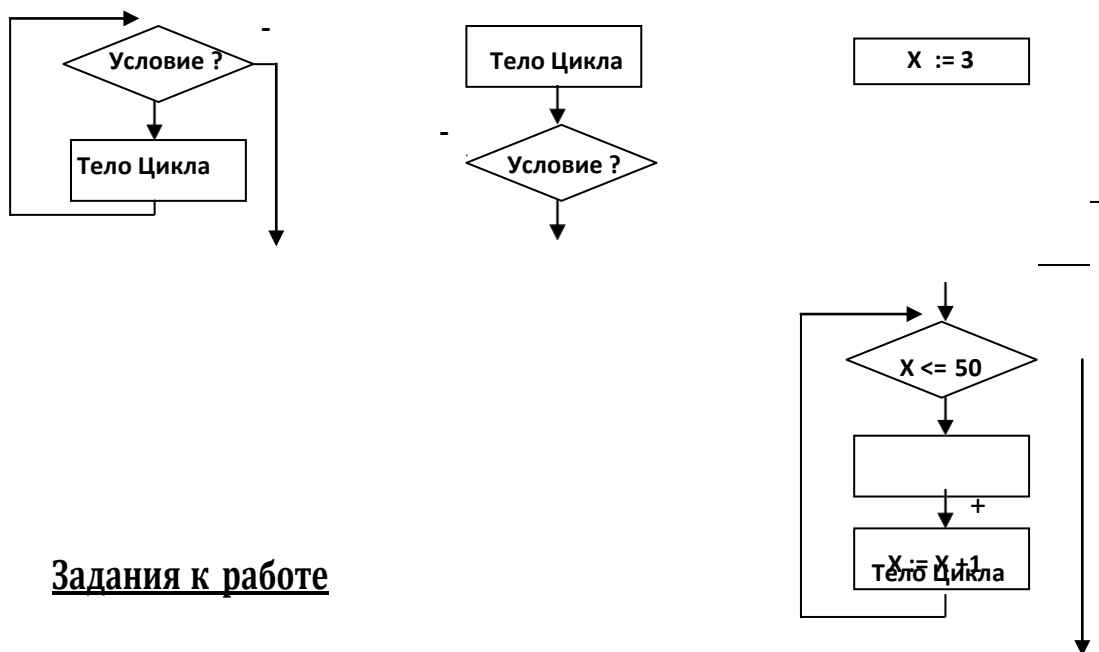
```
  If X = - 20 then Exit;
```

```
Until X > 0;
```

При использовании циклов следует помнить следующее:

- Чтобы цикл гарантированно заканчивался, содержимое цикла должно обязательно влиять на условие цикла.

В заключении нарисует блок-схемы циклических структур Паскаля.



Задания к работе

1. Введите два натуральных числа. Выведите все числа через запятую между ними, используя различные виды циклов.
2. Введите два натуральных числа и вопрос о характере выводимых чисел. Выведите все четные или нечетные числа через запятую между ними, используя все типы циклов.
3. Запиши программу вычисления Факториала указанного числа. ($5! = 1*2*3*4*5$)
4. Получи 5-6 чисел через запятую, используя цикл, если известно, что каждое очередное число получается суммой двух предыдущих. Первые два числа ввести.

5. Ввести 2 произвольных целых числа. На сколько отличаются суммы четных чисел от нечетных чисел от первого до второго числа.
6. Ввести натуральное двухзначное число. Определить, является ли оно простым. (Число – простое, если оно делится без остатка на себя и единицу).
7. Запиши программу нахождения НОД (Наибольшего Общего Делителя двух чисел)
8. При возведении в квадрат иногда последние цифры повторяют эти числа. ($5 \times 5 = 25$). Между 25 и 1000 есть такие числа. Покажи их.
9. Найди все целые двузначные числа кратные 7.
10. Найди все целые двузначные числа, сумма цифр которых равна 8.
11. Найди все трехзначные числа, сумма первых двух цифр которых меньше суммы второй и третьей цифр.
12. Среди четырехзначных целых чисел найди те, в которых все цифры различны.
13. Найди все целые трехзначные числа, которые делятся на 3 и 11.
14. Число 66 можно представить в виде суммы четырех последовательных чисел. Найдите их.
15. Число 1190 можно представить в виде произведения двух последовательных чисел. Найдите их.
16. Сколько 0 в произведении всех натуральных чисел от 10 до 20.
17. Чтобы пронумеровать страницы в книге, понадобилось 1164 цифры. Сколько в ней страниц.
18. Сколько раз встречается цифра 1 в числах от 1 до 1000 включительно.

Практическая работа № 5

Тема: Обработка массивов.

Цель работы:

- Формирование и отработка умений и навыков работы по использованию массивов в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Часто для работы с множеством однотипных данных (числами, датами, текстами и т.п.) оказывается удобным использовать массивы. **Массив** – структурированный тип данных, состоящий из конечного числа однотипных элементов. Для доступа к данным, хранящимся в определенном элементе массива, необходимо указать имя массива и порядковый номер этого элемента, называемый **индексом**. Массивы бывают одномерные (линейные), двумерные (прямоугольные) и т.д.

При работе с массивами их необходимо, как и все данные, описать в разделе переменных **Var**. Это можно сделать так:

```
Var A : array[1..50] of integer;
      B : array[1..12] of string;
      C : array[1..10,1..15] of real;
```

1. Объявлен линейный массив с именем А из целых чисел, состоящий из 50 элементов. Нумерация элементов с 1.
2. Объявлен линейный массив с именем В из текстов, состоящий из 12 элементов.
3. Объявлен двумерный массив с именем С из дробных чисел, состоящий из 10 строк по 15 столбцов.

После объявления массива его элементы **пусты**.

Для обращения к отдельному элементу массива, присвоения значения отдельному элементу используют имя массива и порядковый номер элемента (индекс).

При заполнении массива данными необходимо последовательно вводить 1-й, 2-й, 3-й и т.д. его элементы и аналогичным образом поступать при его выводе на экран или принтер. Следовательно, для ввода и вывода необходимо **организовывать цикл**, в котором практически все операции с массивом необходимо проводить поэлементно.

Пример присваивания значения конкретным элементам:

A[1] := 2; - Первому элементу присвоено значение 2

A[3] := X+1; - Третьему элементу присвоено значение X+1

A[N] := A[N-1]; - Элементу с номером N присвоено значение предыдущего элемента

A[N] := N; - Элементу с номером N присвоено значение N

При заполнении массива элементы могут в цикле заполняться пользователем с клавиатуры или в цикле программно (автоматически).

Пример ручного ввода:

```
Var A : array[1..12] of Real;J, N
      : integer;
```

```
Begin
```

```
    Write(' Укажи кол-во элементов массива не более 12 – ');
    ReadLn(N);
```

```
    For J := 1 to N do
    Begin
```

```
End.
```

```
End;
```

```
Write(' A [',J,'] = ');
```

```
ReadLn(A[J]);
```

Пример программного ввода массива с параллельным показом:

```
Var A : array[1..120] of Integer;J,
      N : integer;
```

```
Begin
```

```
    Write(' Укажи кол-во элементов массива не более 120 – ');
    ReadLn(N);
```

```
    For J := 1 to N do Begin
```

```
        E
```

```
        n
```

Пример вычисления суммы только четных элементов массива кратных 5

Program ab;

Var A : array[1..10] of Integer;

J, N : integer;

Begin N := 0; Randomize; При каждом пуске – новая череда случайных чисел

For J := 1 to 10 doBegin

A[J] := Random(100); клетке с номером J присваивается случайное

Write(A [J] , ' , '); число от 0 до 100 и показывается на экране.

End;

WriteLn; Перевели курсор на следующую строку

WriteLn('-----');

J := 1;

While J <= 10 do

Begin

If (A[J] mod 5 = 0) and (A[J] mod 2 = 0) ThenN :=
N + A[J];

J := J + 1;

End;

WriteLn(' Сумма = ', N:4);

End.

Пример поиска максимального элемента массива и его номера:Var

A : array[1..20] of Integer;

J, Nomer, Max : integer;

Begin Randomize;

For J := 1 to 20 doBegin

A[J] := Random(100); Write(A [J] , ' , '); WriteLn; WriteLn;
End;

Max := A[1]; Nomer := 1;For

J := 2 to 20 do

Begin

If A[J] > Max ThenBegin

End;

Max := A[J]; Nomer := J;

WriteLn(' Максим. элемент = ', Max:4, ' Его номер – ', Nomer:3);

End.

