

Министерство образования Красноярского края
краевое государственное бюджетное профессиональное образовательное учреждение
«Красноярский колледж радиоэлектроники и информационных технологий»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущей и промежуточной аттестации

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

для студентов специальности

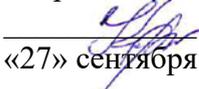
09.02.07 Информационные системы и программирование

г. Красноярск, 2022

Составлена в соответствии с федеральным государственным образовательным стандартом СПО по специальности 09.02.07 Информационные системы и программирование

ОДОБРЕНО

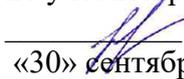
Старший методист

 Т. В. Клачкова

«27» сентября 2022 г.

УТВЕРЖДАЮ

Заместитель директора
по учебной работе

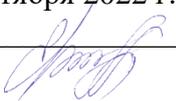
 М. А. Полютова

«30» сентября 2022 г.

РАССМОТРЕНО

на заседании цикловой комиссии укрупненной группы специальностей 09.00.00 Информатика и вычислительная техника №1

Протокол №1 от «26» сентября 2022 г.

Председатель ЦК  Е.А. Ивашова

АВТОР: Казанкова А.А., преподаватель высшей квалификационной категории КГБПОУ «ККРИТ»

Ивашова Е.А., преподаватель высшей квалификационной категории КГБПОУ «ККРИТ»

СОДЕРЖАНИЕ

	стр.
1 ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ	4
2 ОРГАНИЗАЦИЯ КОНТРОЛЯ И ОЦЕНКИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	9
3 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ	9
4 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕ- СТАЦИИ	19
5 ПЕРЕЧЕНЬ ПЕЧАТНЫХ ИЗДАНИЙ, ЭЛЕКТРОННЫХ ИЗДАНИЙ (ЭЛЕК- ТРОННЫХ РЕСУРСОВ), ДОПОЛНИТЕЛЬНЫХ ИСТОЧНИКОВ	20

1 ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1 Область применения

Фонд оценочных средств предназначен для проверки результатов освоения учебной дисциплины ОП.04. Основы алгоритмизации и программирования, которая является обязательной частью профессионального учебного цикла программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности СПО 09.02.07 Информационные системы и программирование.

Фонд оценочных средств позволяет оценить:

1.1.1. Освоенные умения и усвоенные знания:

<i>Освоенные знания</i>	<i>Усвоенные умения</i>
З 1. понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	У 1. разрабатывать алгоритмы для конкретных задач;
З 2. эволюцию языков программирования, их классификацию, понятие системы программирования;	У 2. использовать программы для графического отображения алгоритмов;
З 3. основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	У 3. определять сложность работы алгоритмов;
З 4. подпрограммы, составление библиотек подпрограмм;	У 4. работать в среде программирования;
З 5. объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	У 5. реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
	У 6. оформлять код программы в соответствии со стандартом кодирования;
	У 7. выполнять проверку, отладку кода программы.

1.1.2. Освоение общих и профессиональных компетенций по профессиональному модулю:

ОК 1 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 4 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5 Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9 Использовать информационные технологии в профессиональной деятельности.

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языке.

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств

ПК 1.4 Выполнять тестирование программных модулей

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода

ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Формой промежуточной аттестации в соответствии с учебным планом специальности является экзамен.

Распределение оценивания результатов обучения по видам контроля

1.2 Система контроля и оценки освоения программы учебной дисциплины (МДК)

Контролируемые элементы учебной дисциплины (темы)	Контролируемые знания, умения	Вид контроля	Форма контроля	Контрольно-оценочные материалы
Раздел 1. Основные понятия алгоритмизации.	<p>знать:</p> <ul style="list-style-type: none"> – основные принципы алгоритмизации и программирования; – общие принципы построения алгоритмов; – методы разработки алгоритмов; – правила оформления блок-схем; – логические основы алгоритмизации. <p>уметь:</p> <ul style="list-style-type: none"> – составлять блок-схемы программ. 	Текущий	Выполнение практических заданий, оформление отчета, устный опрос, проверка доклада/реферата, выполнение индивидуального задания	Типовые метод. рекомендации к практическому занятию требования к оформлению отчетов, требования к устному опросу, докладу/реферату, выполнение индивидуального задания (пункт 3)
Раздел 2. Языки и методы программирования	<p>знать:</p> <ul style="list-style-type: none"> – поколения языков программирования; – методы программирования; – общие принципы разработки программного обеспечения; – жизненный цикл программного обеспечения. 	Текущий	Выполнение практических заданий, оформление отчета, устный опрос	Типовые метод. рекомендации к практическому занятию требования к оформлению отчетов, требования к устному опросу (пункт 3)
Раздел 3. Описание и синтаксис языка программирования	<p>знать:</p> <ul style="list-style-type: none"> – описание среды программирования; 	Текущий	Выполнение практических заданий, оформление отчета,	Типовые метод. рекомендации к практическому занятию

	<ul style="list-style-type: none"> – структурированные типы данных. уметь: – обрабатывать структурированные типы данных в среде программирования. 		устный опрос, заполнение таблиц	требования к оформлению отчетов, требования к устному опросу, заполнению таблиц (пункт 3)
Раздел 4. Введение в структурированное программирование	<ul style="list-style-type: none"> знать: – способы организации подпрограмм и рекурсии; – библиотеки подпрограмм. 	Текущий	Выполнение практических заданий, оформление отчета, устный опрос	Типовые метод. рекомендации к практическому занятию требования к оформлению отчетов, требования к устному опросу (пункт 3)
Раздел 5. Программирование в объектно-ориентированной среде	<ul style="list-style-type: none"> знать: – основные принципы объектно-ориентированного программирования. 	Текущий	Выполнение практических заданий, оформление отчета, устный опрос	Типовые метод. рекомендации к практическому занятию требования к оформлению отчетов, требования к устному опросу (пункт 3)
Раздел 6. Визуализация программирования в интегрированной среде разработчика	<ul style="list-style-type: none"> знать: – интегрированную среду разработчика; – визуальное событийно-управляемое программирование. уметь: – проводить разработку оконного приложения. 	Текущий	Выполнение практических заданий, оформление отчета, устный опрос	Типовые метод. рекомендации к практическому занятию требования к оформлению отчетов, требования к устному опросу (пункт 3)
учебная дисциплина ОП.04. Основы алгоритмизации и программирования	<ul style="list-style-type: none"> уметь: – разрабатывать алгоритмы для конкретных задач; – использовать программы для графического отображения алгоритмов; 	Промежуточный	Экзамен	Контрольно-оценочные материалы для промежуточной аттестации (Пункт 4).

	<ul style="list-style-type: none">– определять сложность работы алгоритмов;– работать в среде программирования;– реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;– оформлять код программы в соответствии со стандартом кодирования;– выполнять проверку, отладку кода программы. <p>знать:</p> <ul style="list-style-type: none">– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;– эволюцию языков программирования, их классификацию, понятие системы программирования;– основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;			
--	---	--	--	--

	<ul style="list-style-type: none">– подпрограммы, составление библиотек подпрограмм;– объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.			
--	---	--	--	--

2 ОРГАНИЗАЦИЯ КОНТРОЛЯ И ОЦЕНКИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

Формой промежуточной аттестации по учебной дисциплине ОП.04. Основы алгоритмизации и программирования в соответствии с учебным планом специальности СПО 09.02.07 Информационные системы и программирование является комплексный экзамен.

Условием допуска к комплексному экзамену является положительный результат в ходе текущего контроля в процессе изучения учебной дисциплины и выполнения всех практических занятий (лабораторных работ), предусмотренных рабочей программой. Экзамен проводится в форме устного опроса, обучающегося по билету, включающему 1 теоретический вопрос и 1 практическое задание. Вопросы к экзамену охватывают наиболее значимые из тем, предусмотренных рабочей программой.

При определении уровня достижений, обучающихся на экзамене, учитывается:

- знание программного материала и структуры дисциплины;
- знания, необходимые для решения типовых задач, умение выполнять предусмотренные программой задания;
- владение методологией дисциплины, умение применять теоретические знания при решении задач, обосновывать свои действия.

При определении уровня достижений, обучающихся на экзамене, обращается особое внимание на следующее:

- дан полный, развернутый ответ на поставленный вопрос;
- показана совокупность осознанных знаний об объекте, проявляющаяся в свободном оперировании понятиями, умении выделить существенные и несущественные признаки, причинно-следственные связи;
- знание об объекте демонстрируется на фоне понимания его в системе данной дисциплины и междисциплинарных связей;
- ответ формулируется в терминах дисциплины, изложен литературным языком, логичен, доказателен, демонстрирует авторскую позицию обучающегося;
- теоретические постулаты подтверждаются примерами из практики.

Оценка «отлично» ставится за работу, выполненную без ошибок и недочетов или имеющую не более одного недочета;

- оценка «хорошо», ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета или не более двух недочетов;
- оценка «удовлетворительно» ставится в том случае, если студент правильно выполнил не менее половины работы или допустил:

- а) не более двух грубых ошибок;
- б) не более одной грубой ошибки и одного недочета;
- в) не более двух-трех негрубых ошибок;
- г) не более одной негрубой ошибки и трех недочетов;
- д) при отсутствии ошибок, но при наличии 4-5 недочетов;

- оценка «неудовлетворительно» ставится, когда число ошибок и недочетов превосходит норму, при которой может быть выставлена оценка «3», или если правильно выполнено менее половины работы.

Грубыми являются ошибки, свидетельствующие о том, что студент не усвоил основные понятия темы, не знает формул, последовательность выполнения задания, не умеет формулировать выводы по результатам расчетов.

Негрубыми ошибками являются неточности расчетов, пропуск или неполное написание формул, неполное отражение результатов исследования в выводе.

К недочетам относятся небрежное выполнение заданий, отдельные погрешности в формулировке ответа.

3 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Текущий контроль предназначен для проверки хода и качества формирования компетенций,

стимулирования учебной работы обучаемых и совершенствования методики освоения новых знаний. Он обеспечивается проведением семинаров, оцениванием контрольных заданий, проверкой конспектов лекций, выполнением индивидуальных и творческих заданий, периодическим опросом обучающихся на занятиях. Ниже приведены методические рекомендации по выполнению всех видов текущего контроля в соответствии с рабочей программой.

3.1 Методические рекомендации по подготовке реферата

Написание реферата – это более объемный, чем сообщение, вид самостоятельной работы студента. Реферативные материалы должны представлять письменную модель первичного документа – научной работы, монографии, статьи. Реферат может включать обзор нескольких источников и служить основой для доклада на определенную тему на семинарах, конференциях.

Регламент озвучивания реферата – 7-10 мин.

Затраты времени на подготовку материала зависят от трудности сбора информации, сложности материала по теме, индивидуальных особенностей студента и определяются преподавателем. Ориентировочное время на подготовку – 4 ч.

Структура реферата и требования к его оформлению подробно изложены в Приложении №2 «Положение о составлении реферата»

Критерии оценки реферата

- актуальность темы, 1 балл;
- соответствие содержания теме, 3 балла;
- глубина проработки материала, 3 балла;
- грамотность и полнота использования источников, 1 балл;
- соответствие оформления реферата требованиям, 2 балла;
 - доклад, 5 баллов;
 - умение вести дискуссию и ответы на вопросы, 5 баллов.

Максимальное количество баллов: 20.

19-20 баллов соответствует оценке «5»

15-18 баллов – «4»

10-14 баллов – «3»

менее 10 баллов – «2».

Реферат представляется к защите на листах формата А4. В исключительном случае допускается защита реферата, представленного в рукописном варианте. В тексте реферата могут содержаться рисунки, чертежи, графики прочий иллюстративный материал, необходимый для раскрытия заявленной темы. К реферату могут прилагаться фотографии, выполненные самим обучающимся.

