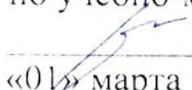


Автономная некоммерческая организация профессионального образования
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»
(АНО ПО «ПГТК»)

УТВЕРЖДАЮ

Заместитель директора
по учебно-методической работе
 О.В. Бушуева
«01» марта 2019 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ
МДК 01.01. Системное программирование

для специальности

09.02.03 Программирование в компьютерных системах
(код и наименование специальности)

Квалификация выпускника
Техник-программист
(базовая подготовка)

Форма обучения
Очная

Пермь, 2019 г

Фонд оценочных средств «МДК 01.01. Системное программирование» составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах (утвержден приказом Министерства образования и науки Российской Федерации от 28.07.2014 г., № 804).

Предназначен для студентов и преподавателей АНО ПО «ПГТК».

Автор – составитель: Тимохова Н.А., старший преподаватель.

Фонд оценочных средств учебной дисциплины рассмотрен и одобрен на заседании кафедры математических и естественно-научных дисциплин, протокол, № 06 от «21» января 2019 г.

Рекомендован к утверждению педагогическим советом АНО ПО «ПГТК» (протокол от «05» февраля 2019г. №3)

Оглавление

| | |
|---|---|
| 1.ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ. | 4 |
| 2.ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ. | 5 |

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1. Общие положения.

Фонды оценочных средств (ФОС) предназначены для контроля и оценки результатов освоения МДК 01.01. «Системное программирование»
 ФОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме экзамена.

| № № п/п | Контролируемые (темы) дисциплины разделы | Код контролируемой компетенции | Оценочное средство | |
|---------------|---|--------------------------------------|--|------------|
| | | | наименование | кол- во |
| 1. | Тема 1. Системное программное обеспечение (СПО) | ОК 1-9 ПК 1.1-1.6 | Контрольные вопросы Тестовые задания Практические работы Темы для рефератов | 11 |
| 2. | Тема 2. Классификация системных программ | | | 19 |
| 3. | Тема 3. Интерфейс операционной системы. | | | 2 |
| 4. | Тема 4. Средства разработки. | | | 3 |
| 5. | Тема 5. Особенности выполнения программ. | | | |
| 6. | Тема 6. Межпроцессные взаимодействия (IPC). | | | |
| 7. | Тема 7. Ввод-вывод. | | | |
| 8. | Тема 8. Программные уровни ввода-вывода | | | |
| 9. | Тема 9. Подсистема ввода-вывода в MS Windows XP. | | | |

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Раздел 1. Разработка спецификаций отдельных компонент

Задание 1. Перечень контрольных вопросов по разделу

1. Что такое программа?
2. Что такое программный модуль?
3. Что включает в себя компонент программного модуля?
4. Что такое программирование?
5. Расскажите о технологии программирования.
6. Какие этапы включает в себя программирование?
7. Расскажите о связи программирования с другими областями наук
8. Расскажите о концепции разработки программного модуля.
9. Что такое жизненный цикл программы?
10. Какие этапы включает в себя жизненный цикл программы?
11. Расскажите о требованиях к качеству программного модуля.

Шкала оценки контрольных вопросов:

Оценка 5 **«отлично»** выставляется студенту, если он правильно ответил на поставленный вопрос.

Оценка 4 **«хорошо»** выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка 3 **«удовлетворительно»** выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки.

Оценка 2 **«неудовлетворительно»** выставляется студенту, который не знает значительной части заданного вопроса.

Задание 2. Тестовые задания к разделу

1. Delphi – это...
 - 1) Язык программирования низкого уровня
 - 2) Система объектно-ориентированного визуального программирования**
 - 3) Программа обработки видео
 - 4) Программа для перевода кода программы на язык машинных кодов.
2. RAD – это...
 - 1) Программа записи видео
 - 2) Средство модульного программирования
 - 3) Среда быстрой разработки приложений**
3. Язык программирования Delphi
 - 1) Object Pascal**
 - 2) Java
 - 3) C#
4. IDE – это...
 - 1) История развития Delphi
 - 2) Приложение, разрабатываемое программистом
 - 3) Интегрированная среда разработки**
5. IDE не включает в себя: *(несколько вариантов ответа)*
 - 1) Редактор кода
 - 2) Высокопроизводительный компилятор в машинный код
 - 3) Объектно-ориентированную модель компонент
 - 4) Эффекты и переходы**
 - 5) Визуальное построение приложений
 - 6) Сопровождение ПП**
 - 7) Средство для построения баз данных
6. Объектно-ориентированная программа - ...
 - 1) Программирование, основанное на объектах
 - 2) Совокупность объектов и способов их взаимодействия**
 - 3) Структура среды программирования
7. Установить соответствие

| | |
|----------------------------|---|
| 1) Объект | а) Атрибуты (основные характеристики), которые описывают особенности объекта (цвет, ширина, положение и т.д.) |
| 2) Событие | б) Совокупность данных (компонентов) и методов работы с ними |
| 3) Свойство | в) Отклик на внешнее воздействие |
| 1 – б, 2 – в, 3 – а | |

8. Дополнить предложение. Основным окном разрабатываемого приложения является ...
 - 1) Код
 - 2) Форма**
 - 3) Библиотека