Обработка определенной части элементов прямоугольного массива:

Элементы главной диагонали:

for I := 1 to N do

begin

{ цикл с элементами a[i,i] }

end;

Выше главной диагонали:for

I := 1 to N do

begin

for J := i+1 to N do

{ цикл с элементами a[i,J] }

end;

end;

Ниже главной диагонали:for

I := 1 to N do

begin

for J := 1 to I - 1 do

{ цикл с элементами a[i,J] }

end;

end;

Элементы обратной диагонали:

```
for I := 1 to N do
  begin
    { цикл с элементами a[i,N+1-i] }
  end;
```

Выше обратной диагонали:

```
For I := 1 to N-1 do begin
  for J := 1 to N-i do begin
    { цикл с элементами a[i,J] }
  end;
end;
```

Ниже обратной диагонали:

```
for I := 2 to N do
  begin
    for j := N-i+2 to N do
      begin
        { цикл с элементами a[i,J] }
      end;
  end;
```

Задания к работе

1. Найди разность между максимальным и минимальным элементами массива.
2. Подсчитайте кол-во и сумму всех положительных элементов числового массива.
3. Введите произвольное число. Заполните массив. Подсчитайте количество элементов, которые меньше указанного числа.
4. Заполни массив целыми случайными числами. Подсчитай сумму всех четных чисел.
5. Заполни массив целыми случайными числами. Подсчитай сумму всех простых чисел
6. Заполни массив целыми случайными числами, покажи его. Поменяй местами числавсех четных и нечетных ячеек. Покажи все элементы после перестановки.
7. Заполни массив случайными дробными числами. Первую половину массива заполнидробной частью числа а вторую половину массива целой частью.
8. Заполни массив целыми случайными числами. Покажи числа. Помести элементы в об-ратном порядке.
9. Заполни массив случайными целыми числами. Сдвинь все элементы налево на1 число а последний элемент перенеси в 1 клетку массива.
10. Заполни двумерный массив случайными целыми числами так, чтобы на его обоих диа-гоналях были только 0.
11. Запиши программу, реализующую сортировку числового массива по убыванию.
12. Заполните массив $A(N,N)$ целыми случайными числами. Покажите индекс самого пер-вого положительного элемента, кратного заданному числу.

13. Заполни массив случайными числами. Найди наибольшее кол-во равных элементов.
14. Заполни массив случайными числами, включая отрицательные. Вычисли максимальное число подряд идущих положительных чисел.
15. Заполни массив случайными числами. Найди сумму уникальных элементов.
16. Заполни массив случайными, не повторяющимися числами.
17. Заполните массив $A(N,N)$ целыми случайными числами. Покажите все числа выше главной диагонали, а все остальные замени 0.
18. Заполните массив $A(N,N)$ целыми случайными числами. Покажите все числа ниже главной диагонали, а все остальные замени 0.
19. Заполните массив $A(N,N)$ целыми случайными числами. Замените самое первое отрицательное число, его индексом.
20. Заполните массив $A(N,N)$ целыми случайными числами. Найдите кол-во строк, не содержащих отрицательных элементов. Покажите их номера.
21. Заполни массив. Задай произвольный числовой диапазон, например 3-8. Подсчитай кол-во строк, содержащих хотя бы один элемент из Вашего диапазона. Выведи их номера.
22. Заполни массив случайными числами. Подсчитай количество различных (не повторяющихся) чисел. Покажи их.
23. Заполни массив. Покажи его элементы. Покажи его после удаления каждого третьего элемента.
24. Заполните массив $A(N,N)$ целыми случайными числами. Покажи все его элементы, расположенные выше главной диагонали, остальные замени пробелом.
25. Заполните массив $A(N,N)$ целыми случайными числами. Покажи все его элементы, расположенные ниже главной диагонали, остальные замени пробелом.
26. Создай проект «Проверка знаний», так чтобы ваш проект проверял знание таблицы умножения у первоклассника.

Практическая работа № 6

Тема: Графика ABC Паскаля .

Цель работы:

- Формирование и отработка умений и навыков по использованию графики в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Для графики в среде Паскаль выделено рабочее поле – так называемый лист, просматриваемый ползунками. Началом координат считается верхний левый угол листа.

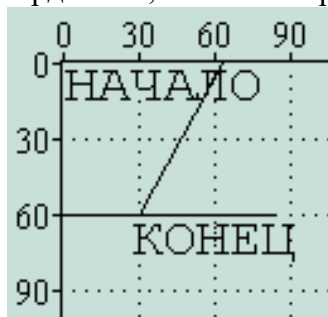
Ось X направлена слева направо, Y - сверху вниз. Для всей графики координаты задают в форме (X,Y). Если координаты при построении изображения выходят за пределы листа, то информация за пределами листа игнорируется. Исключение составляют операторы ReadLn и WriteLn. При попытке вывода за нижнюю границу листа все изображение двигается вверх на величину высоты символа.

Система запоминает последний вывод и параметры инструментов. В момент запуска программы устанавливаются нулевые координаты (верхний левый угол экрана), черный карандаш единичной толщины и черный цвет текста. Цвет заполнения замкнутых фигур (цвет кисти) выбирается прозрачный, т.е. фигуры не закрашиваются.

В процессе работы активные координаты меняются так, чтобы следующий вывод осуществлялся там, где закончился предыдущий.

Увидеть, как меняются текущие координаты, можно на примере такой программы:

```
Program Координаты;  
Begin  
  Write( 'НАЧАЛО' );  
  LineTo( 30, 60 );  
  Write( 'КОНЕЦ' );  
  LineTo( 0, 100 )  
End.
```

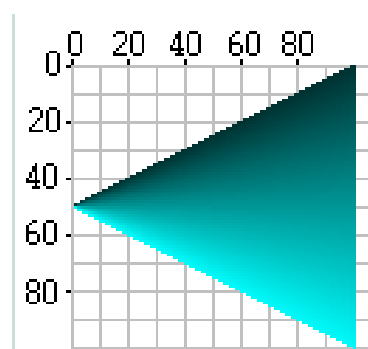


Цвет текста для Read и Write устанавливаются командой ЦветТекста:

По умолчанию установлен черный цвет текста (0,0,0). (Красный, Зеленый, Синий), пользователь регулирует каждый цвет в диапазоне 0-255. Пример:

```
Program ЦветТекста;  
Begin  
  WriteLn( 'Информатика' ); - Текст Информатика выведется черным цветом.  
  TextColor( 0, 0, 255 ); - Включили для будущего вывода Синий цвет.  
  WriteLn( 'Майна' )  
end.
```

Переместиться в точку:



MoveTo (x, y)Program Перемещение;

Var

i: integer;

Begin

For i: =0 to 100 do

Begin

MoveTo(100, i);

Pen(1, 0, 2*i+50, 2*i+50);

LineTo(0, 50)

end

end.

Построение Прямоугольника: Rectangle (x1, y1, x2, y2)

Где **x1,y1** и **x2,y2** - координаты двух диаметрально противоположных вершин прямоугольника. Прямоугольник будет построен активным карандашом и закрашен активным цветом кисти, или не закрашен, если установлен прозрачный цвет кисти.

После выполнения процедуры активная графическая позиция устанавливается в точку с координатами (**x2, y2**).

Program Прямоугольник;

Begin

Rectangle(10, 10, 80, 80);

Pen(1, 255, 0, 0);

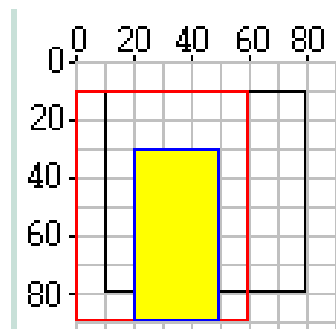
Rectangle(60, 10, 0, 90);

Pen(1, 0, 0, 255);

Brush(1, 255, 255, 0);

Rectangle(20, 30, 50, 90)

end.



Построение Эллипса: Ellipse (x1, y1, x2, y2)

где **x1, y1** и **x2, y2** - координаты двух диаметрально противоположных вершин прямоугольника, в который вписывается эллипс. Эллипс будет построен активным карандашом и закрашен активной кисточкой, или не закрашен, если установлен прозрачный цвет кисти.