На компьютере реферат оформляется в соответствии с требованиями ГОСТ 9327 на стандартных листах формата А4. Текст выполняется на одной стороне листа через одинарный межстрочный интервал шрифтом Times New Roman-14. Для заголовков можно выбрать иной шрифт, с использованием полужирного шрифта, курсива и подчеркивания. Если заголовок не умещается на одной строке, для него следует использовать одинарный межстрочный интервал. Заголовки выполняются без переносов с выравниванием по центру.

Следует использовать размеры полей: левое – 2,5 см, правое 1 см, верхнее – 2,5 см, нижнее 2,5 см, формат набранного материала 175х24 см. При печати текстового материала следует использовать выравнивание и автоматическую расстановку переносов слов. Абзацы в тексте начинаются отступом 12-15 мм.

Цитаты должны обозначаться указанием на первоисточник, которые оформляются в виде сносок в конце листа, на котором присутствует цитируемый отрывок.

Каждый структурный элемент реферата следует начинать с новой страницы. Разделы основной части могут быть разделены на подразделы, т.е., в свою очередь, на пункты и, при необходимости, на подпункты, которые не требуют переноса на новую страницу. Заголовки подразделов, пунктов и подпунктов следует начинать с абзационного отступа, не подчеркивая, без точки в конце. Заголовки по возможности следует делать краткими. Шрифт заголовков одного уровня должен быть единым ко всему тексту. Например, заголовки подразделов можно выполнять полужирным шрифтом, пунктов – полужирным курсивом, подпунктов – курсивом. Заголовки следует отделять от основного текста дополнительным пробелом сверху и снизу.

Нумерация страниц – сквозная, начинается с титульного листа, но номер страницы на нем не выводится. Страницы документа проставляются арабскими цифрами в правом нижнем углу без точки в конце.

3.2 Требования к оформлению отчетов по практическим занятиям

Практические работы выполняются на компьютере в соответствии с выданными методическими указаниями. Результатом выполнения работы является отчет о проделанной работе, который должен быть распечатан и сложен в специальную папку на листах формата А4, которые должны быть скреплены. Первый (титульный) лист (приложение 1) должен содержать сведения об исполнителе.

Студент должен защитить практическую работу индивидуально. Подвести итог и сформулировать основные выводы. Сдать работу преподавателю (т.е. защитить её на оценку) можно на том же занятии, на котором она выполнялась. Защита практической работы осуществляется путем частичной демонстрации проделанной работы и ответов на контрольные вопросы, приведенных в конце методических указаний.

Структура отчета практической работы:

1. Цель и задачи работы. Формулируются в соответствии с метод. указаниями.
2. Ход работы. Выполнение предложенных заданий.
3. Описание выполненной работы, сопровождаемой скриншотами.
4. Выводы.

Программа практических работ по учебной дисциплине:

РАЗДЕЛ 1. ОСНОВНЫЕ ПОНЯТИЯ АЛГОРИТМИЗАЦИИ.

- ПЗ №1. Составление блок-схем линейных алгоритмов.
- ПЗ №2. Составление блок-схем разветвляющихся алгоритмов.
- ПЗ №3. Составление блок-схем циклических алгоритмов.
- ПЗ №4. Составление блок-схем программ.
- ПЗ №5. Составление блок-схем программ.

РАЗДЕЛ 3. ОПИСАНИЕ И СИНТАКСИС ЯЗЫКА ПРОГРАММИРОВАНИЯ

- ПЗ №6. Знакомство с лексикой языка программирования Python. Основные операторы.
- ПЗ №7. Функции ввода-вывода, операции и встроенные функции языка Python.
- ПЗ №8. Составление программ разветвляющейся структуры. Множественное ветвление.
- ПЗ №9. Составление программ циклической структуры. Ошибки и исключения.
- ПЗ №10. Составление программ циклической структуры. Ошибки и исключения.
- ПЗ №11. Обработка списков и кортежей.
- ПЗ №12. Обработка массивов.
- ПЗ №13. Обработка строковых данных.
- ПЗ №14. Обработка множеств.
- ПЗ №15. Применение и обработка словарей.
- ПЗ №16. Файлы. Работа с файлами.

РАЗДЕЛ 4. ВВЕДЕНИЕ В СТРУКТУРИРОВАННОЕ ПРОГРАММИРОВАНИЕ

- ПЗ №17. Применение функций в программировании.
- ПЗ №18. Организация и использование рекурсий.
- ПЗ №19. Программирование модуля.
- ПЗ №20. Создание и использование библиотеки подпрограмм.

РАЗДЕЛ 5. ПРОГРАММИРОВАНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ СРЕДЕ

- ПЗ №21. Наследование и полиморфизм в ООП.
- ПЗ №22. Перегрузка операторов
- ПЗ №23. Декораторы
- ПЗ №24. Декораторы

РАЗДЕЛ 6. ВИЗУАЛИЗАЦИЯ ПРОГРАММИРОВАНИЯ В ИНТЕГРИРОВАННОЙ СРЕДЕ РАЗРАБОТЧИКА

- ПЗ №25. Разработка простейших программ графическим интерфейсом при помощи PyQt5.
- ПЗ №26. Простейшая обработка сигналов и слотов.
- ПЗ №27. Разработка Меню и тулбаров при помощи PyQt5.
- ПЗ №28. Разработка простейшего интерфейса при помощи PyQt5. Знакомство с основными виджетами.
- ПЗ №29. Диалоговые окна в PyQt5.
- ПЗ №30. Управление макетом в PyQt5.
- ПЗ №31. Обработка событий окна в PyQt5.
- ПЗ №32. Обработка событий клавиатуры в PyQt5.
- ПЗ №33. Обработка событий мыши в PyQt5.
- ПЗ №34. Многодокументный интерфейс в PyQt5.
- ПЗ №35. Возможности Drag & Drop в PyQt5.

- ПЗ №36. Размещение нескольких компонентов в окне.
 ПЗ №37. Работа с основными компонентами
 ПЗ №38. Многострочное текстовое поле
 ПЗ №39. Работа с датой и временем. Календарь.
 ПЗ №40. Списки и таблицы
 ПЗ №41. Модели.
 ПЗ №42. Представления.
 ПЗ №43. Создание простого калькулятора в PyQt5
 ПЗ №44. Разработка зачетного проекта. Защита проекта.
 ПЗ №45. Разработка зачетного проекта. Защита проекта.
 ПЗ №46. Разработка зачетного проекта. Защита проекта.

Экспертная оценка выполнения практических работ

Оценка «5»

- выполнил работы в полном объеме с соблюдением необходимой последовательности действий;
- проводит работу в условиях, обеспечивающих получение правильных результатов и выводов;
- соблюдает правила техники безопасности;
- в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления;
- правильно выполняет анализ ошибок.

Оценка «4» ставится, если выполнены требования к оценке 5, но допущены 2-3 недочета, не более одной ошибки и одного недочета.

Оценка «3» ставится, если

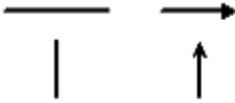
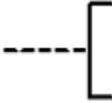
- работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы
- в ходе проведения работы были допущены ошибки.

Оценка «2» ставится, если студент совсем не выполнил работу.

3.3 Типовые задания

Устный опрос по теме «Основные принципы алгоритмизации. Виды алгоритмов.»

1. Назовите и кратко охарактеризуйте принципы программирования.
2. Назовите соответствующий графический элемент и его обозначение:

Графический элемент	Обозначение
	Начало/конец алгоритма
	Ввод/вывод данных
	Оператор цикла
	Условный оператор
	
	Решение задачи (процесс)
	

3. Что такое линейный алгоритм? Какие графические элементы могут входить в его описание?
4. Что такое линейный алгоритм? Какие графические элементы могут входить в его описание?
5. Практическое задание: построить алгоритм решения задачи:
 - а) Построить алгоритм нахождения решения

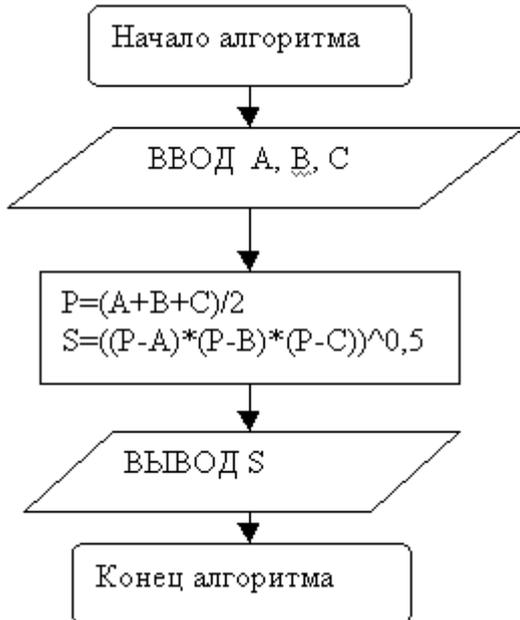
$$f(x) = 3x^2 - \frac{1}{7x^7} + \frac{3}{\sqrt[3]{x^2}} + 1$$

- б) Даны два числа А и В. Построить алгоритм поиска максимального из них

Тестовый опрос по теме «Виды алгоритмов. Построение алгоритмов».

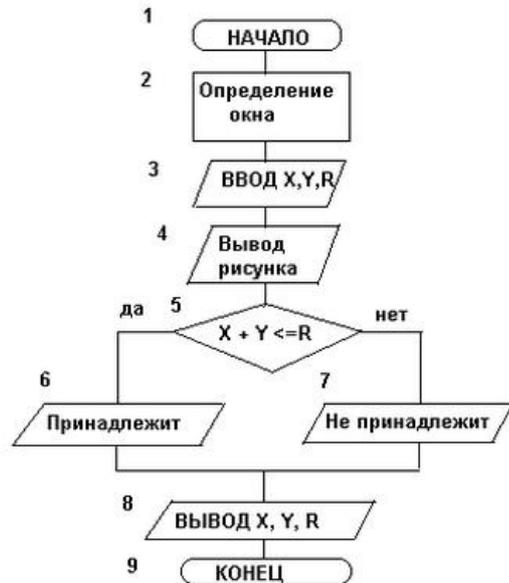
Ссылка на тест <https://forms.gle/888niyrjQeJyCJ547>

1. Определить, к какому виду алгоритмов относится алгоритм на рисунке:



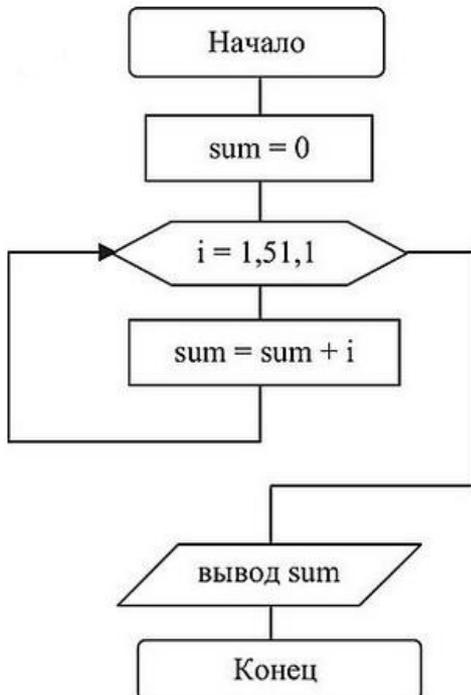
- а) цикл с постусловием
- б) цикл с предусловием
- в) разветвляющийся
- г) линейный
- д) цикл с параметром

2. Определить, к какому виду алгоритмов относится алгоритм на рисунке:



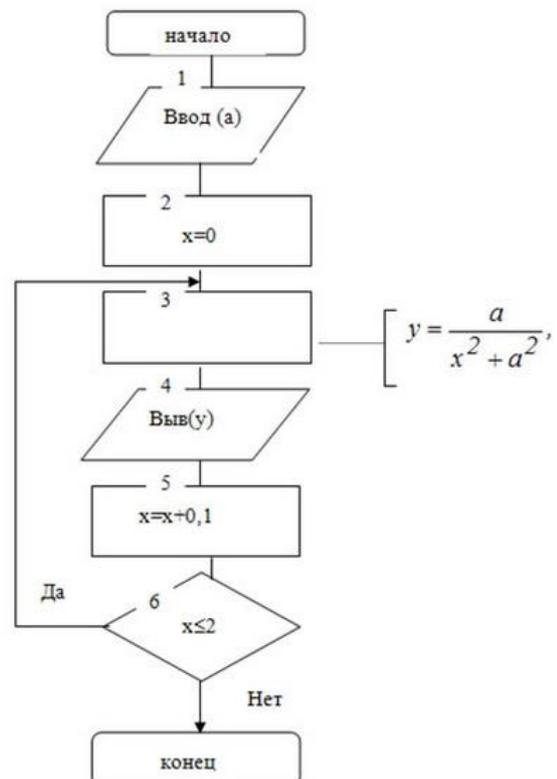
- а) цикл с постусловием
- б) цикл с предусловием
- в) разветвляющийся
- г) линейный
- д) цикл с параметром

3. Определить, к какому виду алгоритмов относится алгоритм на рисунке:

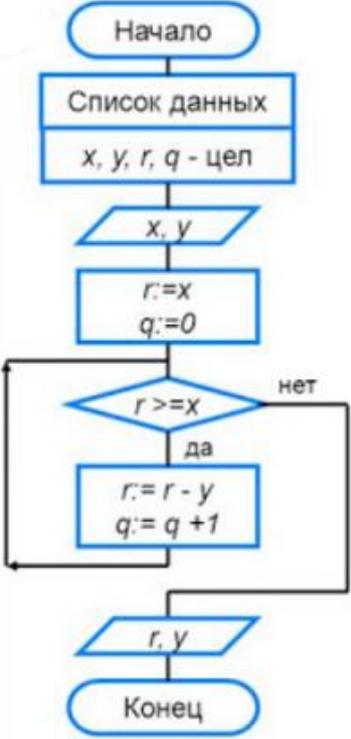


- а) цикл с постусловием
- б) цикл с предусловием
- в) разветвляющийся
- г) линейный
- д) цикл с параметром

4. Определить, к какому виду алгоритмов относится алгоритм на рисунке:



- а) цикл с постусловием
- б) цикл с предусловием

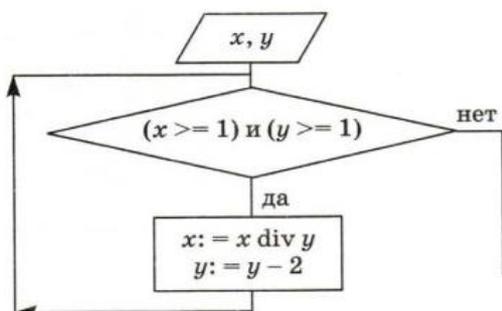
	<p>в) разветвляющийся г) линейный д) цикл с параметром</p>
<p>5. Определить, к какому виду алгоритмов относится алгоритм на рисунке:</p>  <p>а) цикл с постусловием б) цикл с предусловием в) разветвляющийся г) линейный д) цикл с параметром</p>	<p>6. Что НЕ может входить в линейный алгоритм</p> <p>а) конец алгоритма б) ввод данных в) вывод данных г) начало алгоритма д) ветвление е) процесс ж) цикл</p>
<p>7. По каким критериям производится выбор конкретного численного метода решения задачи</p> <p>а) оптимальное время решения задачи б) минимальные стоимостные затраты в) возможность использования стандартных подпрограмм г) сложность задачи д) требуемая точности вычислений е) навыки разработчика ж) оптимальное использования имеющихся ресурсов (памяти)</p>	<p>8. Принцип проектирования «от главного к второстепенному» называется</p> <p>а) Принцип нисходящего проектирования б) Структурное проектирование в) Принцип восходящего проектирования</p>
<p>9. Показанная на рисунке запись алгоритма представлена в форме:</p>	<p>10. Определите значение переменной b после выполнения алгоритма:</p> <p>a := 2 b := 4 a := 2*a + 3*b b := a/2*b</p>

Алгоритм БИТ

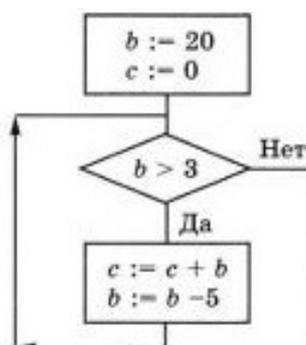
1. **чтение** a, b, c
2. **если** $a < b$ **к 4**
3. $y := b$; **к 5**
4. $y := a$
5. **если** $y > c$ **к 7**
6. $y := c$
7. **запись** y
8. **конец**

- а) табличной записи
- б) развернутой словесной форме
- в) на языке программирования
- г) на алгоритмическом языке
- д) формульной записи

11. Определите значение переменных x и y после выполнения алгоритма: (в ответ записать 2 переменных через запятую)



12. Определите значение переменной C после выполнения алгоритма

**ОТВЕТЫ:**

№ заданий	Варианты ответа
1	г
2	в
3	д
4	а
5	б
6	д, ж
7	а, б, в, д, ж
8	а
9	г
10	32
11	1, 0
12	50

Экспертная оценка выполнения тестового задания

"5" - 85% (от 15 баллов)

"4" - 65% (от 12 баллов)

"3" - 51% (от 9 баллов)

"2" - 35% (от 8 баллов и ниже)

3.4 Методические рекомендации по выполнению индивидуального задания

Индивидуальное задание по практике выполняется в той же форме, что и весь отчет по практике, а конкретно в печатном виде, каждый лист должен иметь поля: 3 см – левое, 2 см – правое, 2,5 см – верхнее и нижнее, красная строка – 1 см., нумерация страниц внизу справа. Каждый лист имеет рамку с маленьким штампом. Таблицы, диаграммы, рисунки, выполненные студентами на отдельных листах, включаются в общую нумерацию, приложения включаются в отчет без нумерации страниц. Все таблицы должны иметь содержательный заголовок. Шрифт Times New Roman, 14 с интервалом 1,5.

Индивидуальное задание должно соответствовать выбранной теме и содержать не менее 6 листов:

- введение (1-2 листа)
- содержание (2 - 3 листа)
- заключение (1 лист)
- приложения (по мере необходимости).

Содержание ИЗ

Введение.

Должна быть отражена актуальность темы для профессионального становления студента, для предприятия.

2.Содержание.

Должны быть отражены следующие вопросы:

- теоретический аспект индивидуального задания;
- практический аспект индивидуального задания в конкретной организации (предприятии);

Заключение.

Должны быть:

- анализ результатов выполнения индивидуального задания

3.5 Типовые методические рекомендации к выполнению практических заданий

Практическая работа №1

Тема: Составление блок-схем линейных алгоритмов с помощью программы Microsoft Visio.

1. Цель работы: выявить способы составления блок-схем линейных и разветвляющихся алгоритмов.

2. Оборудование, приборы, аппаратура, материалы: персональный компьютер, программа Microsoft Visio.

3. Краткие теоретические сведения:

1. ПОНЯТИЕ АЛГОРИТМА:

Одним из фундаментальных понятий в информатике является понятие алгоритма. Происхождение самого термина «алгоритм» связано с математикой. Это слово происходит от Algorithm! — латинского написания имени Мухаммеда аль-Хорезми (787—850), выдающегося математика средневекового Востока. В XII в. был выполнен латинский перевод его математического трактата, из которого европейцы узнали о десятичной позиционной системе счисления и правилах арифметики многозначных чисел. Именно эти правила в то время называли алгоритмами. Сложение, вычитание, умножение столбиком, деление уголком многозначных чисел — вот первые алгоритмы в математике. Правила алгебраических преобразований, способы вычислений корней уравнений также можно отнести к математическим алгоритмам.

В наше время понятие алгоритма трактуется шире. Алгоритм — это последовательность команд управления каким-либо исполнителем. В качестве исполнителя выступает компьютер. Компьютер работает с величинами — различными информационными объектами: числами, символами, кодами и т. п. Поэтому алгоритмы, предназначенные для управления компьютером, принято называть алгоритмами работы с величинами.

Алгоритм – это понятное и точное предписание (указание) исполнителю совершить определенную последовательность действий для достижения указанной цели или решения поставленной задачи (приводящую от исходных данных к искомому результату).

2. ОСНОВНЫЕ ОСОБЕННОСТИ И СВОЙСТВА АЛГОРИТМОВ:

- Наличие ввода исходных данных.
- Наличие вывода результата выполнения алгоритма, поскольку цель выполнения алгоритма – получение результата, имеющего вполне определенное отношение к исходным данным.

Дискретность – разбивка алгоритма на элементарные команды, и выполнение очередной команды начинается после завершения предыдущей.

Формальность – свойство, позволяющее любому исполнителю, способному воспринимать и выполнять отдельные указания алгоритма, правильно выполнить весь алгоритм.

Определенность (точность) – однозначность предписанной последовательности действий, не допускающей ее двоякого толкования.

Понятность – свойство, предусматривающее то, что алгоритм должен состоять только из тех команд, которые исполнитель может выполнить.

Результативность (конечность) – исполнение алгоритма должно закончиться за конечное число шагов.

Корректность – алгоритм должен задавать правильное решение задачи.

Массовость – алгоритм разрабатывается для решения некоторого класса задач, различающихся исходными данными.

Эффективность – алгоритм должен выполняться за разумное конечное время. При этом выбирается наиболее простой и короткий способ решения задачи при соблюдении всех ограничений и требований к алгоритму.

Свойства дискретности, формальности, точности, понятности и конечности являются необходимыми (иначе это не алгоритм). Свойство массовости не является необходимым свойством алгоритма, оно скорее определяет его качество.

3. ФОРМЫ ЗАПИСИ АЛГОРИТМОВ

Алгоритмы можно записывать по-разному. Форма записи, состав и количество операций алгоритма зависит от того, кто будет исполнителем этого алгоритма.

Способы описания алгоритма:

- **Формульная запись** (производится с помощью простых математических, химических, физических формул. Например: $S=V*t$; $V=S/t$; $t=S/V$)
- **Табличная запись** (производится с помощью букв латинского алфавита и знака, который называется знаком присвоения :=)

ШАГ	ОПИСАНИЕ ДЕЙСТВИЙ
1.	$a:=5x$
2.	$b:=a+3$
3.	$c:=4x$
4.	$d:=c-7$
5.	$y:=b/d$

- **Развернутая словесная** (описание алгоритма производится на любом национальном языке, единственным условием является соблюдение свойств алгоритма. Используется обычно для описания алгоритмов, предназначенных исполнителю – человеку. Команды записываются на обычном языке и выполняются по порядку. В командах могут использоваться формулы, специальные обозначения, но каждая команда должна быть понятна исполнителю. Естественный порядок команд может быть нарушен, в этом случае команды можно нумеровать и указывать команду, к которой требуется перейти.)

ШАГ	ОПИСАНИЕ ДЕЙСТВИЙ
1.	число 7 умножить на переменную x
2.	из результата шага 1 вычесть число 4

3.	число 5 умножить на переменную x
4.	к результату шага 3 прибавить число 3
5.	результат шага 2 разделить на результат шага 4

- **На алгоритмическом языке** (записывается с помощью служебных слов, языка и формул (знаков присвоения).)