- 4) Объект
9. Дополнить предложение. Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...
- 1) Компилятора
- 2) C++
- 3) **Windows**
10. Названия процедур на определенное событие:
- 1) Компоненты
- 2) **Обработчики событий**
- 3) Редактор кода
11. Назначение главного меню:
- 1) **Осуществление функций управления при разработке программ**
- 2) Сопровождение программных продуктов
- 3) Автоматизирует процесс тестирования программ
12. Назначение панели инструментов:
- 1) Обработка событий
- 2) **Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто употребляемым командам Главного меню**
- 3) Построение баз данных
13. Назначение палитры компонентов:
- 1) **содержит пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение**
- 2) Подделка подписей
- 3) Управление Paint
14. Дополнить предложение. Главное окно управляет окнами ...
(несколько вариантов ответа)
- 1) **Инспектор объектов**
- 2) Язык ассемблера
- 3) **Редактора кода**
- 4) **Проектировщика форм**
- 5) Трансляции программы
15. Заготовка разрабатываемого приложения - ...
- 1) Компилятор
- 2) Свойство
- 3) **Окно проектировщика форм**
- 4) Главное окно
16. Действия, которые нельзя выполнить с помощью проектировщика форм
- 1) Добавить компоненты в форму
- 2) **Сменить язык программирования**
- 3) Модифицировать форму и её компоненты
- 4) Связать обработчики событий компонента с программой на Object Pascal, содержащейся в редакторе кода
- 5) **Документирование программы**

17. Дополнить предложение. Инспектор объектов позволяет ...
(несколько вариантов ответа)

- 1) **Устанавливать свойства объектов**
- 2) Изменять структуру программного кода
- 3) Изменять наследование классов объектов
- 4) **Назначать методы обработки событий**

18. Дополнить предложение. Инспектор объектов состоит из ...
(несколько вариантов ответа)

- 1) Unit – программного модуля
- 2) **Properties – списка свойств,**
- 3) **Events – списка событий.**
- 4) Begin...end.

19. Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:

- 1) Окно компилятора
- 2) **Окно редактора кода**
- 3) Окно проектировщика
- 4) Главное окно

Шкала оценки тестовых вопросов:

«2» – от 0 до 40%

«3» – от 41% до 60%

«4» – от 61% до 80%

«5» – от 81% до 100%

Задание 3. Лабораторные работы

Перечень тем лабораторных работ:

Практическая работа №1 «Разработка спецификаций качества отдельных компонент»

Практическая работа №2 «Разработка функциональных спецификаций отдельных компонент»

Шкала оценки лабораторных работ:

оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

Задание 4. Рефераты

Перечень тем рефератов:

1. Требования к качеству программного модуля
2. Спецификация качества программного модуля
3. Функциональная спецификация программного модуля

Шкала оценки рефератов:

Критерии и показатели, используемые при оценивании учебного реферата

| Критерии | Показатели |
|---|--|
| 1. Новизна реферированного текста Макс. - 20 баллов | <ul style="list-style-type: none">- актуальность проблемы и темы;- новизна и самостоятельность в постановке проблемы, в формулировании нового аспекта выбранной для анализа проблемы;- наличие авторской позиции, самостоятельность суждений. |
| 2. Степень раскрытия сущности проблемы Макс. - 30 баллов | <ul style="list-style-type: none">- соответствие плана теме реферата;- соответствие содержания теме и плану реферата;- полнота и глубина раскрытия основных понятий проблемы;- обоснованность способов и методов работы с материалом;- умение работать с литературой, систематизировать и структурировать материал;- умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы. |
| 3. Обоснованность выбора источников Макс. - 20 баллов | <ul style="list-style-type: none">- круг, полнота использования литературных источников по проблеме;- привлечение новейших работ по проблеме (журнальные публикации, материалы сборников научных трудов и т.д.). |
| 4. Соблюдение требований к оформлению Макс. - 15 баллов | <ul style="list-style-type: none">- правильное оформление ссылок на используемую литературу;- грамотность и культура изложения;- владение терминологией и понятийным аппаратом проблемы; |

| | |
|--|--|
| | <ul style="list-style-type: none"> - соблюдение требований к объему реферата; - культура оформления: выделение абзацев. |
| 5. Грамотность Макс. - 15 баллов | <ul style="list-style-type: none"> - отсутствие орфографических и синтаксических ошибок, стилистических погрешностей; - отсутствие опечаток, сокращений слов, кроме общепринятых; - литературный стиль. |

Оценивание реферата

Реферат оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

- 86 – 100 баллов – «отлично»;
- 70 – 75 баллов – «хорошо»;
- 51 – 69 баллов – «удовлетворительно»;
- мене 51 балла – «неудовлетворительно».

Баллы учитываются в процессе текущей оценки знаний программного материала.

Задание 1. Перечень контрольных вопросов

1. Отличительные особенности языка C++
2. Что такое инициализация объекта данных?
3. Что такое запись выражений в языке программирования C++?
4. Что такое условный оператор?
5. Что такое оператор цикла?
6. Что такое массив?
7. Что такое вектор?
8. Что такое указатели?
9. Для чего нужна запись файлов?
10. Для чего нужно чтение файлов?
11. Что такое процедурное программирование?
12. Что такое функции?
13. С помощью чего можно совершить вызов функции?
14. Как можно объявить функцию?
15. Что такое шаблонные функции?
16. Где используются шаблонные функции?
17. Цели использования компьютеров при решении прикладных задач.
18. Что такое системное программирование?
19. Что такое прикладное программирование?
20. Особенности системного программирования
21. Особенности прикладного программирования

22. Чем программист руководствуется при выборе языка программирования?
23. Цели технологии разработки прикладного программного обеспечения.
24. Задачи технологии разработки прикладного программного обеспечения.
25. Основные принципы технологии разработки прикладного программного обеспечения.
26. Инструменты технологии разработки прикладного программного обеспечения.
27. Что такое алгоритмическая декомпозиция?
28. Что такое объектно-ориентированная декомпозиция?
29. Что такое абстрагирование?
30. Что такое инкапсуляция?
31. Что такое наследование?
32. Что такое полиморфизм?
33. Что такое модульность?
34. Что такое сохраняемость?
35. Что такое параллелизм?
36. Что такое объект?
37. Какие существуют типы объектов?
38. Что такое атрибут?
39. Какие существуют типы атрибутов?
40. Что такое экземпляры?