После выполнения процедуры активная графическая позиция не изменяется

Program Эллипс;

Begin

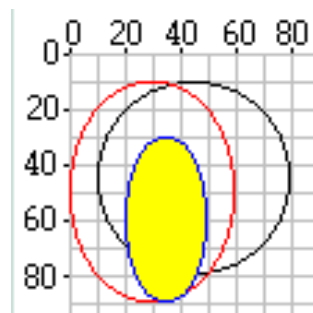
Ellipse(10, 10, 80, 80);

Pen(1, 255, 0, 0);

Ellipse(60, 10, 0, 90);

Pen(1, 0, 0, 255);

Brush(1, 255, 255, 0);



Ellipse(20, 30, 50, 90)

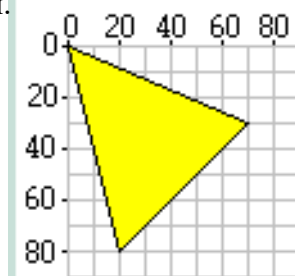
end.

Закрасить замкнутую фигуру: Fill (x, y)

Начиная из точки с координатами x,y, процедура заполняет экран во всех направлениях активным цветом кисти до тех пор, пока не встретится граница другого цвета, чем цвет указанной точки (x,y), т.е. процедура меняет цвет запертой области, внутри которой лежит точка (x,y) на активный цвет кисти.

Процедура не выполняет никаких действий, если задан прозрачный цвет заполнения или цвет точки (x,y) совпадает с цветом заполнения.

```
Program Закраска;  
Begin  
  LineTo( 70, 30 );  
  LineTo( 20, 80 );  
  LineTo( 0, 0 );  
  Brush( 1, 255, 255, 0 );  
  Fill( 10, 10)
```



end.

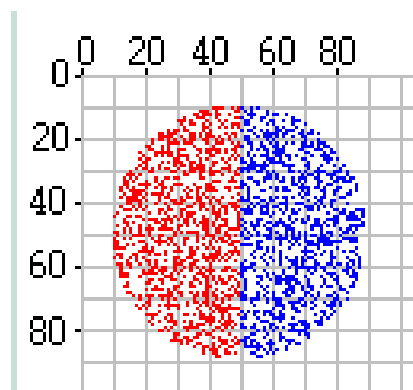
Очистить окна результатов: Clear

Очищает окно результатов и устанавливает начальные параметры графики (цвет, толщину, координаты ...)

Нарисовать Точку: Point (x, y)

Цвет точки определяется активным цветом карандаша. После выполнения процедуры активная графическая позиция устанавливается в точку с координатами (x, y).

```
Program Точка;  
Var  
  i, x, y: integer;  
Begin  
  For i:=1 to 5000 do  
    Begin  
      x := random( 100 );  
      y := random( 100 );  
      If  $\text{sqr}( x-50 ) + \text{sqr}( y-50 ) < 1600$  then  
        Begin  
          If x > 50 then  
            Pen( 1, 0, 0, 255 )  
          end.  
        end.  
    end.  
  end.  
end.
```



end

else end

```
Pen(
  1,
  255,
  0, 0 );
Point(
  x, y )
```

Нарисовать Линию : Line (x1, y1, x2, y2)

где (x1, y1) и (x2, y2) - координаты концов линии. Линия будет проведена активным карандашом. После выполнения процедуры активная графическая позиция устанавливается в точку с координатами (x2,y2).

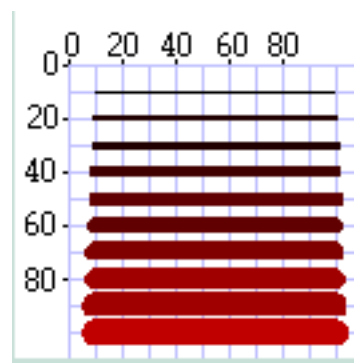
```
Program Линия;
Begin
  Line( 0, 0, 90, 90 );
  Pen( 2, 0, 0, 255 );
  Line( 30, 30, 90, 30 );
  Pen( 6, 255, 0, 0 );
  Line( 10, 10, 10, 90)
end.
```



Установить цвет и толщину линий : Pen (n, r, g, b)

где n задает толщину линии в пикселях, r, g, b - задают части красного, зеленого и синего в цвете линий. Установки действуют на команды **LineTo**, **Line**, **Rectangle**, **Ellipse**, **Point**. По умолчанию установлен черный цвет линий (0,0,0) и единичная толщина.

```
Program Цвет_Толщина;
Var
  i: integer;
Begin
  For i:= 1 to 10 do
    Begin
      Pen( i, i*20, 0, 0 );
      Line( 10, i*10, 100, i*10 )
    end
  end
end.
```



Когда следующая линия начинается из конца предыдущей, как бывает при построении ломанных линий или графиков, удобно пользоваться процедурой ЛинияДо: **LineTo** (x, y)

где (x, y) - координаты конца линии. Линия начинается от активной графической позиции. Линия будет проведена выбранным карандашом.

После ЛинияДо графическая позиция устанавливается в точку с координатами (x, y).

Program График;

Var

i: integer;

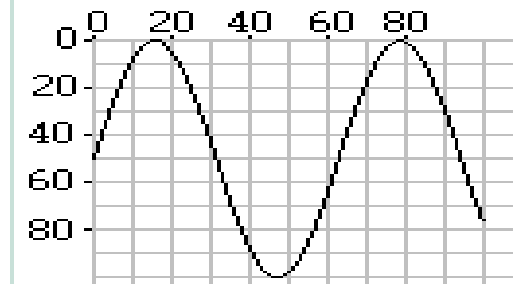
Begin

MoveTo(0, 50);

For i:=1 to 100 do

LineTo(i, round(50-50*sin(i/10)));

end.



Установка цвета заливки: Brush (k, r, g, b)

где **r, g, b** - задают части красного, зеленого и синего в цвете заливки, а параметр **k** задает **стиль** заливки. Если **k = 1** то замкнутые фигуры закрашиваются, а при **k = 0** прозрачный цвет, не закрашиваются. Установки действуют на **Rectangle**, **Ellipse** и **Fill**.

Program Кисть;

Begin

Brush(1, 255, 255, 0);

 Rectangle(10, 10, 50, 50);

Brush(1, 255, 255, 255);

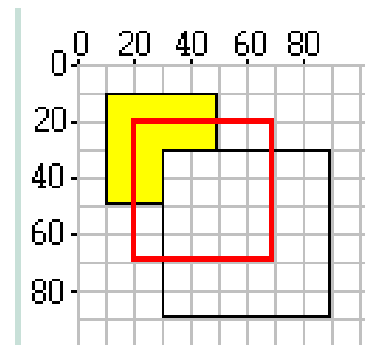
 Rectangle(30, 30, 90, 90);

 Pen(2, 255, 0, 0);

Brush(0, 0, 0, 0);

 Rectangle(20, 20, 70, 70);

end.



Установка Шрифта: Font (h, a, b)

где **h** - высота в пикселях, **a** - угол наклона в градусах, **b** - насыщенность шрифта. Насыщенность число из диапазона 0 - 1000. Обычный шрифт имеет насыщенность 400, а жирный - 600. Установки действуют на **Read** и **Write**.

По умолчанию установлен шрифт (8,0,400).

Замечание: не все шрифты подвергаются наклону.

Program Шрифты;

Begin

 Font(10, 0, 400);

 WriteLn('Шрифт(10,0,400)');

 Font(8,0,800);

 WriteLn('Шрифт(8,0,800)');

 MoveTo(80, 100);

 Font(10,30,400);

 WriteLn('Шрифт(10,30,400)');

end.

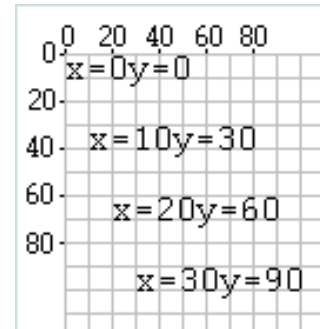
Шрифт(10,0,400)
Шрифт(8,0,800)
Шрифт(10,30,400)

Определение координат активной графической позиции: Coordinates(x, y)Program

Координаты;

Var

```
i, x, y: integer;  
Begin  
  For i:=0 to 3 do  
    Begin  
      MoveTo( 10*i, 30*i );  
      Coordinates( x, y );  
      Write( 'x=', x, 'y=', y )  
    end  
  end.  
end.
```

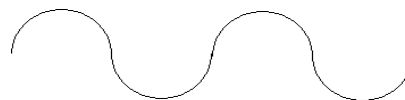


Задания к работе

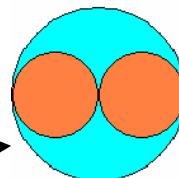
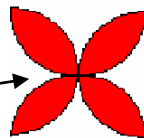
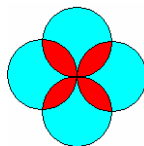
1. Используя цикл, нарисуйте рисунок.