найти большее из трех чисел.

Алгоритм БИТ

1. **чтение** a, b, c
 2. **если** $a < b$ **к 4**
 3. $y := b$; **к 5**
 4. $y := a$
 5. **если** $y > c$ **к 7**
 6. $y := c$
 7. **запись** y
 8. **конец**
- **Графический** (в виде блок-схемы) ((или графическая запись). Алгоритмы представляются в виде блок-схем. Существуют специальные стандарты для построения блок-схем, где определяются графические изображения блоков. Команды алгоритмов записываются внутри блоков на обычном языке или с использованием математических формул. **Блоки соединяются по определенным правилам линиями связи, которые показывают порядок выполнения команд.**)
 - **На языке программирования** (Если алгоритм разработан для решения задачи на ЭВМ, то для того, чтобы он мог выполняться исполнителем – ЭВМ, его необходимо записать на языке, понятном этому исполнителю. Для этого разработано множество языков программирования для решения задач разных классов. Запись алгоритма на языке программирования называется программой.)

4. ВИДЫ АЛГОРИТМОВ:

Линейный алгоритм – это алгоритм, в котором все операции выполняются последовательно одна за другой.

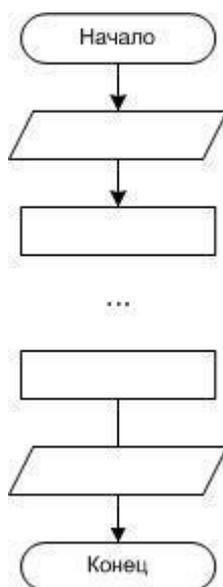


Рисунок 1 - Размещение блоков в линейном алгоритме

Рассмотрим несколько примеров линейных алгоритмов.

ПРИМЕР 1. Зная длины трех сторон треугольника, вычислить площадь и периметр треугольника. Пусть a , b , c – длины сторон треугольника. Необходимо найти S – площадь треугольника, P – периметр. Для нахождения площади можно воспользоваться формулой Герона. Входные данные: a , b , c . Выходные данные: S , P . Блок-схема алгоритма представлена на рисунке ниже.

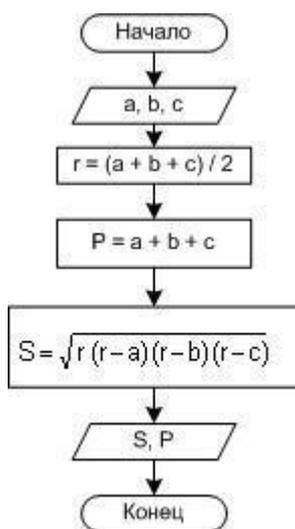


Рисунок 2 - Алгоритм примера 1

Внимание!!! В этих блоках знак «=» означает не математическое равенство, а операцию присваивания. Переменной, стоящей слева от оператора, присваивается значение, указанное справа. Причем это значение может быть уже определено или его необходимо вычислить с помощью выражения.

Например, операция $r = (a+b+c)/2$ – имеет смысл (переменной r присвоить значение $r=(a+b+c)/2$), а выражение $(a+b+c)/2=r$ – бессмыслица.

ПРИМЕР 2. Известны плотность и геометрические размеры цилиндрического слитка, полученного в металлургической лаборатории. Найти объем, массу и площадь основания слитка. Входные данные: R – радиус основания цилиндра, h – высота цилиндра, ρ – плотность материала слитка. Выходные данные: m – масса слитка, V – объем, S – площадь основания. Блок-схема представлена на рисунке ниже.

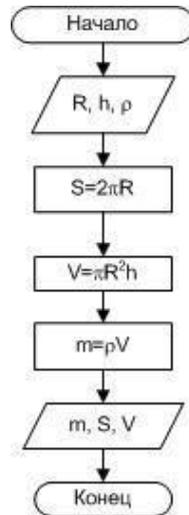


Рисунок 3 - Алгоритм примера 2

ПРИМЕР 3.3. Заданы длины двух катетов в прямоугольном треугольнике. Найти длину гипотенузы, площадь треугольника и величину его углов. Входные данные: a, b – длины катетов. Выходные данные: c – длина гипотенузы, S – площадь треугольника, α, β – углы. Блок-схема представлена на рисунке ниже.

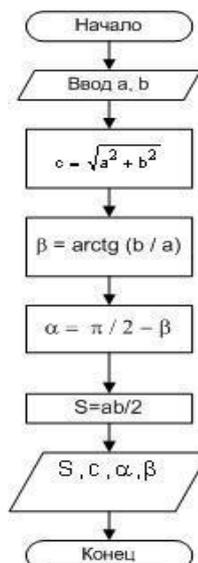


Рисунок 4 - Алгоритм примера 3

5. ПРОГРАММА MICROSOFT VISIO

Программа Microsoft Visio – векторный графический редактор, редактор диаграмм и блок-схем для Windows. В Microsoft Visio сложные для понимания таблицы и текст можно преобразовать в наглядные доступные схемы.

Приложение Microsoft Visio содержит современные фигуры и шаблоны для создания самых разнообразных схем в таких областях, как управление ИТ-средой, моделирование процессов, строительство и архитектурное проектирование, разработка пользовательского интерфейса, управление кадрами, проектами и т. д.

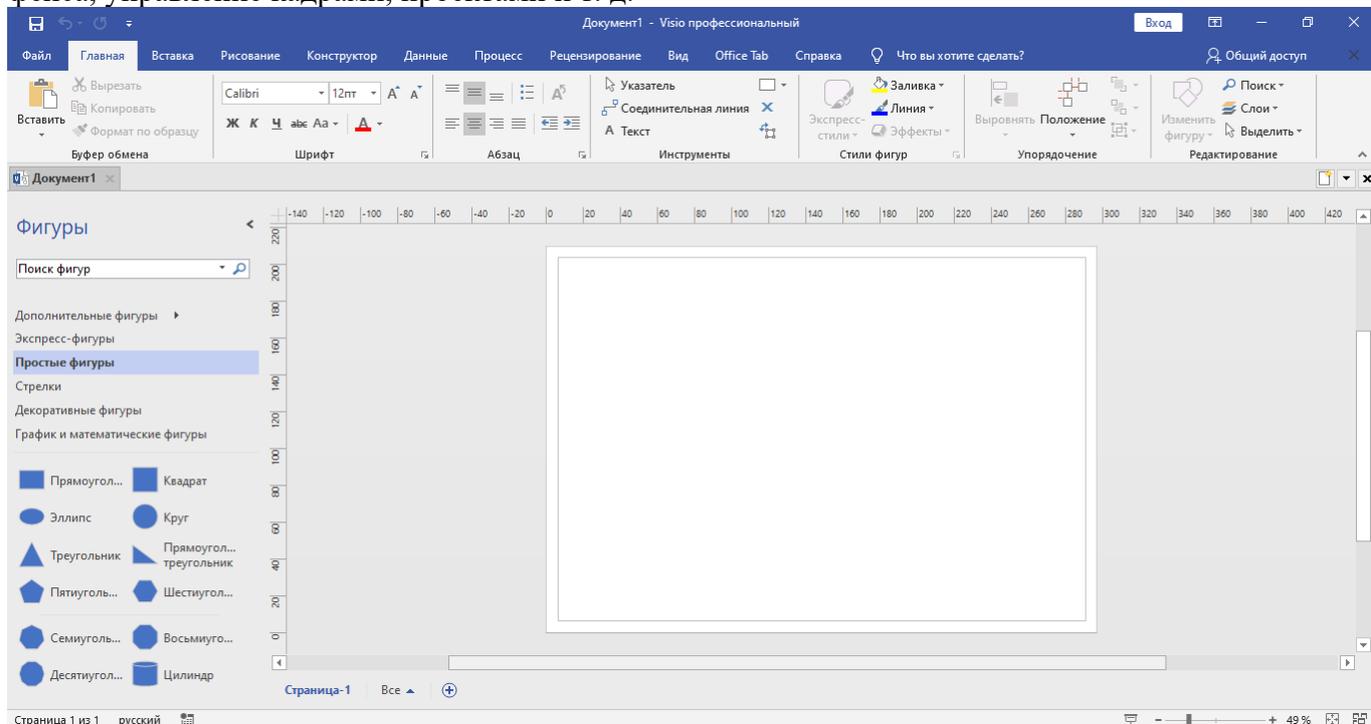
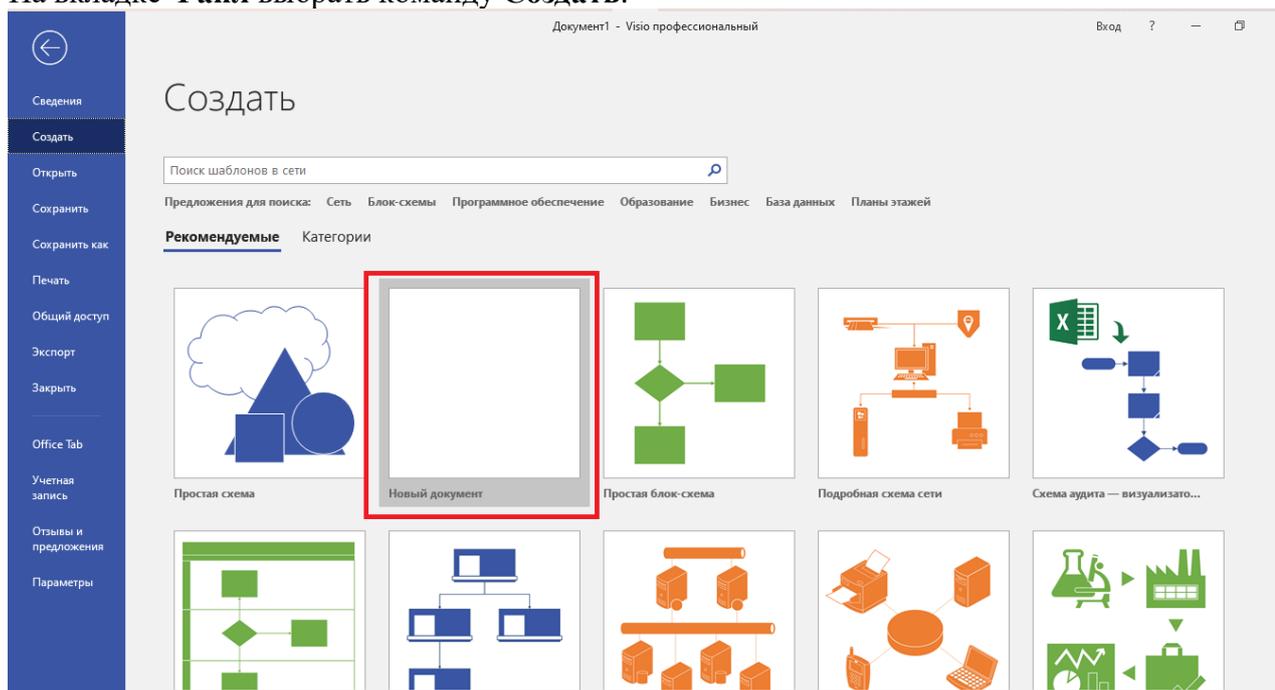