Шкала оценки контрольных вопросов:

Оценка 5 «**отлично**» выставляется студенту, если он правильно ответил на поставленный вопрос.

Оценка 4 «**хорошо**» выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка 3 «**удовлетворительно**» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки.

Оценка 2 «**неудовлетворительно**» выставляется студенту, который не знает значительной части заданного вопроса.

Задание 2. Самостоятельная работа

Самостоятельная работа №1

ООП – новая концепция языка программирования

Цель работы

- рассмотреть практическое применение компонента TEdit (поле ввода данных);
- на примере рассмотреть использование оператора условия.

Задание 1. Разработка проекта «Кроссворд»

На основе примера кроссворда «Культура первобытности» составьте кроссворд на тему предложенную преподавателем и переложите его на код системы программирования DELPHI.

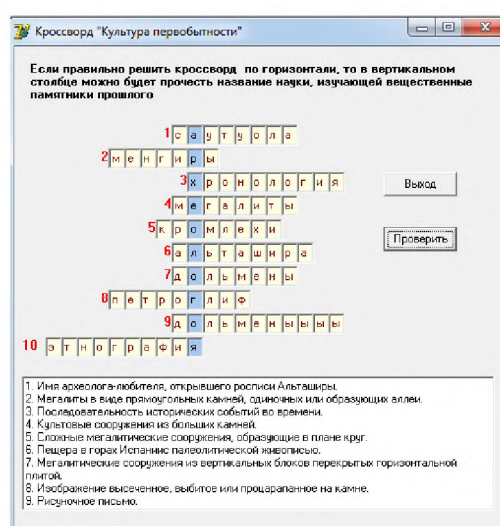
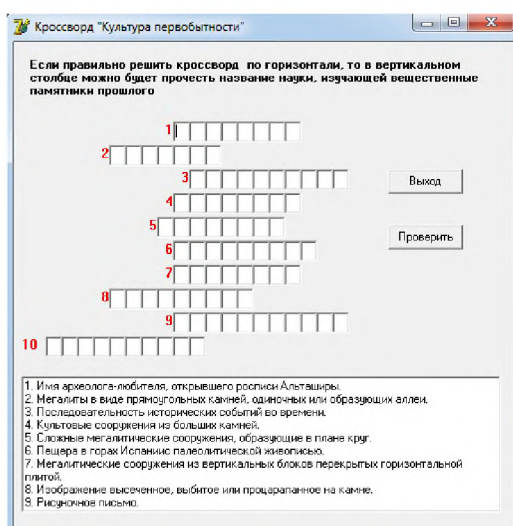
Пример «Кроссворд: Культура первобытности»

Пользователь вводит слова в клетки кроссворда.

При нажатии кнопки «проверить» программа стирает неверный ответ.

В случае правильного ответа надпись остается, поля кроссворда закрашиваются в кремовый цвет, посередине появляется ключевое слово «археология» голубого цвета.

Ответы: 1. Саутуола; 2. менгиры; 3. хронология; 4. мегалиты; 5. кромлехи; 6. Атьташира; 7. дольмены; 8. петроглиф; 9. пиктограмма; 10. этнография.



Самостоятельная работа №2

Проектирование многооконных приложений

Цель работы: рассмотреть этапы создания проекта приложения, включающего несколько окон.

Задание 1. Разработка проекта «Электронный учебник»

На основе примера электронного учебника «Основы алгоритмизации» разработайте форму электронного учебника на тему предложенную преподавателем и переложите его на код системы программирования DELPHI.

Пример «Электронный учебник: Основы алгоритмизации»

Для начала нужно создать htm-файлы, которые будут отображаться при нажатии на определённую тему, или практическую работу.

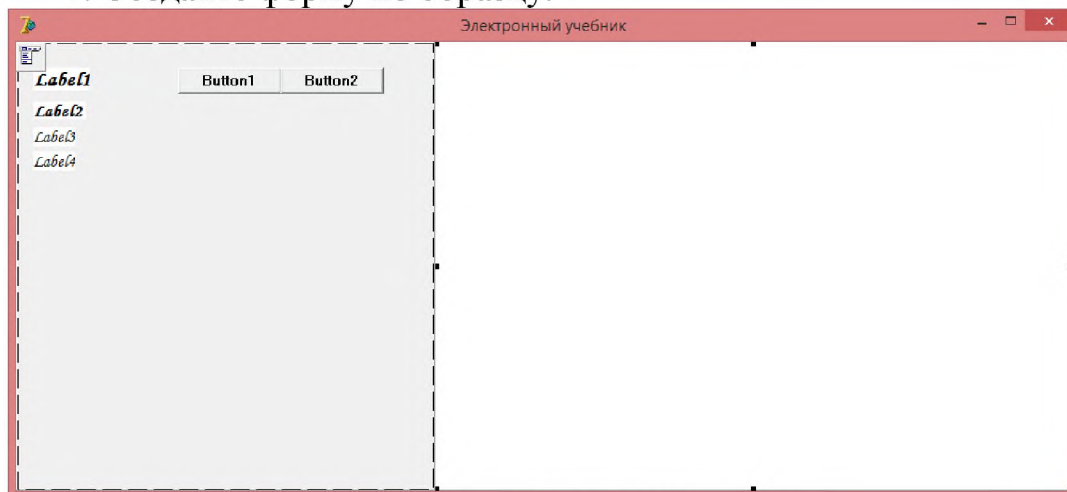
Для этого, откройте в MSWord документ с необходимой темой, и выполните следующие действия:

1. Щёлкните по кнопке **Office**.
2. Выберите пункт «Сохранить как».
3. В открывшемся контекстном меню выбрать пункт «Другие форматы».
4. Откроется окно в котором будет предложено выбрать имя для нового файла, и расширение.
5. В строчке «Тип файла» выбрать «Веб-страница с фильтром»
6. Прodelайте эти действия с каждым документом, который будет использован в программе.