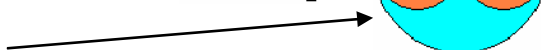
2. Используя цикл, нарисуйте рисунок.



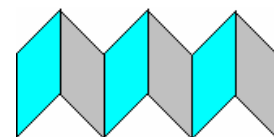
3. Создайте рисунок.



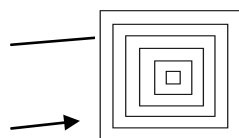
4. Создайте рисунок.



5. Создайте рисунок.




6. Создайте рисунок, используя цикл.

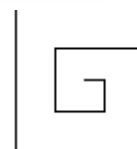


7. Создайте рисунок, используя цикл.

8. Нарисуй забор, используя цикл. Покрась его.



9. Нарисуй орнамент Меандр, используя цикл.
10. Создай проект, выводящий 100 точек внутри квадрата, используя случайные числа.
11. Создай проект, выводящий 100 разноцветных точек в левой половине окна.
12. Нарисуй серию горизонтальных линий, используя цикл.
13. Нарисуй серию вертикальных линий, используя цикл, запрашивая интервал.
14. Нарисуй треугольник, а внутри закрасенный квадрат другого цвета.
15. Нарисуй серию окружностей с общим центром, цвета окружностей меняются.
16. Нарисуй серию квадратов с общим нижним левым углом, цвета квадратов меняются.
17. Создай проект рисования объемного куба с разноцветными гранями.
18. Нарисуй сетку – серию вертикальных и горизонтальных линий.
19. Нарисуй сетку – серию взаимно перпендикулярных наклонных линий.
20. Нарисуй серию соприкасающихся окрашенных внутри окружностей, центры которых расположены на диагонали.
21. Нарисуй серию горизонтальных не соприкасающихся квадратов, закрасенных внутри в разные цвета.
22. Нарисуй шахматную доску.
23. Нарисуй квадрат и вписанный в него круг, используя различные цвета.
24. Нарисуй окружность и впиши в него треугольник, используя различные цвета.
25. Нарисуй окружность и впиши внутрь правильный многоугольник. Количество углов запрашивается.
26. Создай проект рисунка, радиус запрашивается. 
27. Создай проект рисунка спирали. Каждая очередная сторона больше на запрашиваемую в начале величину.
28. Создай проект, в котором две вертикальные линии разного



- цветадвигаются навстречу друг другу.
29. Создай проект, управления движущимся кругом.

Практическая работа № 7

Тема: Использование процедур и функций в Паскале.

Цель работы:

- Формирование и отработка умений и навыков работы по использованию процедуры функций в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

В программировании часто возникают ситуации, когда одну и ту же группу операторов, реализующих определенную цель, требуется повторить без изменений в нескольких местах программы. Чтобы не писать заново несколько раз один и тот же фрагмент программы в разных местах используют **подпрограммы**.

Подпрограмма – именованная, логически законченная группа операторов языка, которую можно вызывать, по мере необходимости, для выполнения любое количество раз из различных мест программы.

В Паскале существуют два вида подпрограмм: **Процедуры** и **Функции**.

Каждая процедура или функция должна быть предварительно описана в разделе описания процедур и функций. Для использования процедуры необходимо написать оператор вызова.

Описание процедуры состоит из заголовка процедуры и тела процедуры. Заголовок включает служебное слово **Procedure**, имя процедуры и заключенный в круглые скобки список параметров (переменных) с указанием их типа. Например:

Program ab;

Var A, B, C : integer;

Procedure KUB(X : integer; Var Y, Z : integer); ← Заголовок процедуры

Begin

Y := X*X; ← Тело процедуры

Z := X+X;

End;

Begin ← Начало программы

Write(' Укажи число – '); ReadLn(A);

KUB(A, B, C); ← Вызов процедуры

WriteLn(' Квадрат = ',B:6, ' Сумма = ',C:4);

Write(' Укажи число – '); ReadLn(A);

KUB(A, B, C); ← Вызов процедуры с фактическими параметрами.

WriteLn(' Квадрат = ',B:6, ' Сумма = ',C:4);

End.

При использовании процедур количество, типы данных и их порядок следования формальных и фактических параметров должны совпадать. Иногда раздел описания входных и выходных параметров процедуры может отсутствовать. Например, в случае, если процедура просто должна выдавать какое-то сообщение на экран:

```
procedure Print;  
begin  
    writeln(' Процедура выполнена ! ');  
    Writeln(' Привет ! ');  
end;
```

Отметим, что функций и процедур в программе может быть несколько.

Рассмотрим пример на использование процедуры при вычислении Факториала произвольного числа, например 6 ($6! = 1*2*3*4*5*6$)

Program ab;

Var A, J, F : integer;

Procedure Fakt(X : integer; Var Y : integer);

← Заголовок процедуры

Var K : integer;Begin

Y := 1;

← Тело процедуры

For K :=2 to X do

Y := Y*K;

End;

Begin

← Начало программы

For J :=1 to 4 do

Begin

Write(' Укажи число – '); ReadLn(A);

Fakt(A, F);

← Вызов процедуры

WriteLn(' Факториал ',A:2,' = ',F:6);End;

End.

Функции в языке Паскаль:

Функции описываются в том же разделе программы, где и процедуры, и могут чередоваться с ними. Для этого используется служебное слово **function** Все замечания относительно входных и выходных параметров процедур верны и для функций. Различие между

процедурой и функцией заключается в том, что Имя_функции принимает какое-либо значение, тогда как Имя_процедуры - нет. В теле функции всегда должен быть один оператор, присваивающий значение имени Функции. Примеры описаний Функций:

```
Function Max(X, Y: Real) : Real;  
Function Z(X : Real; Y : Real) : Real;
```

Рассмотрим пример на использование Функции при вычислении Факториала числа.

```
Program ab;
```

```
Var A, J, X, F : integer;
```

```
Function Fakt(Var X : integer) : integer;
```

← Заголовок Функции

```
Var K, Y : integer;
```

```
Begin
```

```
    Y := 1;
```

← Тело Функции

```
    For K :=2 to X do
```

```
        Y := Y*K;
```

```
        Fakt := Y;
```

```
End;
```

```
Begin
```

← Начало программы

```
    For J :=1 to 4 do
```

```
        Begin
```

```
            Write(' Укажи число – ');      ReadLn(A);
```

```
            WriteLn(' Факториал ',A:2,' = ',Fakt(A):6);
```

← Печать результата

```
        End;
```

```
End.
```


Рассмотрим использование Функции при вычислении значения математического выражения. Пример: дана функция $y(x)=5\cos(x)+3\sin(4x)-2x^2$.

Запишем для нее функцию на Паскале.

Program ab; Var

J : integer;

T : Real;

Function Y(Var X : Real) : Real;

← Заголовок Функции

Begin

Y:=5*Cos(X)+3*Sin(4*X)-2*Sqr(X);

End;

Begin

← Начало программы

For J :=1 to 3 do

Begin

Write(' Укажи X – '); ReadLn(T);

WriteLn(' Y = ',Y(T):8:2);

← Печать результата

End;

End.

Задания к работе

4. Запиши программу нахождения наибольшего числа из трех указанных, используя процедуру или функцию.
5. Запиши программу вычисления по формуле, используя процедуру или функцию модуля числа.
$$a = \frac{|x - y|}{10x + |xy + x + y|}$$
6. Запиши программу нахождения **НОД** (наибольший общий делитель) для двух указанных чисел, используя функцию или процедуру.
7. Найди разность между максимальным и минимальным элементами массива, с использованием процедуры или функции.
8. Заполни массив случайными целыми положительными и отрицательными числами с использованием процедуры или функции.
9. Подсчитайте кол-во и сумму всех положительных элементов числового массива с использованием процедуры или функции.
10. Введите произвольное число. Заполните массив. Подсчитайте количество элементов, которые меньше указанного числа с использованием процедуры или функции.
11. Заполни массив целыми случайными числами. Подсчитай сумму всех четных чисел с использованием процедуры или функции.
12. Заполни массив целыми случайными числами. Подсчитай сумму всех простых чисел с использованием процедуры или функции.
13. Заполни массив целыми случайными числами, покажи его. Поменяй местами числа всех четных и нечетных ячеек с использованием процедуры или функции. Покажи все элементы после перестановки.