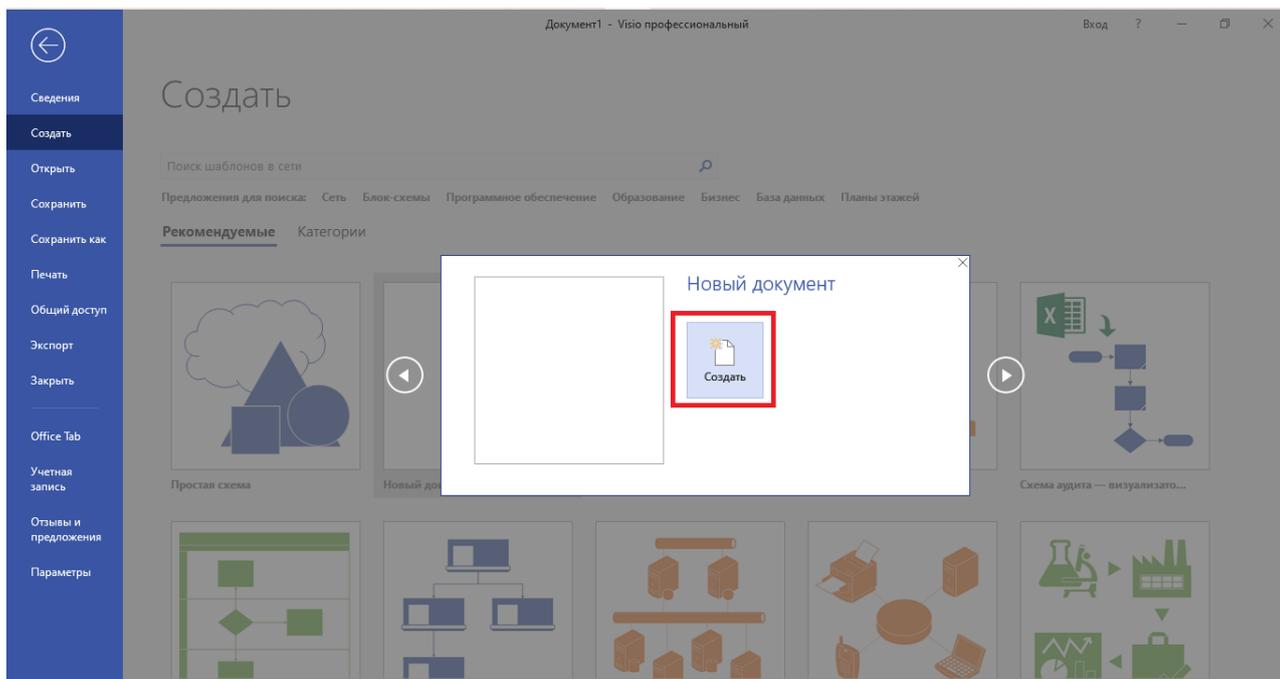
Рисунок 5 - Главное окно программы

6. НАЧАЛО РАБОТЫ:

1. На вкладке **Файл** выбрать команду **Создать**.

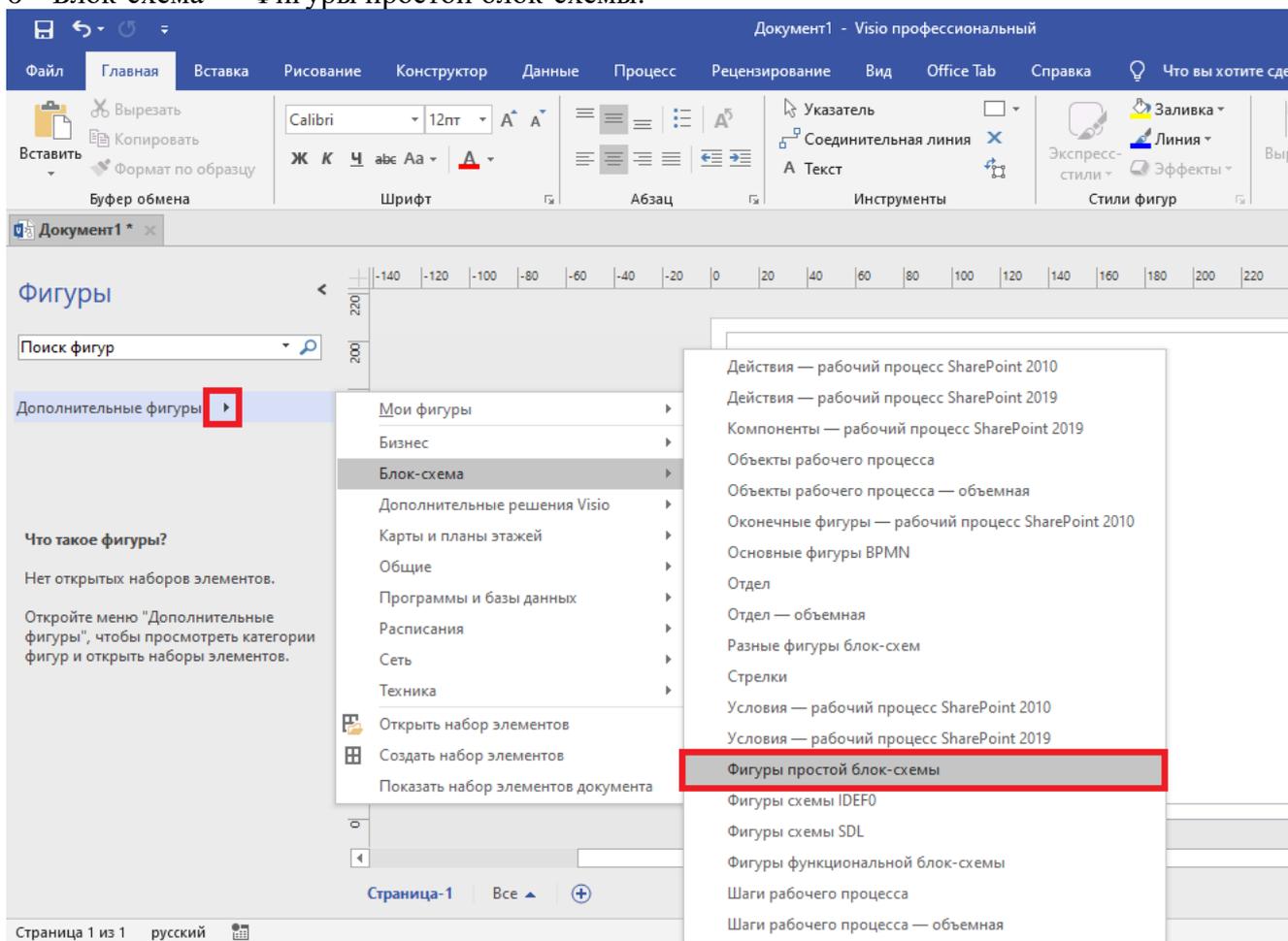


2. На вкладке **Файл** выбрать команду **Создать**.

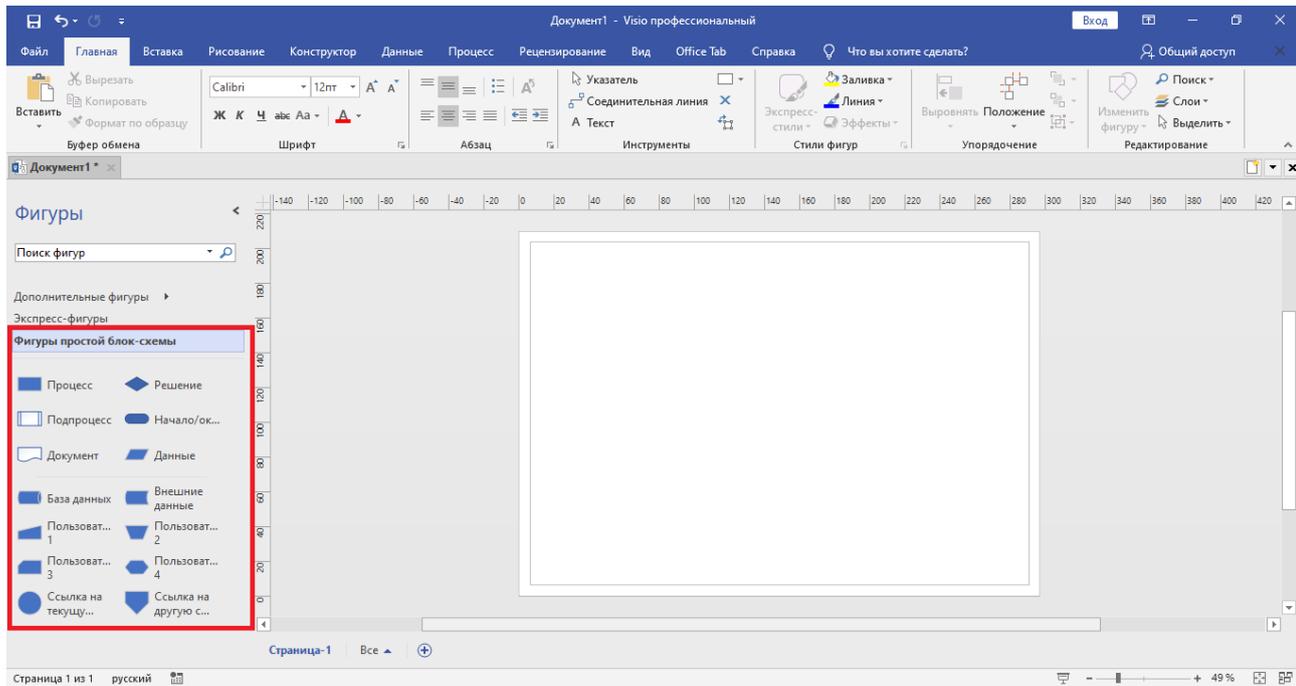


3. Отобразить в окне **Фигуры** набор элементов «**Фигуры простой блок-схемы**»:

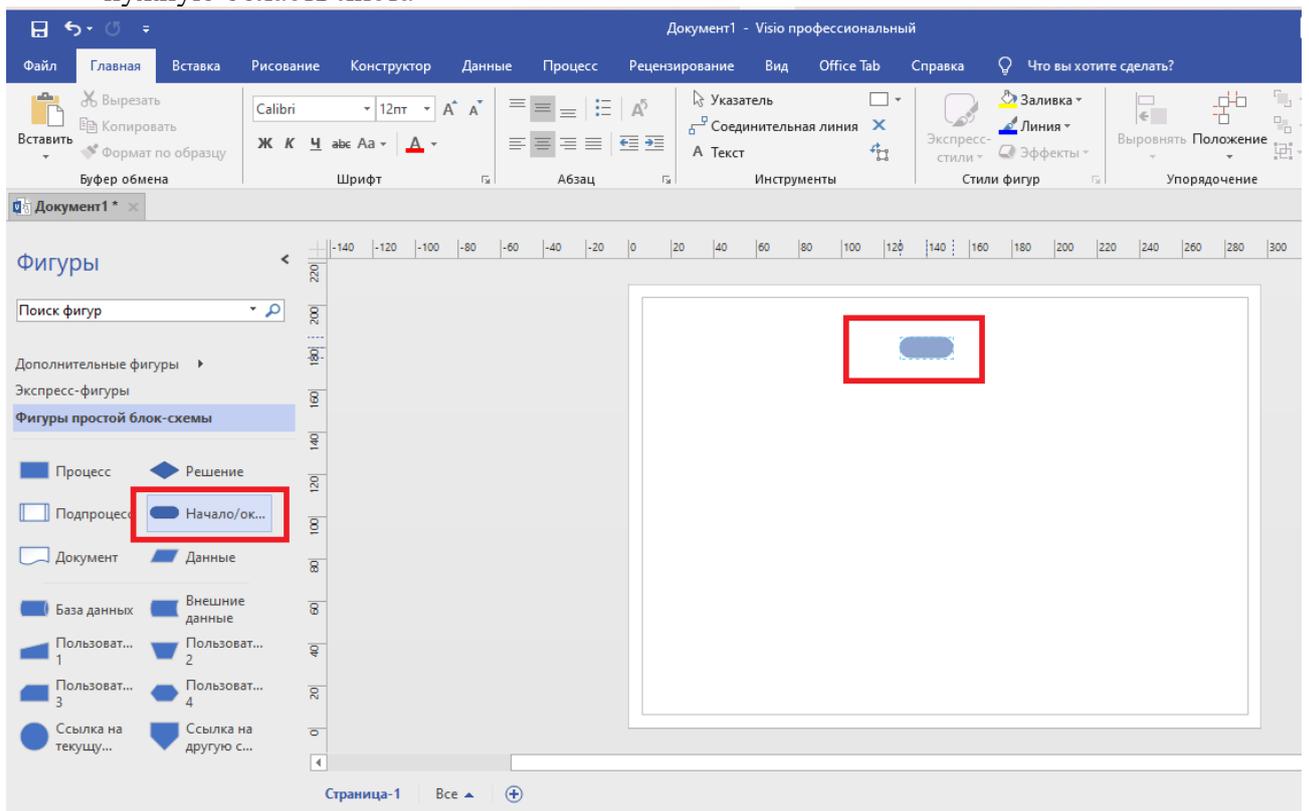
- В окне Фигуры щелкнуть **Дополнительные фигуры**.
- В открывшемся списке навести мышь на название категории нужного набора элементов, затем щелкнуть название набора элементов:
 - Блок-схема — Фигуры простой блок-схемы.