Файлы **111.htm** и **112.htm** будут использованы для отображения тем теоретической части.

Файлы **1.htm**, **2.htm**, **3.htm** будут использованы для отображения практических работ.

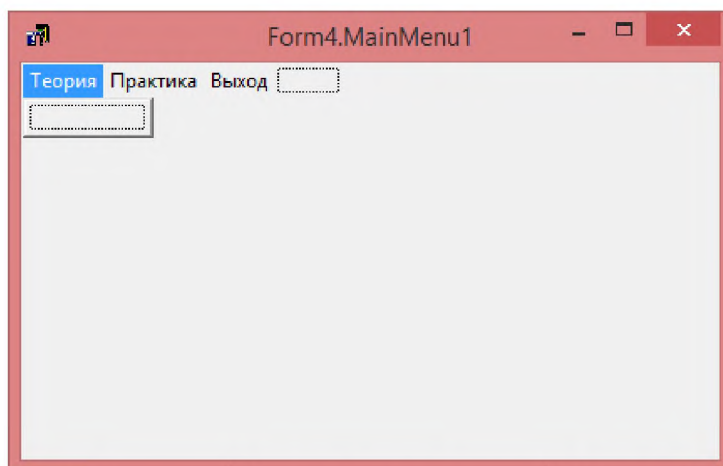
7. Создайте форму по образцу.



| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/имя события | Действие |
|-------------------|-------------------------------|--------------------------|---------------------|
| Form1 | Properties | Caption | Электронный учебник |
| Label1 | Properties | Caption | Теоретическая часть |
| | Properties | Font.Style.fsBold | True |
| Label2 | Properties | Caption | 1.1. Основные |

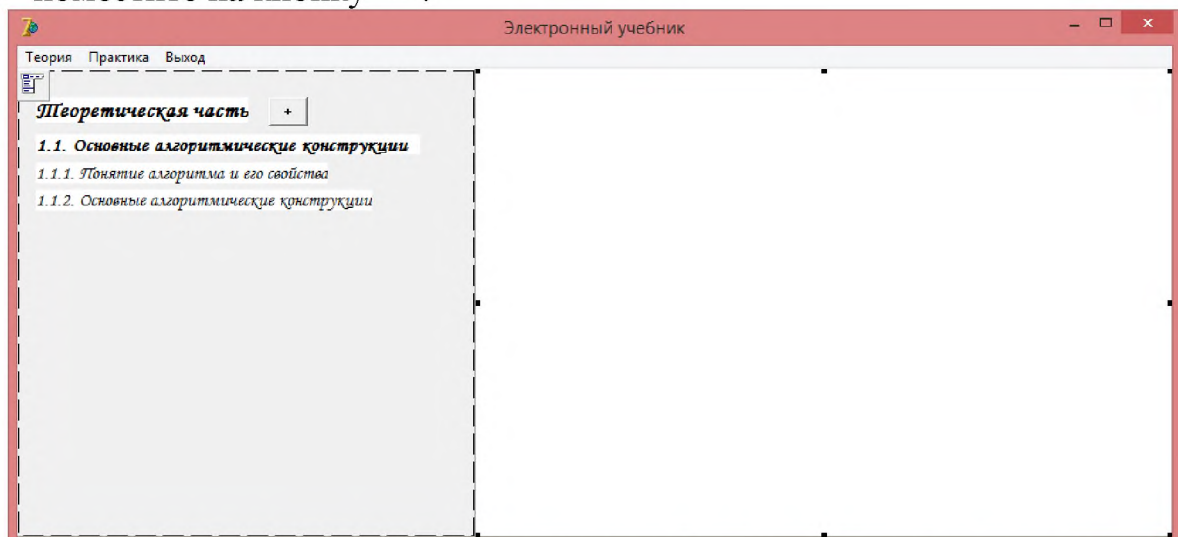
| | | | |
|--|-------------------|---------|---|
| | | | алгоритмические конструкции |
| | Properties | Visible | False |
| Label3 | Properties | Caption | 1.1.1. Понятие алгоритма и его свойства |
| | Events | OnClick | WebBrowser1.Navigate(Extract FilePath(Application.ExeName) + 'html\111.htm'); |
| | Properties | Visible | False |
| Label4 | Properties | Caption | 1.1.2. Основные алгоритмические конструкции |
| | Events | OnClick | WebBrowser1.Navigate(Extract FilePath(Application.ExeName) + 'html\112.htm'); |
| | Properties | Visible | False |
| Button1 | Properties | Caption | - (наложение двух компонент Button1, Button2) |
| | Events | OnClick | Label2.Visible:=False; Label3.Visible:=False; Label4.Visible:=False; Button1.Visible:=False; Button2.Visible:=true; |
| Button2 | Properties | Caption | + (наложение двух компонент Button1, Button2) |
| | Events | OnClick | Label2.Visible:=true; Label3.Visible:=true; Label4.Visible:=true; Button2.Visible:=False; Button1.Visible:=true; |
| Panel1 | | | Выделить компоненты Label1, Label2, Label3, Label4, Button1, Button2 в окне Object TreeView и добавить в Panel1 |
| Panel2 | | | Выделить компонент WebBrowser1 в окне Object TreeView и добавить в Panel2 |
| WebBrowse r1 (вкладка Internet) | | | |
| MainMenu1 | | | См. ниже |

8. Действия для MainMenu1.



| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/ имя события | Действие |
|-------------------|--------------------------------------|---------------------------|--|
| N1 | Properties | Caption | Теория |
| | Events | OnClick | Label2.Visible:=true; Label3.Visible:=true; Label4.Visible:=true; Button2.Visible:=False; Button1.Visible:=true; |
| N2 | Properties | Caption | Практика |
| | Events | OnClick | Form2.Show; Form1.Hide; |
| N3 | Properties | Caption | Выход |
| | Events | OnClick | Close; |

9. Расположите компоненты на первой форме согласно образцу. Кнопку “+” поместите на кнопку “-”.