14. Заполни массив случайными дробными числами. Первую половину массива заполни дробной частью числа а вторую половину массива целой частью с использованием процедуры или функции.
15. Заполни массив целыми случайными числами. Покажи числа. Помести элементы в обратном порядке с использованием процедуры или функции.
16. Заполни массив случайными целыми числами. Сдвинь все элементы направо на 1 число а последний элемент перенеси в 1-ю клетку массива с использованием процедуры или функции.
17. Заполни массив случайными, не повторяющимися числами. Используй функцию или процедуру.

Практическая работа № 8

Тема: Обработка текстовых величин в Паскале.

Преобразование типов данных.

Цель работы:

- Формирование и отработка умений и навыков работы по использованию текстовых данных и преобразованию типов данных в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Практически любая программа содержит обработку текста (строки) или преобразование типов данных. Любой символ занимает в памяти один байт. Значение символьной переменной – это символ (символы), заключенные в апострофы. Значение (длина текста) ограничена 255 любыми символами, включая пробел, цифры и русский алфавит.

Символы, как и любые однотипные данные, могут быть объединены в массивы. Обработка символьных массивов практически ничем не отличается от обработки чисел. Вот примеры описания строковых (текстовых) переменных в разделе **Var**:

X : String; - Объявлен длинный текст **X**, до 255 любых символов.

Y : String[15]; - Объявлен текст **Y** длиной 15 любых символов.

Z : Char; - Объявлен текст **Z** длиной в один любой символ.

A : Array[1..20] of Char; - Объявлен массив **A** из 20 элементов из символов.

B : Array[1..30] of String; - Объявлен массив **B** из 30 элементов из слов и предложений. Для строк определены операции **конкатенации (+)** и **сравнение (=, <>, <, >, <=, >=)**

Begin

A := 'Турбо';

B := 'Паскаль';

C := 'Windows';

D := A + ' ' + B + ' ' + C;

Write(D);

If A = B Then

Write(C);

End;

Сравнение строк возможно, т.к. каждый символ в таблице кодов имеет свой номер и символы расположены по возрастанию.

Сравнение строк производится слева направо до первогонесовпадающего символа: длиннее считается та строка, в которой первый несовпадающий символ имеет больший номер в таблице кодов: 'ABD' > 'ABC'

Процедуры и функции обработки строк:

- Функция **POS(S, ST)** – функция поиска. Она определяет, с какой позиции строка

S входит в строку **ST**. Если вхождение имеет место, то результатом работы функции будет номер символа (число) в строке **ST**, с которого начинается строка **S**.

Если вхождения нет, то результат – ноль. Например:

```
ST := 'р.п. Майна'S  
:= 'Май'
```

```
Write(Pos(S,ST));
```

- Процедура **DELETE(ST, P, N)** удаляет из строки **ST** (тип String) **N** (тип Integer) символов начиная с позиции **P** (тип Integer). Например:

```
ST := 'Майна'  
Delete(ST,4,2);
```

- Процедура **INSERT(S, ST, N)** вставляет в строку **S** (тип string) подстроку **ST** (тип string) начиная с позиции **N** (тип integer).
- Функция **LENGTH(S)** возвращает длину (количество символов) текста **S**
- Процедура **STR(X, S)** преобразует числовое значение **X** (тип Real) в текст **S** (Возможно задание формата числа - **X**)
- Процедура **VAL(S, X, err)** преобразует текст **S** (тип String) в число **X** (тип Real)

Разберем два конкретных примера на обработку текста: Пример 1:

Ввести слово. Определить, является ли оно «перевертышем» (слова, читаемые одинаково в обоих направлениях называются «Палиндромами») Примеры палиндромов - «ПоП», «потоп», «Боб».

```
Var text : string;
```

```
К, J, N : integer;
```

```
Flag : Boolean;
```

```
Begin
```

```
Write(' Укажи текст – '); ReadLn(text); N :=  
Length(text);
```

```
For J := 1 to N div 2 do
```

```
  If text[J] <> text[N-J+1] Then Flag :=  
    False;
```

```
  If Flag Then
```

```
    Write(' Перевертыш !');
```

```
  Else
```

```
    Write(' Нет !');
```

```
End.
```

Пример 2:

Ввести предложение. Группа символов между пробелами считается словом. Подсчитайте количество слов в тексте. Считае

```

Var    text : string;

      K, J : integer;

Begin

  Write(' Укажи текст – '); ReadLn(text);For J
:= 1 to Length(text) do

    If text[J] = '7' Then K := K+1;

    Write(' В Вашем тексте слов - ', K:2);

```

End.

Пример 3:

Ввести текст из цифр. Встречается ли в тексте цифра «7»? Сложи все цифры текста. Например ввели текст - 2376. Ответ - «Да». Сумма = 18.

```

Var    text : string;

      J, zifra, summa, kod : integer;

      a : char; NaL_7
      : Boolean;

begin

  writeln(' Введите Текст из цифр'); read(text);
  for J:=1 to length(text) do

    begin

      a:=text[J]; if a='7' then nal_7 := true;
      val(a,zifra,kod); summa := summa + zifra;

    end;

  if nal_7 = true then

    writeln(' Текст содержит 7')

  else

    writeln(' Текст не содержит 7');

  writeln(' Сумма всех цифр = ',Summa:10);end.

```

Задания к работе

1. Создай проект, подсчитывающий количество запятых в указанном тексте.
2. Подсчитайте сколько раз встречается в вашем тексте указанный символ.
3. Подсчитайте сколько раз встречается в вашем тексте указанное слово.
4. В произвольно введенном тексте вместо одного пробела вставьте везде два пробела.
5. Среди четырехзначных целых чисел найди те, в которых все цифры различны.
6. Сколько раз встречается цифра 1 в числах от 1 до 1000 включительно.
7. Подсчитайте количество цифр в числе. Ввели - 345623. Кол-во цифр - 6.
8. Переверните введенное число. Ввели 12345, получили 54321.
9. Выведи все натуральные 2-е числа, в которых нет одинаковых цифр.
10. Вычисли сумму только четных чисел введенного числа. Ввели 19352, сумма = 14.
11. Введите число и уберите ближние повторы. Ввели 23444552 получили 23452.
12. Преобразуйте введенное число в максимально число. Ввели 528, получили 852.
13. Преобразуйте введенное число в минимальное. Ввели 528, получили 258.
14. Вводите число и удвойте каждое число. Ввели 233451 получили 223333445511.
15. Замените все цифры в произвольном тексте на символ «+».
16. Замените текст «Мин» на «Мах» в произвольном тексте.
17. Подсчитайте общее количество букв «а» и «б» в вашем тексте.
18. Замените в произвольном тексте все буквы «а» на «б», а все буквы «б» на «а».
19. Подсчитайте сколько раз в произвольном тексте встречается слово «Майна».
20. Выясните, есть ли в предложении все буквы, входящие в указанное слово.
21. Удали слова, содержащие хотя бы одну цифру.
22. Поменяй местами самое длинное и короткое слова.
23. Из заданного текста удалите те его части, которые заключены в скобки.
24. Найдите самое длинное и самое короткое слова в произвольном тексте.
25. В заданном тексте подсчитай наибольшее количество подряд идущих пробелов.
26. Введи произвольное слово. Удали из него все повторяющиеся символы.
27. Из заданного текста выбери символы, встречающиеся только один раз.
28. Введите предложение и удалите из него все лишние пробелы. Покажите результат.
29. Подсчитайте сколько раз встречается каждая буква вашего текста.
30. Введите текст. На какую букву начинается больше всего слов этого текста.
31. Введите слово. Получи новое слово, где буквы расположены в алфавитном порядке.

Практическая работа № 9

Тема: Работа с файлами в языке ABC Паскаль.

Цель работы:

- Формирование и отработка умений и навыков по работе с файлами в среде Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Существуют три вида файлов – типизированные, бестиповые, и текстовые, позволяющие считывать большие объемы данных непосредственно с диска, не вводя их с клавиатуры.

Текстовые файлы состоят из любых символов. Они организуются по строкам, каждая из которых заканчивается символом «конец строки». Конец файла обозначается символом «конец файла». При записи информации в текстовый файл, просмотреть который можно любым текстовым редактором, все данные преобразуются в символьный тип и хранятся в этом виде.