4. Доступен просмотр фигуры набора элементов **Фигуры простой блок-схемы**. Для этого в окне Фигуры щелкните название набора и просмотрите список фигур.



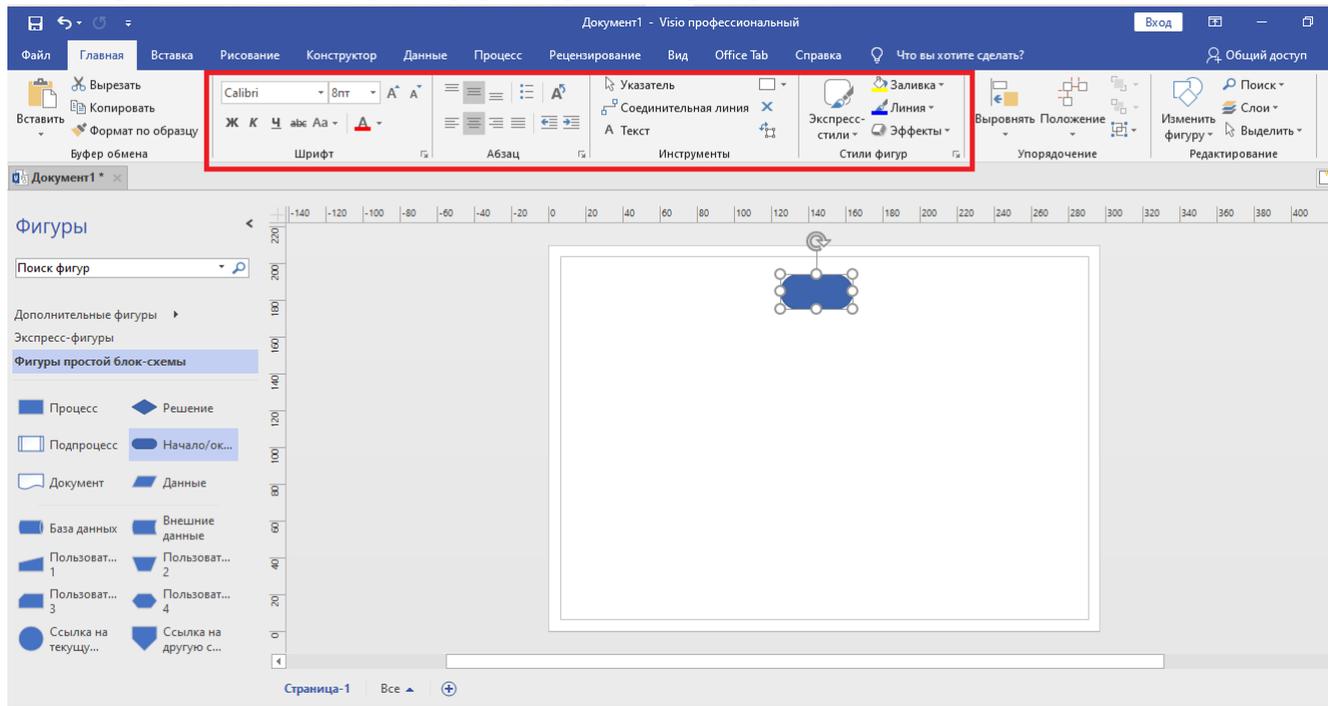
5. Чтобы добавить фигуру на лист, нужно мышкой перетащить необходимый элемент в нужную область листа



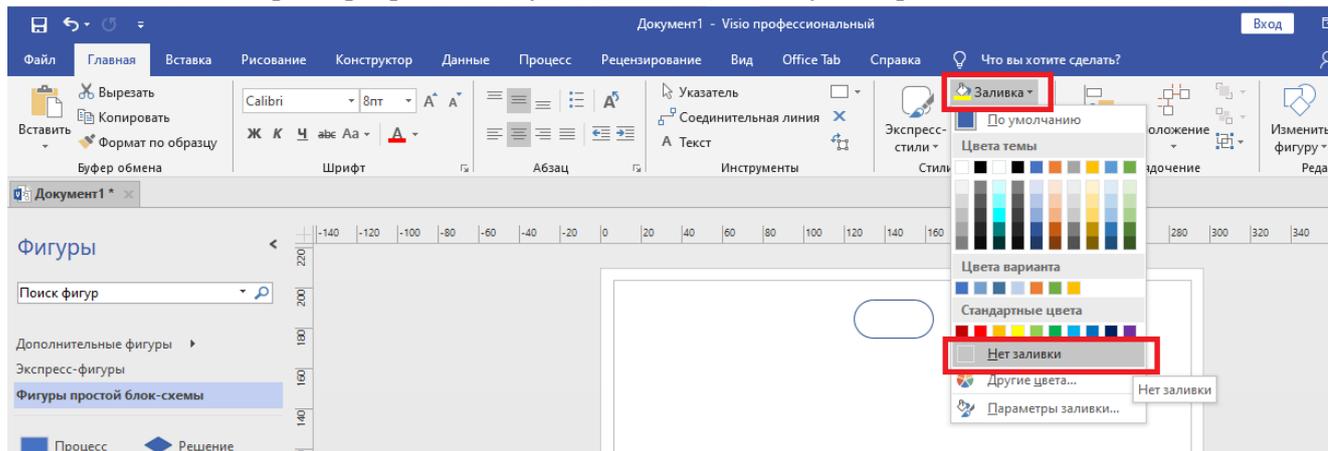
6. При необходимости можно изменить размер фигуры, потянув за углы



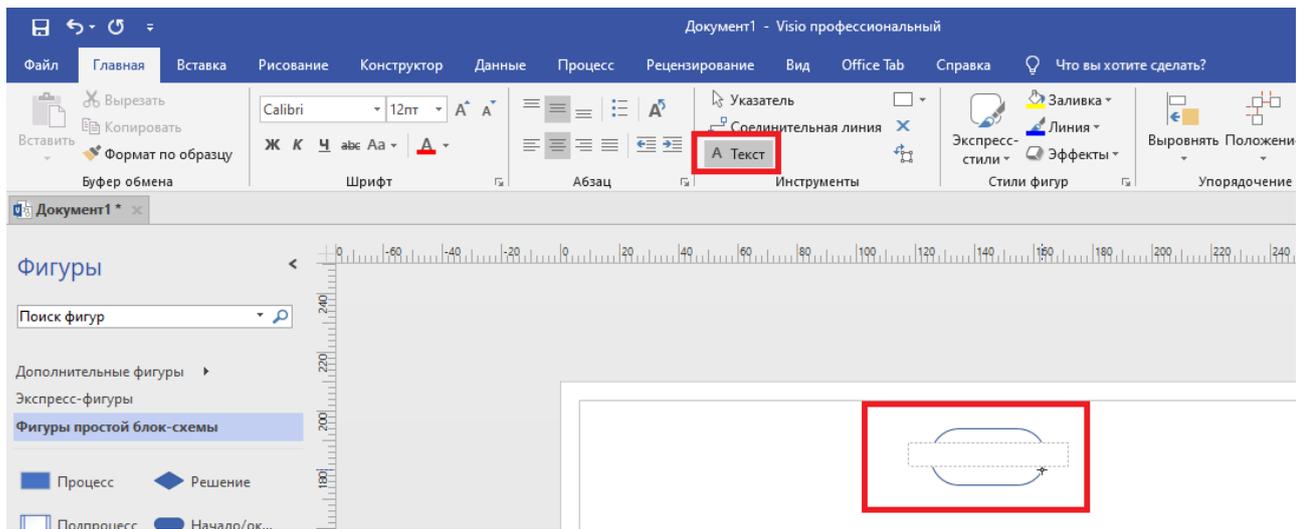
7. В верхней панели доступно форматирование фигуры.



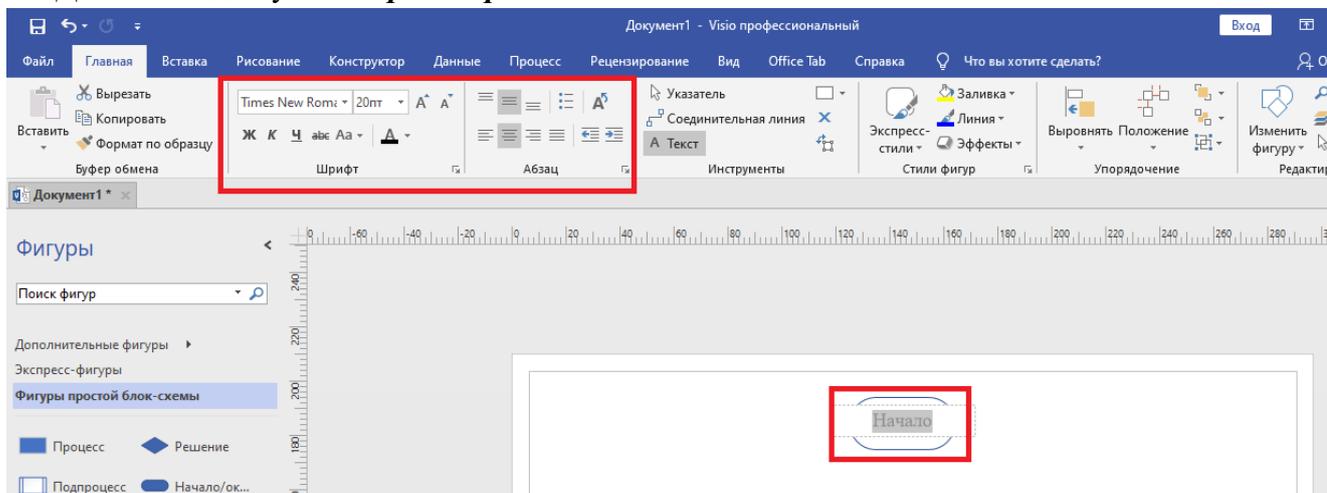
8. Чтобы сделать фон прозрачным, нужно зайти в **Заливку**, выбрать **Нет заливки**.