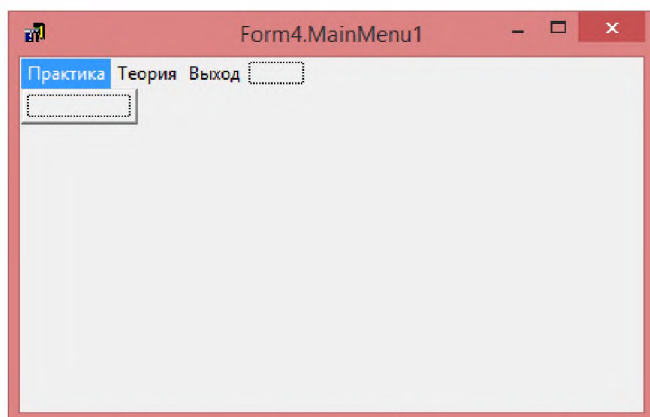


10. Создайте вторую форму (New/Form), расположите компоненты как и на первой форме.

Можно выделить все компоненты формы 1 (Ctrl+A) и скопировать их на новую форму 2 (Ctrl+C, Ctrl+V)

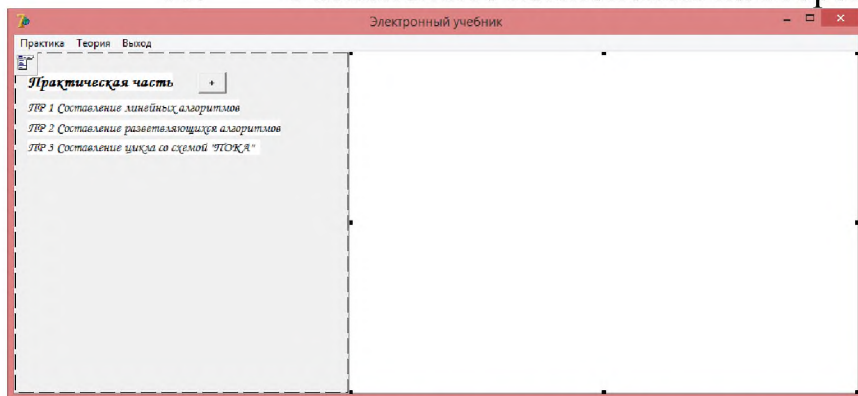
| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/ имя события | Действие |
|-------------------|--------------------------------------|---------------------------|---------------------|
| Form2 | Properties | Caption | Электронный учебник |

| | | | |
|-------------------------|-------------------|-----------------------|---|
| Label1 | Properties | Caption | Практическая часть |
| | Properties | Font.Style.fsBo ld | True |
| Label2 | Properties | Caption | ПР 1 Составление линейных алгоритмов |
| | Events | OnClick | WebBrowser1.Navigate(ExtractF ilePath(Application.ExeName)+' html\1.htm'); |
| | Properties | Visible | False |
| Label3 | Properties | Caption | ПР 2 Составление разветвляющихся алгоритмов |
| | Events | OnClick | WebBrowser1.Navigate(ExtractF ilePath(Application.ExeName)+' html\2.htm'); |
| | Properties | Visible | False |
| Label4 | Properties | Caption | ПР 3 Составление цикла со схемой "ПОКА" |
| | Events | OnClick | WebBrowser1.Navigate(ExtractF ilePath(Application.ExeName)+' html\3.htm'); |
| | Properties | Visible | False |
| Button1 | Properties | Caption | - |
| | Events | OnClick | Label2.Visible:=False; Label3.Visible:=False; Label4.Visible:=False; Button1.Visible:=False; Button2.Visible:=true; |
| Button2 | Properties | Caption | + |
| | Events | OnClick | Label2.Visible:=true; Label3.Visible:=true; Label4.Visible:=true; Button2.Visible:=False; Button1.Visible:=true; |
| Panel1 | | | Выделить компоненты Label2, Label3, Label4, Button1, Button2 в окне Object TreeView и добавить в Panel1 |
| Panel2 | | | Выделить компонент WebBrowser1 в окне Object TreeView и добавить в Panel2 |
| WebBrowse r1 | | | |
| MainMenu1 | | | См. ниже |

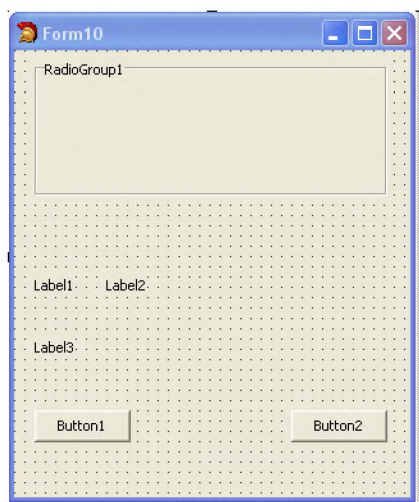


| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/имя события | Действие |
|-------------------|-------------------------------|--------------------------|---|
| N1 | Properties | Caption | Практика |
| | Events | OnClick | Label2.Visible:=true Label3.Visible:=true; Label4.Visible:=true; Button2.Visible:=False; Button1.Visible:=true; |
| N2 | Properties | Caption | Теория |
| | Events | OnClick | Form2.Hide; Form1.Show; |
| N3 | Properties | Caption | Выход |
| | Events | OnClick | Close; |

11. Расположите компоненты на второй форме согласно образцу.