Типизированные файлы состоят из компонентов одного типа данных, число которых заранее не определено и может быть любым. Они заканчиваются символом «конец файла», хранятся в двоичных кодах и не просматриваются с текстовыми редакторами.

Бестиповые файлы записываются и считываются блоками определенного размера. В них могут храниться данные любого вида и структуры.

Текстовый файл описывается так:

Var F : text;

Для начала работы с файлом любого типа необходимо связать файловую переменную в программе с файлом на диске. Для этого используют процедуру **assign(f,d)**, где **f** - имя файловой переменной, а **d** - полное имя файла на диске (файл должен находиться в текущем каталоге при условии, что к нему специально не указывается путь). Например:

Var f : text;

Begin

Assign(f, 'd:\turboPascal\tmp\abc.dat');

Рассмотрим несколько примеров работы с файлами:

Пример 1: Записать N действительных чисел в файл

Program ab;

Var f : Text;

A : real;

J, N : Integer;Begin

Assign(f, 'd:\abc.txt');

- Связываем файловую переменную с файлом

Rewrite(f);

- Открываем пустой файл для записи.

Read(N);

- Указываем кол-во элементов в файле

For J := 1 to N doBegin

Write(' a = '); Read(a);	- В цикле вводим очередной элемент
Write(f, a);	- и записываем его в файл abc.Dat
End;	
Close (f);	- Закрываем файл. Делаем это обязательно !
End.	

Пример 2:

В корне диска D: находится файл ABC.DAT дробных чисел. Нужно вывести его содержимое и подсчитать кол-во элементов

Program ab;	
Var f : Text;	
A : real;	
N : Integer;	
Begin	
Assign(f, 'd:\abc.txt');	- Связываем файловую переменную с файлом
Reset(f);	- Открываем файл для чтения.
Read(N);	- Указываем кол-во элементов в файле
N := 0	
While Not eof(f) do	- Работаем внутри цикла пока не достигнут
Begin	конец открытого файла.
 ReadLn(f, a);	- Считываем очередной элемент из файла
 N := N + 1;	
 WriteLn(N, '-й = ', a:10:4);	- Выводим на экран элемент a с номером N
End;	
WriteLn;	
End. WriteLn(' В файле элементов -', N);	- Просто выводим N
Close (f);	- Закрываем файл.

При программировании бывают весьма полезными следующие функции Паскаля для работы с файлами:

Функция **Filesize(f)** - возвращает число реальных элементов открытого файла **f**.

Процедура **Seek(f, n)** – устанавливает указатель в открытом файле **f** на компонент с номером **n** (Нумерация с нуля). Затем значение может быть считано.

Процедура **Seek(f, n)** не работает с текстовыми файлами.

Для текстовых файлов можно пользоваться процедурой **Append(f)**. Применяется для открытия уже существующих текстовых файлов для будущего добавления в них текстовой информации в конец файла. В текстовых файлах операторы **Write** записывают ин-

формацию целого, дробного, символьного типа преобразовав их предварительно в тип **String**. Оператор **WriteLn** в добавок добавляет символ конца строки.

Задания к работе

Во всех заданиях составить две программы. Первая формирует файл исходных данных. Вторая – считывает данные из созданного файла, выполняет необходимые действия.

1. Запиши список своей группы по информатике, с указанием пола.
Запиши вторую программу по подсчету количества лиц мужского и женского пола.
2. Запиши программу записи в файл 20 случайных целых чисел, покажи числа.
Создай вторую программу по удвоению всех четных записанных чисел с показом.
3. Запиши программу записи в файл 30 дробных случайных чисел.
Создай программу, записывающую округленные числа в файл с показом чисел.
4. Запиши программу записи в файл 40 дробных случайных чисел.
Создай программу, записывающую целую часть чисел в файл с показом чисел.
5. Запиши программу записи в файл 40 целых случайных чисел.
Создай программу нахождения среднего арифметического всех чисел. Замени все числа меньше среднего арифметического самим средним арифметическим.
6. Запиши список своей группы по информатике.
Запиши вторую программу по подсчету количества имени «Оля».

Практическая работа № 10

Тема: Работа с записями в языке ABC Паскаль.

Цель работы:

- Формирование и отработка умений и навыков по обработке таблиц помощью записей в среде программирования Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Запись – это структурированный тип данных, состоящий из определенного числа компонентов, которые называются **полями записи**. Они могут иметь разный тип. Описание переменной типа запись начинается ключевым словом **Record**, за которым следует список полей с указанием их типов. Заканчивается описание служебным словом **End**.

Примеры: **Var**

или **Type**

Name : Record

Abc = Record;

X : string;

A, B : String;

Y : Real;

C, D : Real;

Z : Integer;

E : array[1..30] of integer;

End;

End;

Var

SKL : ABC

Для того чтобы обратиться к полю записи, необходимо указать имя переменной и через точку – имя поля. Например: **SKL .C :=SKL . D + 122;**

Записи часто используют при работе с таблицами, где каждая запись – это одна строка таблицы. Следовательно, для обработки всей таблицы, необходимо использовать массивы записей. Например:

Type

Ved = Record; Fio

: String[25];data :

Integer; Zarplata :

Real;

End;

Var

A : array[1..30] of Ved;

Для обращения к некоторому полю например к полю Fio в пятой строке таблицы достаточно указать соответствующий элемент массива - **A[5] . Fio**

Рассмотрим пример :

Program ab;

Type TabL = Record

Fio : String[24];

Klass : String[4];

Info : Integer;

End;

Var I, J : Integer; A:TabL;

Sr_BaLL : Real;

Mas : array[1..3] of TabL;

Begin

Sr_BaLL := 0;

For i:=1 to 3 doBegin

Write(i, ' Уч-ся - '); ReadLn(Mas[i].Fio);

Write(' Класс '); ReadLn(Mas[i].Klass);

Write(' Оценка '); ReadLn(Mas[i].Info);

Sr_BaLL := Sr_BaLL + Mas[i].Info;

End;

{ Упорядочение массива записей в алфавитном порядке фамилий }

For i :=1 to 2 do

For J :=1 to 2 do

If Mas[J].Fio > Mas[J+1].Fio ThenBegin

A := Mas[J];

Mas[J] := Mas[J+1];

Mas[J+1] := A;

End;

{ Вывод результатов }

WriteLn(' ----- ');

WriteLn(' Уч-ся: Класс: Оценка:');

WriteLn(' ----- ');

О каждом учащемся есть информация:

- ✓ Фамилия, Имя
- ✓ Класс
- ✓ Оценка за четверть по Информатике

Заполним таблицу, отсортируем по алфавиту, выведем ее на экран, подсчитаем средний балл по предмету.

For i := 1 to 3 do begin

Write(Mas[i].Fio:15);

Write(Mas[i].Klass:9);

WriteLn(Mas[i].Info:9);

End;

WriteLn(' ----- ');

Write(' Средний балл = ',Sr_BaLL/3:6:2);

End.

Задания к работе

1. Сформируй записи пропусков уроков учащимися разных классов. Отсортируй записи по болезням или кол-ву пропущенных занятий. Подсчитай среднее кол-во пропусков по школе и по классу.
2. О сотрудниках некоторой фирмы известна информация: Фамилия, год рождения, стаж работы, оклад. Сформируйте таблицу записей со всей информацией, причем добавьте 10 % к окладу, работающим более 10 лет и 15 % к окладу работающим более 20 лет.
3. Известна информация о клиентах банка: Фамилия, первоначальная сумма вклада, срок вклада в месяцах. За каждый месяц банк начисляет 1 % от суммы в начале месяца. Подсчитай сумму вклада клиентов, подлежащую выплате каждому клиенту в конце срока. Сформируй таблицу записей.
4. Известен список проданных товаров с ценами и количеством. Сформируйте таблицу записей стоимости всех проданных товаров. Отсортируйте товары по стоимости.