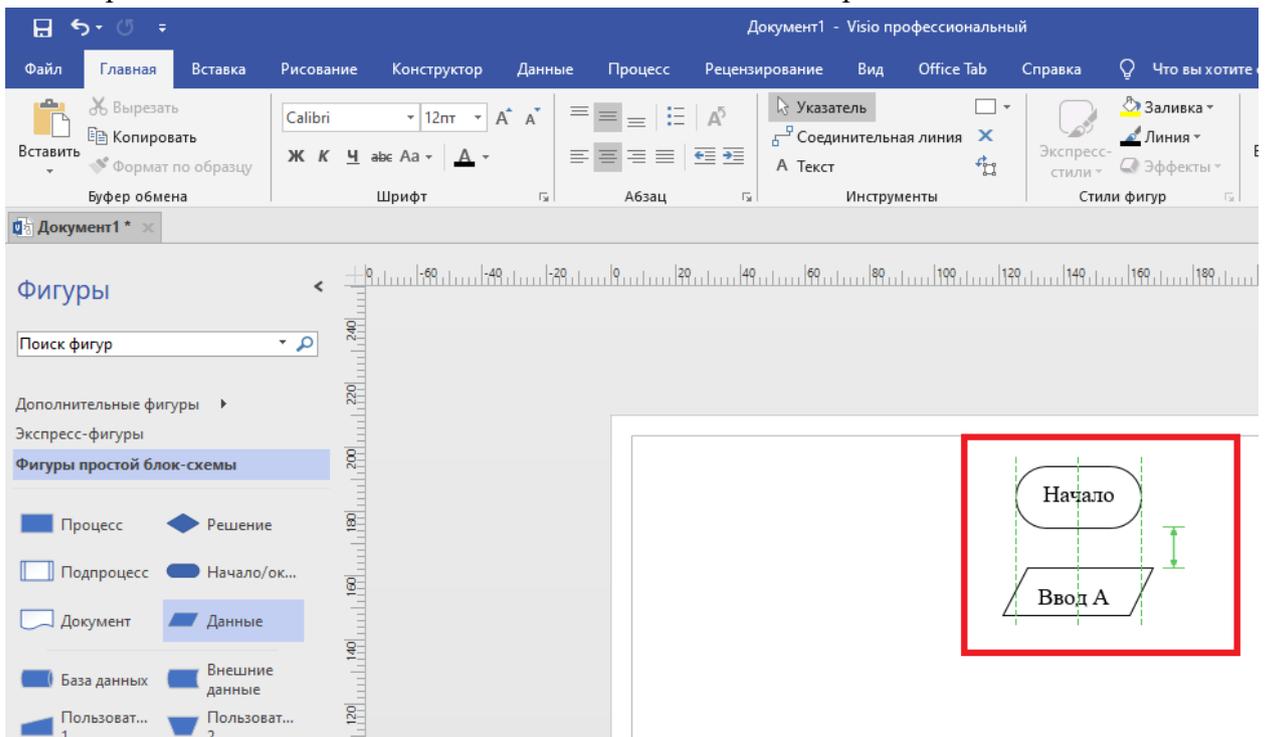
9. Чтобы добавить текст в блок, нужно выбрать инструмент **Текст** и нажать в нужное место на блок-схеме.



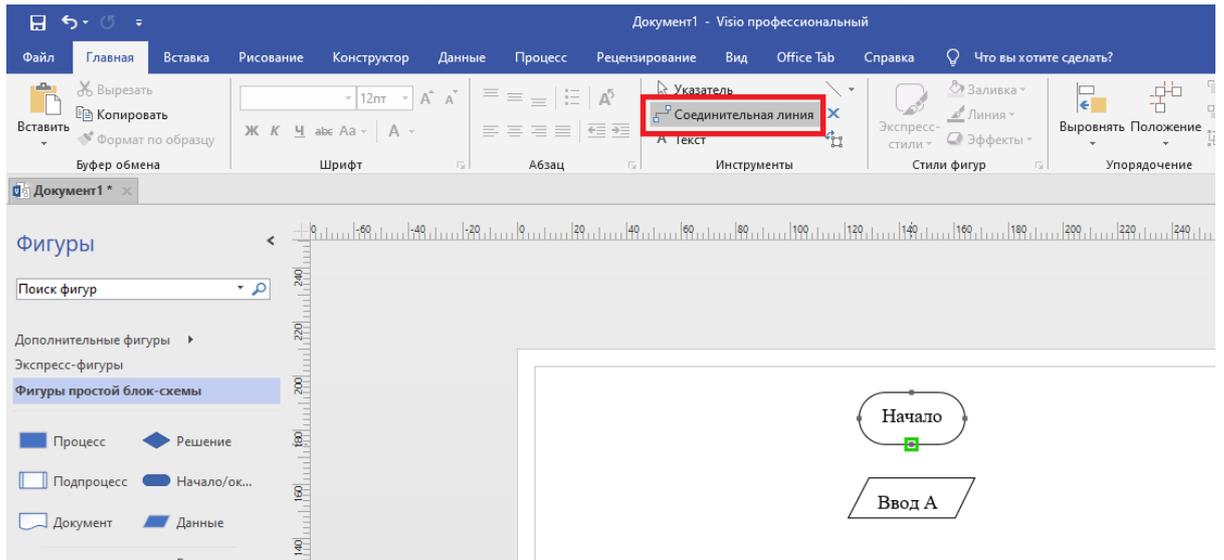
10. Для текста доступно **Форматирование**



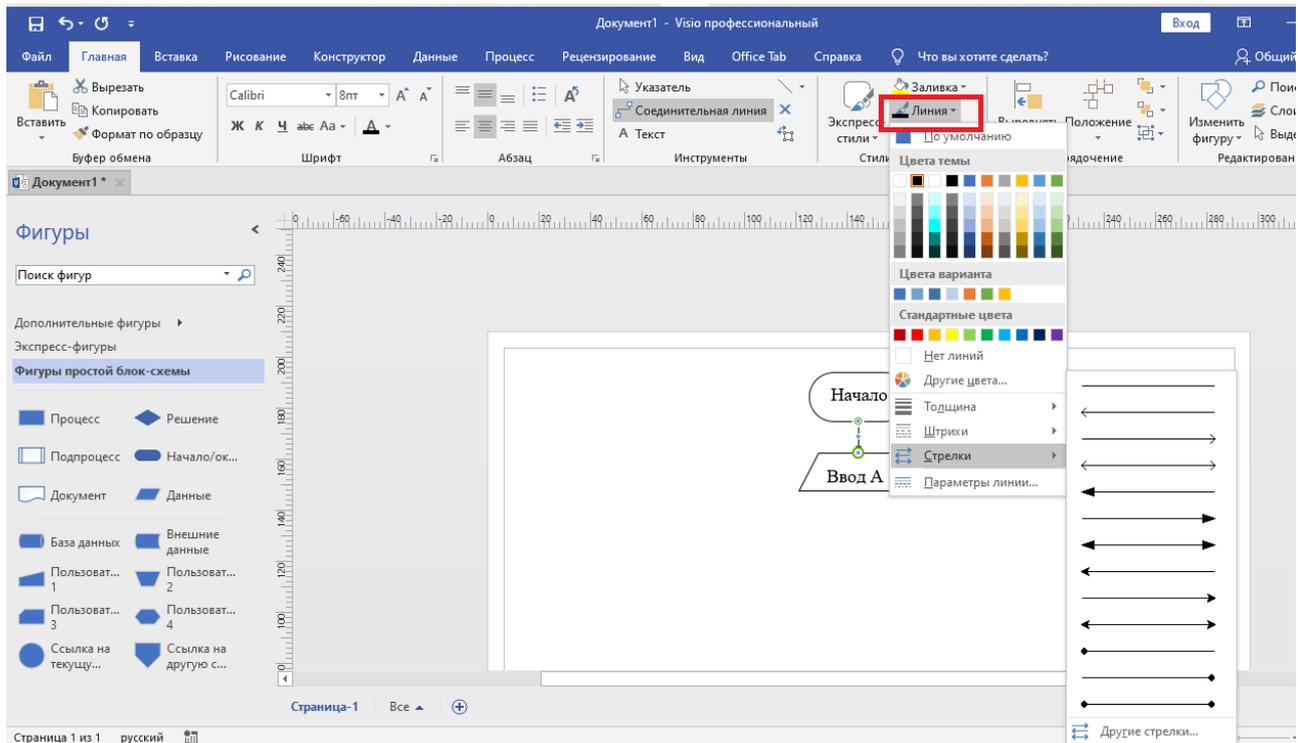
11. При добавлении нового блока использовать позиционирование с помощью мыши



12. Чтобы соединить блоки, нужно использовать *соединительную линию*



13. Чтобы провести линию, нужно навести курсор на исходный блок и с помощью левой клавиши мыши протянуть линию. Для линии доступно различное форматирование.



14. Для сохранения работы нажать *Файл – Сохранить как*.

4. Задания:

1 Задание: выполнить создание линейного алгоритма с помощью программы Microsoft Visio.

Ознакомиться с краткими теоретическими сведениями. Особенности работы с Microsoft Visio

- Создать алгоритм, рассчитывающий разницу между количеством заказанного и проданного товаров.

- b) Создать алгоритм, реализующий получение произведения 7 любых чисел.
 c) Вычислить значения по формулам при заданном значении всех переменных

1	$F(x) = x - \frac{x^3}{3} + \frac{x^5}{3}$	2	$F(x) = 3^x - 4x + (y - \sqrt{ x })$
---	--	---	--------------------------------------

- d) Вычислить расстояние между двумя точками с заданными координатами (x_1, y_1) и (x_2, y_2) .

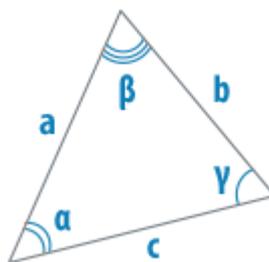
Справка:

Формулы вычисления расстояния между двумя точками:

- Формула вычисления расстояния между двумя точками $A(x_a, y_a)$ и $B(x_b, y_b)$ на плоскости:

$$AB = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

- e) Дан произвольный треугольник ABC , для которого определен набор характерных параметров: a, b, c – стороны треугольника. Найти, чему равны углы α, β, γ .



Справка:

$$\cos(\alpha) = \frac{a^2 + c^2 - b^2}{2ac}$$

$$\cos(\beta) = \frac{a^2 + b^2 - c^2}{2ab} \quad \cos(\gamma) = \frac{b^2 + c^2 - a^2}{2cb}$$

- f) Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов A и B .
 g) Дано число a . Используя только умножение, получить значения:

1	a^8 за три операции
2	a^{13} за пять операций
3	a^7 за четыре операции
4	a^{28} за шесть операций

Справка:

1	a^8 за три операции $b = a * a \quad (a^2)$ $c = b * b \quad (a^2 * a^2 = a^4)$
---	---

	$d = c * c \quad (a^4 * a^4 = a^8)$
--	-------------------------------------

5. Содержание отчета

Отчет должен содержать:

1. Название работы.
2. Цель работы.
3. Ход выполнения работы.
4. Результаты выполнения задания со скриншотами.
5. Вывод по работе.

6. Контрольные вопросы

1. Что такое алгоритм? Привести примеры.
2. Перечислить свойства алгоритма. Привести примеры.
3. Назовите формы записи алгоритмов. Привести примеры.
4. Что такое линейный алгоритм? Из каких функциональных блоков он состоит?
5. Что такое Microsoft Visio? Для чего применяется?
6. Создать простейший алгоритм по заданию преподавателя.

3.6 Методические указания по подготовке к устному опросу

Целью устного собеседования являются обобщение и закрепление изученного курса.

Студентам предлагаются для освещения сквозные концептуальные проблемы. При подготовке следует использовать лекционный материал и учебную литературу. Для более глубокого постижения курса и более основательной подготовки рекомендуется ознакомиться с указанной дополнительной литературой. Готовясь к семинару, студент должен, прежде всего, ознакомиться с общим планом семинарского занятия. Следует внимательно прочесть свой конспект лекции по изучаемой теме и рекомендуемую к теме семинара литературу. При этом важно научиться выделять в рассматриваемой проблеме самое главное и сосредотачивать на нем основное внимание при подготовке. С неизвестными терминами и понятиями следует ознакомиться в предлагаемом глоссарии, словаре или энциклопедии.