12. Следующим шагом будет создание теста. Создайте новую форму и расположите на ней следующие компоненты.



| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/имя события | Действие |
|---|-------------------------------|--------------------------|--|
| Form1 | Properties | Caption | Тест 1 |
| RadioGroup1 | Properties | Caption | Название вопроса |
| | Properties | Items | Вписать варианты ответов на вопрос |
| | Events | OnClick | a1:=0; if RadioGroup1.ItemIndex=0 then a1:=a1+1 else a1:=a1; Form2.RadioGroup2.Enabled:=True; |
| RadioGroup2 (Итак далее по количеству заданий) | Properties | Caption | Название вопроса |
| | Properties | Items | Вписать варианты ответов на вопрос |
| | Events | OnClick | a2:=0; if RadioGroup2.ItemIndex=0 then a2:=21+1 else a2:=a2; Form2.RadioGroup3.Enabled:=True; |
| | | | |
| Label1 | Properties | Caption | Количество баллов |
| Label2 | Properties | Caption | |
| Label3 | Properties | Caption | |
| Button1 | Properties | Caption | Выход |
| | Events | OnClick | Form3.Close; |
| Button2 | Properties | Caption | Завершить тестирование |
| | Events | OnClick | score:=(a1+a2+a3+a4+a5+a6+a7+a8+a9+a10+a11+a12+a13+a14 |

| | | | |
|--|--|--|---|
| | | | <pre> +a15+a16+a17+a18+a19+a20+a21+a22+a23+a24+a25); if ((score<=27)and(score>=25)) then begin Label2.Caption:=IntToStr(score); Label3.Caption:='Отлично!'; end else if ((score<25)and(score>=22)) then begin Label2.Caption:=IntToStr(score); Label3.Caption:= ('Хорошо! Но Вам необходимо повторить теоретический материал. '); end else if ((score<22)and(score>=19)) then begin Label2.Caption:=IntToStr(score); Label3.Caption:='Плохо! Учите!'; end else if ((score<19)and(score>=0)) then begin Label2.Caption:=IntToStr(score); Label3.Caption:='Очень плохо! Задумайтесь над этим!'; end; end; </pre> |
|--|--|--|---|

В событии кнопки Button2 указываются переменные от a2 до a25. Они означают количество вопросов в тесте. Критерии оценки выбрать самостоятельно.

Самостоятельная работа №3

Работа с графикой и средствами мультимедиа

Цель работы

- рассмотреть практическое применение классов для реализации графических операций;
- на примере рассмотреть методы рисования.

Задание 1. Разработка проекта «Движущиеся объекты»

На основе примера кроссворда «Синее море, белый пароход» разработать свою модель движущегося объекта, совместно с преподавателем и переложите его на код системы программирования DELPHI.

Шкала оценки самостоятельной работы:

оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

Задание 3. Тестовое задание

Список тестовых вопросов

1. Задание метки текста из редактора ввода:
 - a. Memo1.Caption := Edit1.Text;
 - b. **Label1. Caption := Edit1.Text;**
 - c. Form1. Caption := Edit1.Text;

2. Обнуление строки ввода:
 - a. **Edit1.Text :='';**
 - b. Edit1.Text := TMemo;
 - c. Edit1.Text := 'Закругляемся'
3. Передача фокуса ввода на редактор ввода
 - a. Label1.SetFocus;
 - b. Edit1.Add;
 - c. **Edit1.SetFocus;**
4. Открыть файл проекта Project1 можно, нажав:
 - a. CTRL+F4
 - b. CTRL+ALT+F10
 - c. **CTRL+F12**
 - d. ALT+F12
5. Для обозначения комментария не используются:
 - a. (*комментарий*)
 - b. \комментарий/
 - c. {комментарий}
 - d. //комментарий
6. Зарезервированное слово, объявляющее блок подключаемых к проекту модулей:
 - a. Program
 - b. Begin..end
 - c. **Uses**
 - d. Forms
7. Дополнить предложение. Объект Application определяет ...
 - a. **Свойства и методы программы-приложения для Windows**
 - b. Отображение размещения формы
 - c. Перемещение по программному коду
8. Метод Application.Run:
 - a. Создает окно формы, регистрирует входящие в него компоненты
 - b. **Подключает автоматический цикл обработки сообщений Windows к приложению**
 - c. Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия
9. Контейнер, содержащий все элементы программы:
 - a. Unit1
 - b. **Form1**
 - c. Edit1
 - d. Memo1
10. Дополнить предложение. Свойство **Position** определяет ...
 - a. **Размещение и размер формы**
 - b. Подключение модулей