Практическая работа № 11

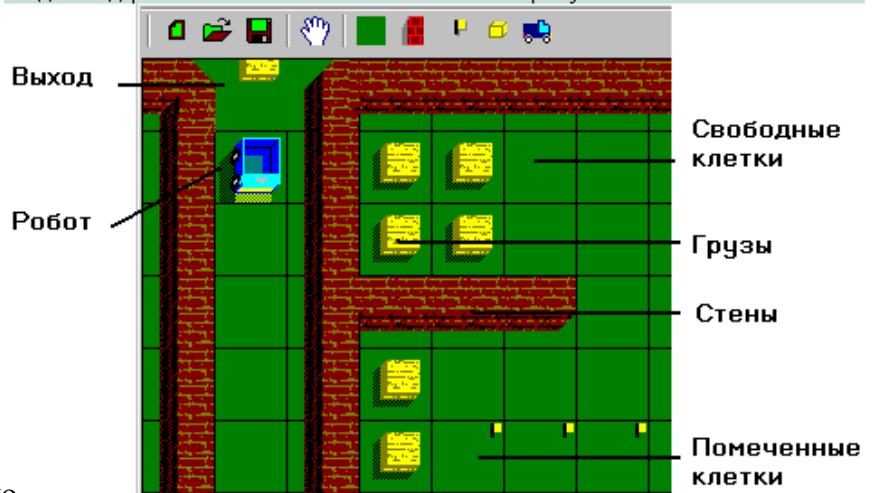
Тема: ABC Паскаль АЛГО. Управление Роботом.

Цель работы:

- Формирование и отработка умений и навыков по управлению и моделированию поведения Робота в среде ABC Паскаль.
- Развивать познавательный интерес к предмету, к изучаемой теме.

Краткая теория

Робот-грузчик изображен в виде маленького грузовика. Он движется только по вертикали или по горизонтали. Шаг перемещения - одна клетка склада. Вид робота и типы клеточек показаны на рисунке.



Робот может перевозить один груз, положить груз на свободную клеточку, а также пометить клеточку флажком или стереть метку. Исключением является клеточка выход. Робот не может

заходить на эту клеточку, но может брать с нее и класть на нее произвольное количество грузов.

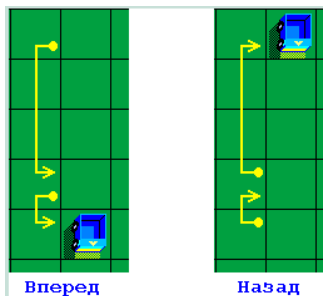
Перемещения робота : Робот Вперед(n) и Робот Назад(n)

Где n задает количество клеток перемещения робота. Если количество клеточек не указано, то робот перемещается на одну клеточку. При этом скобки употреблять нельзя. Движение осуществляется в том направлении, в котором ориентирован робот в момент обращения к процедуре.

Если на пути встречается стена или груз, то выдается сообщение об ошибке.

Program Движение;
Begin

```
RobotForw( 3 );  
RobotForw;  
RobotBack;  
RobotBack( 3)
```

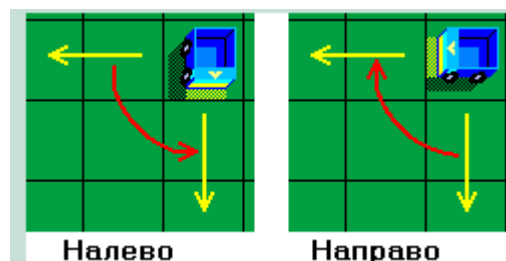


end.

Повороты робота : Робот Налево(n) Робот Направо(n):

Если аргумент не указан, то робот поворачивается на 90° , а если указан целый параметр n , то робот повторяет поворот заданное число раз. Обратите внимание, что направление поворота задается относительно робота, а не относительно наблюдателя.

Program Поворот;
Begin



```

RobotLeft;
ReadLn;
RobotRight

```

end.

Проверка клеток:

Робот может двигаться только по свободным клеточкам. Для проверки соседних клеток используют логические функции СвободноВперед, СвободноПозади, СвободноСлева, СвободноСправа:

Функции не имеют параметров и возвращают истину (true), если клеточка свободная или помечена, и значение (false), если - груз или стена. Направление опрашивания задается относительно робота, а не относительно наблюдателя.

```

Program Проверка;
Begin
  While FreeForw do
    RobotForw

```



end.

В результате выполнения программы робот двигается вперед, пока не встретит стену или груз.

Локатор - детальное опрашивание обстановки перед роботом.

В зависимости от ситуации в направлении движения робота функция возвращает одно из таких целых значений:

- 0 - перед роботом стена;
- +1 - перед роботом груз;
- +2 - перед роботом выход;
- N - перед роботом N свободных клеточек;

```

Program Локатор;

```

```

Begin

```

```

  Write( 'Перед роботом ' );
  Case InFront of

```

```

    0: Write( 'стена' );

```

```

    1: Write( 'груз' );

```

```

    2: Write( 'выход' );

```

```

  else

```

```

    Write( - InFront, ' свободных клеточек' );

```

```

  end;

```

end.

Пометить Клетку

Робот может пометить клетку, на которой находится, флажком. Для этого надо обратиться к процедуре Обозначить: Если клеточка уже обозначена, то команда игнорируется роботом. Если клетка, в которой робот, обозначена, то функция возвращает истину (true), если не обозначена - значение (false). Убирают метку процедурой СтеретьМетку:



```

Program Пометить;
Var
    i: integer;
Begin
    For i:=0 to 10 do
        Begin
            RobotForw;
            If i mod 2 = 0 then Select

                end;
            RobotLeft( 2 );
            ReadLn;
            For i:=0 to 10 do
                Begin
                    If Selected then Unselect;
                    RobotForw

                end
            end.

```

Взять груз: Take

По команде **Взять** робот берет груз, который находится в клетке непосредственно перед ним. Если груза нет или робот уже загружен, то выдается сообщение об ошибке.

Положить груз: Put

Робот кладет груз на клетку непосредственно перед собой. Если робот пустой или клетка перед ним занята, то выдается сообщение об ошибке. Если клетка, на которую кладут груз, помечена, то метка удаляется.

Робот может взять груз если он не загружен, и положить груз, если загружен.

Для проверки, пустой ли робот, обращаются к функции **Empty**

Если робот пустой, то возвращается значение (true), если нет значение (false)

```

Program Проверка;
Begin

```

```

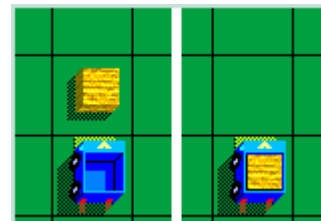
    RobotLeft( 2 );
    Take;
    RobotLeft( 2 );

```

```

    If FreeForw and not Empty then Put;
end.

```

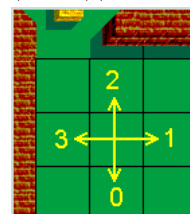


Направление Робота (Ориентация): Direction

Можно проверить ориентацию робота относительно наблюдателя, функцией Direction

В зависимости от ориентации робота функция возвращает одно из таких значений:

- 0 - робот направлен вниз (на юг);
- 1 - робот направлен вправо (восточнее);
- 2 - робот направлен вверх (на север);



3 - робот направлен влево (на мероприятие);

Текущие координаты робота: RobotCoord

Процедура возвращает переменной **X** значение номера столбца, а **Y** - номера строки, на пересечении которых находится робот. Нумерация строк ведется сверху вниз, а нумерация столбцов - слева направо.

Program Координаты;

var

x, y: integer;

Begin

RobotCoord(x,y);

WriteLn('Робот на пересечении ', x, '-го столбца и ', y, '-й строки');

Write('и направлен на ');

Case Direction of

0: Write('юг');

1: Write('восток');

2: Write('север');

3: Write('запад');

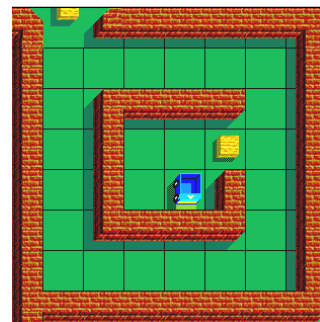
end

end.



Задания к работе

1. Создайте свой лабиринт, используя встроенные инструменты робота.
2. Выберите ручное управление роботом и выполните 5-6 различных команд.
3. Напишите программу обхода роботом всего поля по периметру.

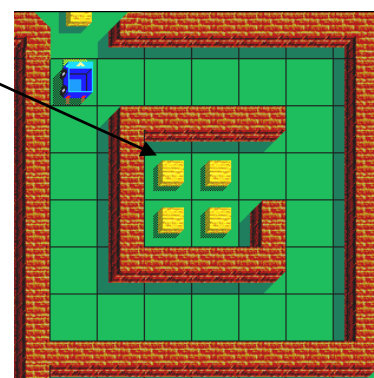


4. Положите грузы во все клетки первого столбца.
5. Положите грузы через клетку первой строки.