Ответ на каждый вопрос из плана семинарского занятия должен быть доказательным и аргументированным, студенту нужно уметь отстаивать свою точку зрения. Для этого следует использовать документы, монографическую, учебную и справочную литературу. Активно участвуя в обсуждении проблем на семинарах, студенты учатся последовательно мыслить, логически рассуждать, внимательно слушать своих товарищей, принимать участие в спорах и дискуссиях.

Для успешной подготовки к устному опросу, студент должен законспектировать рекомендуемую литературу, внимательно осмыслить фактический материал и сделать выводы. Студенту надлежит хорошо подготовиться, чтобы иметь возможность грамотно и полно ответить на заданные ему вопросы, суметь сделать выводы и показать значимость данной проблемы для изучаемого курса. Студенту необходимо также дать анализ той литературы, которой он воспользовался при подготовке к устному опросу на семинарском занятии.

При подготовке, студент должен правильно оценить вопрос, который он взял для выступления к семинарскому занятию. Но для того чтобы правильно и четко ответить на поставленный вопрос, необходимо правильно уметь пользоваться учебной и дополнительной литературой.

Перечень требований к любому выступлению студента примерно таков:

- связь выступления с предшествующей темой или вопросом.
- раскрытие сущности проблемы.
- методологическое значение для научной, профессиональной и практической деятельности.

сти.

Разумеется, студент не обязан строго придерживаться такого порядка изложения, но все аспекты вопроса должны быть освещены, что обеспечит выступлению необходимую полноту и завершенность. Приводимые участником семинара примеры и факты должны быть существенными, по возможности перекликаться с профилем обучения. Выступление студента должно соответствовать требованиям логики. Четкое вычленение излагаемой проблемы, ее точная формулировка, неукоснительная последовательность аргументации именно данной проблемы, без неоправданных отступлений от нее в процессе обоснования, безусловная доказательность, непротиворечивость и полнота аргументации, правильное и содержательное использование понятий и терминов

3.7 Типовое задание

Контрольная работа по теме «Списки, кортежи, массивы»

Задание 1. Выполнить контрольную работу согласно своему варианту.

I вариант	II вариант
1. Ответить на вопросы:	
<p>а) Объявите список <i>sp</i> из элементов: 4, -3, -18, 98, «Python». Можно ли из данного списка получить массив элементов? Ответ обоснуйте.</p> <p>б) Создать кортеж, в котором значение «-5.3» повторяется 8 раз. Определить сумму элементов кортежа. Обоснуйте ответ на вопрос: возможно ли изменить 3 элемент кортежа на 5.3?</p> <p>в) Продемонстрируйте сравнение списков: <code>num1 = [1, 2, 3, 4, 5]</code> и <code>num2 = [range(1,6)]</code>. Как проверить в данных списках наличие элемента 3?</p> <p>г) Дан список <code>spisok</code>, состоящий из элементов: 12, 17, 0, -67. Примените известные вам методы списков:</p> <ul style="list-style-type: none"> – из списка удалить число 17; – список принял вид: 12, 12, 0, -67 – список принял вид: 0, -67, 28 – список принял вид: -67. 	<p>а) Объявите кортеж <i>kor</i> из элементов: 56, 2.34, -18, «C++», (28, 13). Можно ли из данного кортежа получить список элементов? Ответ обоснуйте.</p> <p>б) Создать список, в котором значение «4/4» повторяется 7 раз. Определить среднее арифметическое элементов кортежа. Обоснуйте ответ на вопрос: возможно ли изменить 3 элемент списка на 5.3?</p> <p>в) Продемонстрируйте сравнение кортежей: <code>num1 = (0, 1, 2, 3, 4, 5)</code> и <code>num2 = (range(0,6))</code>. Как проверить в данных кортежах наличие элемента 6?</p> <p>г) Дан кортеж <code>korteg</code>, состоящий из элементов: 34, 1, -3, 56, 12. Примените известные вам методы кортежей:</p> <ul style="list-style-type: none"> – сделать срез из кортежа, чтобы получить число 34; – конвертировать кортеж в список и дописать в конец списка элемент 23. – сделать срез из кортежа, чтобы получить числа 1, -3; – сделать срез всех элементов из кортежа;
2. Написать программу с использованием массива:	
<p>а) Дан массив [-2, 11, 4, 23, 4]. Найти количество вхождений элемента 4.</p> <p>б) Дан массив [17, -12, 8, 4, 5, 7]. Найти сумму элементов массива, больших 5, но меньших 11.</p>	<p>а) Дан массив [7, -2, 0, 0, -1]. Найти сумму элементов массива. Вывести последний элемент массива.</p> <p>б) Дан массив [12, 21, 13, 4, -5, 3, 11]. Найти сумму элементов массива, больших 10, но меньших 21.</p>

Экспертная оценка выполнения работы

Оценка «5»

- выполнил работы в полном объеме с соблюдением необходимой последовательности действий;
- в ответе правильно и аккуратно выполняет все записи;
- правильно выполняет анализ ошибок.

Оценка «4» ставится, если выполнены требования к оценке 5, но допущены 2-3 недочета, не более одной ошибки и одного недочета.

Оценка «3» ставится, если

- работа выполнена не полностью, но объем выполненной части таков, что позволяет получить правильные результаты и выводы
- в ходе проведения работы были допущены ошибки.

Оценка «2» ставится, если студент совсем не выполнил работу.

4 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ Теоретические вопросы к экзамену:

1. Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритма. Схема решения задач на ЭВМ. Формы записи алгоритмов. Общие принципы построения алгоритмов. Методы разработки алгоритмов.
2. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.
3. Составление блок-схем программ. Правила оформления блок-схем. Примеры реализации.
4. Языки программирования. Эволюция языков программирования. Классификация языков программирования. Понятие системы программирования. Исходный, объектный и загрузочный модули. Интегрированная среда программирования.
5. Методы программирования: структурный, модульный, объектно-ориентированный. Достоинства и недостатки методов программирования. Общие принципы разработки программного обеспечения.
6. Типы приложений. Консольные приложения. Оконные Windows приложения. Web-приложения. Библиотеки. Web-сервисы.
7. Язык программирования Python: лексика языка, переменные и константы, типы данных.
8. Язык программирования Python: функции ввода-вывода, операторы, операции и встроенные функции языка.
9. Логические выражения и операторы. Ветвление. Условный оператор.
10. Составной оператор. Множественное ветвление.
11. Ошибки и исключения. Циклические конструкции. Циклы с предусловием и постусловием.
12. Понятие списков и кортежей. Стандартные операции со списками и кортежами. Понятие среза.
13. Понятие массива. Применение отдельных методов массивов (Array) в Python.
14. Строки. Основные операции и функции.
15. Множества. Основные операции и функции.
16. Словарь. Основные операции и функции.
17. Файлы. Работа с файлами. Файлы последовательного доступа. Файлы прямого доступа.
18. Понятие подпрограммы. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров.
19. Понятие функции в программировании: синтаксис, передача аргументов. Формальные и фактические параметры.
20. Рекурсия: способы организации и описание. Вызов рекурсии. Программирование рекурсивных алгоритмов. Стандартные функции.

21. Программирование модулей. Модуль: синтаксис, заголовок, разделы. Библиотеки подпрограмм: понятие и виды. Схемы вызова библиотек. Статическое и динамическое связывание. Использование библиотек подпрограмм.
22. История развития ООП. Базовые понятия: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Перегрузка операторов.
23. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. Классы объектов. Компоненты и их свойства. Документирование кода. Декораторы.
24. Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Окно кода проекта. Форма и размещение на ней управляющих элементов.
25. Панель компонентов и их свойства. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Компиляция и выполнение проекта. Этапы разработки объектно-ориентированного приложения. Создание интерфейса пользователя.
26. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления.
27. Свойства компонентов (элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства.
28. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определенные пользователем: синтаксис, передача аргументов. Вызов событий.
29. Принципы разработки Меню и тулбара. Диалоговые окна.
30. Обработка событий окна, клавиатуры, мыши.
31. Функциональная схема работы приложения. Создание простейшего приложения. Средства управления параметрами проекта и среды разработки.
32. Работа с датой и временем. Календарь. Списки и таблицы. Модели и представления.

Примеры заданий для выполнения практической части.

1. Составление блок-схем линейных алгоритмов.
2. Составление блок-схем разветвляющихся алгоритмов.
3. Составление блок-схем циклических алгоритмов.
4. Составление блок-схем программ.
5. Применение основных операторов языка Python.
6. Применение функций ввода-вывода, операции и встроенные функции языка Python.
7. Составление программ разветвляющейся структуры. Множественное ветвление.
8. Составление программ циклической структуры. Ошибки и исключения.
9. Обработка списков и кортежей.
10. Обработка массивов.
11. Обработка строковых данных.
12. Обработка множеств.
13. Применение и обработка словарей.
14. Файлы. Работа с файлами.
15. Применение функций в программировании.
16. Организация и использование рекурсий.
17. Программирование модуля.

18. Создание и использование библиотеки подпрограмм.
19. Применение наследования и полиморфизма в ООП.
20. Перегрузка операторов.
21. Применение декораторов.
22. Разработка простейших программ графическим интерфейсом при помощи PyQt5.
23. Простейшая обработка сигналов и слотов.
24. Разработка Меню и тулбаров при помощи PyQt5.
25. Разработка диалоговых окон в PyQt5.
26. Управление макетом в PyQt5.
27. Обработка событий окна в PyQt5.
28. Обработка событий клавиатуры в PyQt5.
29. Обработка событий мыши в PyQt5.
30. Размещение нескольких компонентов в окне.
31. Работа с датой и временем. Календарь.
32. Списки и таблицы
33. Модели. Представления.

5 ПЕРЕЧЕНЬ ПЕЧАТНЫХ ИЗДАНИЙ, ЭЛЕКТРОННЫХ ИЗДАНИЙ (ЭЛЕКТРОННЫХ РЕСУРСОВ), ДОПОЛНИТЕЛЬНЫХ ИСТОЧНИКОВ

Основные источники:

1. Семакин И.Г., Шестаков А.П., Основы алгоритмизации и программирования. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / И.Г. Семакин, А.П. Шестаков. – 5 изд., испр. и доп. – М.: Академия, 2017. – 144 с.
2. Семакин И.Г., Шестаков А.П., Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И.Г. Семакин, А.П. Шестаков. – 4 изд., испр. и доп. – М.: Академия, 2017. – 304 с.
3. Трофимов, В.В. Основы алгоритмизации и программирования: учебник для СПО / В.В. Трофимов, Т.А. Павловская; под ред. В.В. Трофимова. - М.: Издательство Юрайт, 2018. – 137 с.

Дополнительные источники:

1. Голицына, О.Л., Попов И.И. Основы алгоритмизации и программирования: учеб. пособие / О.Л. Голицына, И.И. Попов. – 4-е изд., испр. и доп. - М.: Форум: ИНФРА-М, 2015. – 432 с.
2. Юркин, А. Г. Задачник по программированию. – СПб.: Питер, 2002. – 192 с.
3. Потопахин, В.В. Искусство алгоритмизации. – М.: ДМК Пресс, 2011. – 320 с.
4. Голицына, О.Л., Попов И.И. Основы алгоритмизации и программирования: учеб. пособие / О.Л. Голицына, И.И. Попов. – М.: Форум: ИНФРА-М, 2002. – 432 с.