- с. Цвет шрифта
- 11. **poDesigned:**
 - а. Форма выводится в центр экрана
 - б. Windows автоматически определяет начальную позицию и размеры формы
 - с. **Форма отображается в той позиции и с теми размерами, которые были установлены при её конструировании**
- 12. Определяет способ выравнивания компонента внутри контейнера:
 - а. Свойство Position
 - б. **Свойство Align**
 - с. Свойство Memo
 - д. Свойство ScrollBars
- 13. **alClient:**
 - а. **Компонент занимает всю поверхность контейнера**
 - б. Выравнивание не используется
 - с. Компонент помещается в нижнюю часть контейнера
- 14. **alCustom:**
 - а. Компонент помещается в верхнюю часть контейнера
 - б. Компонент помещается в нижнюю часть контейнера
 - с. **Размеры и положение компонента устанавливаются разработчиком**
- 15. Объект **Memo1** - ...
 - а. Однострочный редактор
 - б. **Многострочный редактор, содержащий несколько строк текста**
 - с. Текстовая надпись
- 16. Свойство **ScrollBars:**
 - а. **Задаёт в поле редактирования полосы прокрутки**
 - б. Определяет способ выравнивания компонента внутри контейнера
 - с. Задаёт имя объекта
- 17. Функциональная кнопка с изображением устанавливается из:
 - а. Палитра компонентов -> Standart -> Button
 - б. Палитра компонентов -> Standart -> Label
 - с. **Палитра компонентов -> Additional -> BitBtn**
- 18. Дополнить предложение. За стиль оформления внешней и внутренней рамок отвечают свойства ... (несколько вариантов ответа)
 - а. Align
 - б. **BevelOuter**
 - с. **BevelInner**
 - д. ssBoth
 - е. Caption
- 19. Установить соответствие

| | |
|----------------|--|
| bvLowered | Компонент помещается в нижнюю часть контейнера |
| poScreenCenter | Есть обе полосы прокрутки |
| bvNone | Рамка вдавлена |

| | |
|--|---|
| alBottom | Форма выводится в центре экрана, её высота и ширина не изменяются |
| ssBoth | Рамка отсутствует |
| 1 – c, 2- d, 3 – e, 4 – a, 5 -b | |

20. VCL – это...

- a. Приложение Delphi
- b. Алгоритмизация решения задачи
- c. Библиотека визуальных компонентов**
- d. Характеристики объектов

21. Дополнить предложение. Компоненты, которые видны на форме, как во время создания приложения, так и во время работы приложения называются ...

- a. Визуальными**
- b. Реальными
- c. Невизуальными
- d. Основными

22. К невидимым компонентам относят:

- a. Кнопки, метки, списки блоков
- b. Таймеры, компоненты для работы с базами данных, списки изображений**
- c. Библиотеки, звуки, коды

23. Установить соответствие:

| | |
|---|---|
| Standart | Обеспечивает доступ к 32-битным элементам Windows |
| Additional | Компоненты, реализующие интерфейс с пользователем и процесс управления данными для БД |
| Win32 | Включает стандартные компоненты, обеспечивающие некоторые функции интерфейса пользователя |
| System | Специализированные компоненты, организующие доступ к БД |
| DataAccess | Набор компонентов для доступа к системным ресурсам (OLE, DDE) |
| DataControl | Дополнительные интерфейсные компоненты для красочного оформления приложения |
| – c, 2) – f, 3) – a, 4) – e, 5) – d, 6) – b; | |

24. Дополнить предложение. Имя компонента состоит из... (несколько вариантов ответа)

- a. Названия компонента**
- b. Заготовки компонента
- c. Порядкового номера компонента**
- d. Произвольного значения ItemIndex

25. Дополнить предложение. Свойства Height и Width показывают...

- a. Какое действие должен выполнить программист
- b. Размеры компонента**
- c. Исполняемый блок программы

26. Дополнить предложение. Проект Delphi представляет собой...

- a. Набор программных единиц – модулей**
- b. Основной интерфейсный элемент в Delphi

с. Атрибуты (основные характеристики), которые описывают особенности объекта

27. Модуль – это...

а. Средство для создания, изменения исходных файлов, которые содержат написанную на языке программирования программу

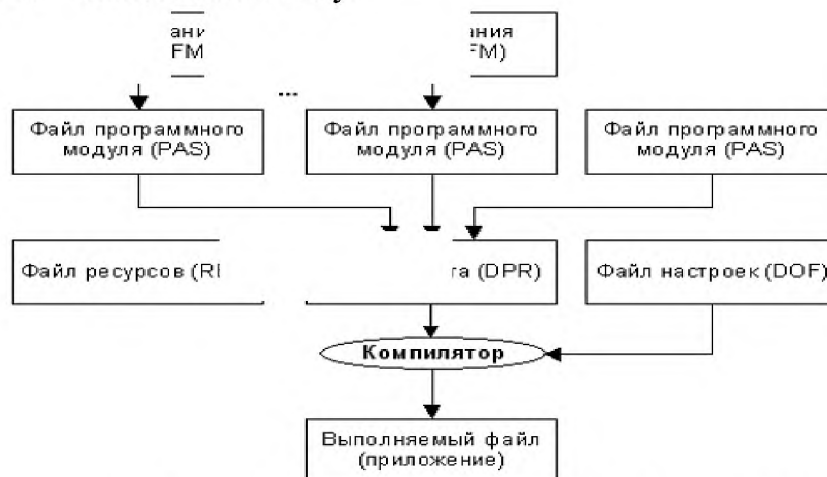
б. Автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры и функции) и, возможно некоторые исполняемые операторы иницилирующей части

с. Набор инструментов, которые используются для преобразования символов

28. Установить соответствие

| | |
|---|---|
| Файл проекта | (.pas) соответствующий файл модуль для хранения кода |
| Файл модуля | (.res) содержит пиктограмму и прочие ресурсы |
| Файл формы | (.dpr) текстовый файл используется для хранения информации о формах и модулях, содержит операторы инициализации и запуска программы на выполнение |
| Файл опций проекта | (.dfm) двоичный файл, который создается для хранения информации о ваших формах и фреймах |
| Файл ресурсов | (.cfg) хранит установки проекта |
| Файл конфигурации проекта | (.dof) хранит установки опций проекта |
| – с, 2) – а, 3) – d, 4) – f, 5) – b, 6) - e | |

29. Заполнить схему:



а. 1 – файл формы (DFM); 2 – файл проекта (DPR)

б. 1 – файл проекта (DPR); 2 – файл формы (DFM)

с. 1 – файл формы (DPR); 2 – файл проекта (DFM)

д. 1 – файл проекта (DFM); 2 – файл формы (DPR)

30. Дополнить предложение. Набор всех файлов, необходимых для создания приложения, называется...