6. Есть некий лабиринт. Верните Робота в начало. →

7. Заполни склад грузами по схеме.

8. На поле расположены произвольно несколько грузов, соберите их.



9. На поле расположены произвольно несколько грузов, соберите их, а клетки пометьте.

10. На поле расположены произвольно несколько грузов. Напишите программу подсчета роботом их количества.

11. На поле есть одинаковое количество грузов и помеченных клеток. Перенеси грузы в помеченные клетки.

12. Положите грузы ниже главной диагонали.

13. Положите грузы выше главной диагонали.

14. Положите грузы в шахматном порядке

Рецензия


на рабочую программу внеурочной деятельности по информатике «Программирование на языке Pascal» для учащихся 8 класса, разработанную учителем информатики МАОУ СОШ № 34 Гайнутдиновым Равилем Ринатовичем.

Рабочая программа разработана на основе: приказа Министерства просвещения Российской Федерации от 31.05.2021 № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования»; примерной программы воспитания - протокол Федерального учебно-методического объединения по общему образованию № 3/22 от 23.06.2022; примерной основной образовательной программы основного общего образования - протокол Федерального учебно-методического объединения по общему образованию № 1/22 от 18.03.2022., программы УМК Босова Л.Л./Босова А.Ю. Информатика, 7-9 класс, программы Алгоритмы и исполнители К.Ю.Полякова. Элективный курс рассчитан на один год обучения. Образование осуществляется в виде теоретических и практических занятий для учащихся – 1 час в неделю, всего – 34 часа за год.

Курс ориентирован на учащихся 8-х классов, которые желают изучать языки программирования и дает им возможность получить практические навыки программирования, которые они применяют на олимпиадах и итоговой аттестации (ОГЭ, ЕГЭ). В содержании рабочей программы учителем информатики Равилем Ринатовичем умело и концентрично изложен перечень тем, которые должны освоить обучающиеся 8-х классов. Учителем предлагается перечень практических работ в качестве закрепления теоретических навыков, которые выходят за рамки изучения школьного курса. Термины и задачи, в представленных учителем темах, доступны для понимания обучающихся, и носят как ознакомительный, так и основной вид обучения.

Новизна курса «Программирование на языке Pascal» для обучающихся 8-х классов заключается в содержании рабочей программы, включающей новые знания, представляющие высокую степень актуальности и полезности для выбора школьниками профиля обучения в старших классах и других целей профильной подготовки. Автор акцентирует внимание на том, что цель программы – развитие научно-технического мышления, творческого воображения. Программа направлена на развитие индивидуальной траектории образования каждого обучающегося, соответствует требованиям ФГОС ООО. Программа предполагает формирование базовых компетентностей современного человека: информационной, коммуникативной, самоорганизации, самообразования.

Актуальность рабочей программы «Программирование на языке Pascal» служит ориентиром для решения проблем профилизации при изучении предмета за счет расширения теоретической части курса информатики,

Директор
МАОУ СОШ № 34  А.А.Артеменко

решения разнообразных практических задач. Элективный курс «Основы программирования», 8 класс предусматривает интеграцию между предметами «Математика» и «Информационные технологии».

Для поднятия интереса к изучению программирования, учителем разработан цикл соответствующих заданий разного уровня сложности теоретического и практического формата, которые необходимо решить, как с помощью персонального компьютера, так и без... Программа «Программирование на языке Pascal» для учащихся 8-х классов соответствует требованиям, предъявляемым к программам элективных курсов, позволяет расширить базовый курс изучения информатики в школе, реализовать способности обучающихся.

Рабочая программа внеурочной деятельности «Программирование на языке Pascal», разработанная учителем информатики МАОУ СОШ № 34 Гайнутдиновым Равилем Ринатовичем, содействует сохранению единого образовательного пространства, обладает мощным потенциалом по формированию интереса обучающихся к предмету и может быть рекомендована для использования в МАОУ СОШ № 34 г. Новороссийска во внеурочной деятельности.

28 февраля 2024 г.

Главный специалист МКУ ЦРО

М.А. Сарнавская

Подпись специалиста, заверяю.

Директор МКУ ЦРО



Е.Л. Тимченко

Директор

МАОУ СОШ № 34

А.А.Артеменко



МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ И МОЛОДЕЖНОЙ ПОЛИТИКИ
КРАСНОДАРСКОГО КРАЯ

Государственное бюджетное образовательное учреждение
дополнительного профессионального образования
«Институт развития образования» Краснодарского края
(ГБОУ ИРО Краснодарского края)

УДОСТОВЕРЕНИЕ О ПОВЫШЕНИИ КВАЛИФИКАЦИИ

231500023836

Настоящее удостоверение свидетельствует о том, что _____
Гайнутдинов Равиль Ринатович

с « 04 » мая 2023 г. по « 12 » мая 2023 г.

прошел(а) повышение квалификации в _____
ГБОУ ИРО Краснодарского края
(наименование образовательного учреждения (подразделения) дополнительного профессионального образования)
по теме: «Деятельность учителя по достижению результатов
обучения в соответствии с ФГОС с использованием
цифровых образовательных ресурсов»

в объеме: 48 часов
(количество часов)

За время обучения сдал(а) зачеты и экзамены по основным дисциплинам программы:

Наименование	Объем	Оценка
Государственная политика в сфере образования. Внедрение обновленных ФГОС	6 часов	зачтено
Цифровые образовательные ресурсы как средство реализации ФГОС	14 часов	зачтено
Современный урок с использованием ЦОР: технологические особенности проектирования и проведения в условиях внедрения обновленных ФГОС: обшедидактические и предметные особенности	28 часов	зачтено

Прошел(а) стажировку в (на) _____
(наименование предмета,

организации, учреждения)

Итоговая работа на тему: _____



Директор
10963/23 СОШ № 34
Регистрационный номер № _____

А.А. Артемьев

М.П. _____
Ректор
Секретарь

Т.А. Гайдук
А.И. Илющенко

Дата выдачи 12 мая 2023 г.

РОССИЙСКАЯ ФЕДЕРАЦИЯ

Министерство просвещения
Российской Федерации

федеральное государственное бюджетное образовательное
учреждение высшего образования
«Армавирский государственный
педагогический университет»

УДОСТОВЕРЕНИЕ

О ПОВЫШЕНИИ КВАЛИФИКАЦИИ

Серия 23У №1767007216

Документ о квалификации

Регистрационный номер

0568/22

Город
Армавир

Дата выдачи

08.11.2022 г.

Директор

М.А. Артемко



Руководитель

Секретарь

в объёме

108 часов

С.А. Родин

Д.С. Шевелева

Настоящее удостоверение свидетельствует о том, что

Гайнутдинов

Равиль Ринатович

прошел (а) повышение квалификации в
федеральном государственном бюджетном образовательном
учреждении высшего образования
«Армавирский государственный педагогический
университет»

по дополнительной профессиональной программе

*"Электронная информационно-образовательная
среда образовательной организации"*

17.10.2022 г. - 07.11.2022 г.

РОССИЙСКАЯ ФЕДЕРАЦИЯ

Министерство просвещения

Российской Федерации

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Армавирский государственный

педагогический университет»

УДОСТОВЕРЕНИЕ

О ПОВЫШЕНИИ КВАЛИФИКАЦИИ

Серия 23У № 1767009224

Документ о квалификации

Регистрационный номер

1623/23

Город

Армавир

Дата выдачи

20.12.2023 г.

Директор

МАОУ СОШ № 34

А.А.Артемько



Настоящее удостоверение свидетельствует о том, что

Гайнутдинов

Равиль Ринатович

прошел (а) повышение квалификации в

федеральном государственном бюджетном образовательном

учреждении высшего образования

«Армавирский государственный педагогический

университет»

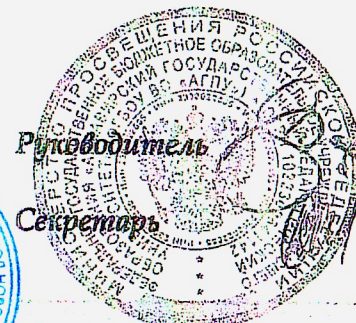
по дополнительной профессиональной программе

"Актуальные вопросы теории и методики преподавания информатики в соответствии с обновленными ФГОС ООО и ФГОС СОО"

12.12.2023 г. - 19.12.2023 г.

в объёме

36 часов



С.А. Родин

Д.С. Шевелева