а. Модулем

б. Проектом

с. Задачей

д. Формой

31. Файлы описания форм – это...

- a. Отклики на внешнее воздействие
 - b. Файлы подключающие все используемые программные модули
 - c. **Текстовые файлы с расширением DFM, в котором сохраняются значения свойств формы и ее компонентов**
- 32. Содержимое dfm-файла вызывается:
 - a. **Контекстное меню формы -> View as Text**
 - b. Контекстное редактора кода -> View as Form
 - c. File -> Save as...
- 33. Дополнить предложение. При сборке приложения описание из DFM-файла помещается в...
 - a. Файл модуля
 - b. **Область ресурсов**
 - c. Область установок проекта
- 34. Программный модуль (Unit) содержит:
 - a. **Все относящиеся к форме объявления и методы обработки событий, написанные на языке Object Pascal**
 - b. Заготовку, макет одного из окон разрабатываемого приложения
 - c. Пиктограммы, которые представляют компоненты VCL , которые можно включить в приложение
- 35. Дополнить предложение. Имя модуля должно совпадать с...
 - a. Приложением Windows
 - b. Спецификацией задачи
 - c. Объектом в Delphi
 - d. **Именем файла при сохранении File->Save as...**
- 36. Интерфейсная секция модуля (interface) содержит: (несколько вариантов ответа)
 - a. **Список стандартных модулей библиотеки VCL**
 - b. **Раздел описания типов**
 - c. Компилятор
 - d. **Раздел описания переменных (объявление собственно объекта формы)**
 - e. Инспектор объектов и проектировщик форм
- 37. Дополнить предложение. Класс формы объявляется в разделе...
 - a. **Type**
 - b. Uses
 - c. Project
 - d. Var
- 38. Дополнить предложение. В описании класса форм помещенные на форму компоненты представлены...
 - a. Границами формы
 - b. **Полями формы**
 - c. Индексами формы
- 39. Published – это...

- a. Атрибут видимости, позволяющий работать с компонентами и методами обработки событий на визуальном уровне
 - b. Директива препроцессора
 - c. Текст программы на языке Object Pascal
40. Пустые секции, в которых можно размещать любые вспомогательные поля, методы и свойства: *(несколько вариантов ответа)*
- a. **Private**
 - b. Uses
 - c. Var
 - d. **Public**
 - e. Program
41. Объявление объекта формы:
- a. unit Unit1;
 - b. type TForm1 = class(TForm)
 - c. **var Form1: TForm1;**
42. implementation – это...
- a. Тип данных Delphi
 - b. Племя в Южной Африке
 - c. **Раздел реализации объявлений интерфейса**
43. Подключение файла описания формы:
- a. **{R *.dfm}**
 - b. {\$R *.pas}
 - c. {R *.dpr}

Шкала оценки тестовых вопросов:

- «2» – от 0 до 40%
- «3» – от 41% до 60%
- «4» – от 61% до 80%
- «5» – от 81% до 100%

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

языке ассемблера»

Практическая работа №8 «Структура программы на C++. Константы и переменные»

Практическая работа №9 «Организация ввода и вывода на C++»

Практическая работа №10 «Оператор выбора и организация ветвления на C++. Решение задач на организацию ветвления»

Практическая работа №11 «Операторы цикла в программах на C++. Цикл со счетчиком»

Практическая работа №12 «Циклы с предусловием и с постусловием. Выбор типа цикла»

Практическая работа №13 «Процедуры и функции»

Практическая работа №14 «Массивы. Одномерные и двумерные массивы.»

Практическая работа №15 «Динамические массивы Алгоритмы поиска в массивах. Перебор элементов массива.»

Практическая работа №16 «Классические алгоритмы сортировки массивов. Строковый тип данных в программах на C++»

Шкала оценки практических работ:

оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

Задание 5. Рефераты

Перечень тем рефератов:

1. Общие понятия об алгоритмизации.
2. Процесс создания программы.
3. Назначение и характеристика современных языков программирования.
4. Особенности и характеристика языка C++.
5. Типы данных и операции в языке C++
6. Структуры в языке C++
7. Функции ввода-вывода в языке C++
8. Операторы условия в языке C++
9. Операторы цикла в языке C++,

Шкала оценки рефератов:

Критерии и показатели, используемые при оценивании учебного реферата

| Критерии | Показатели |
|---|---|
| 1. Новизна реферированного текста Макс. - 20 баллов | <ul style="list-style-type: none"> - актуальность проблемы и темы; - новизна и самостоятельность в постановке проблемы, в формулировании нового аспекта выбранной для анализа проблемы; - наличие авторской позиции, самостоятельность суждений. |
| 2. Степень раскрытия сущности проблемы Макс. - 30 баллов | <ul style="list-style-type: none"> - соответствие плана теме реферата; - соответствие содержания теме и плану реферата; - полнота и глубина раскрытия основных понятий проблемы; - обоснованность способов и методов работы с материалом; - умение работать с литературой, систематизировать и структурировать материал; - умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы. |
| 3. Обоснованность выбора источников Макс. - 20 баллов | <ul style="list-style-type: none"> - круг, полнота использования литературных источников по проблеме; - привлечение новейших работ по проблеме (журнальные публикации, материалы сборников научных трудов и т.д.). |
| 4. Соблюдение требований к оформлению Макс. - 15 баллов | <ul style="list-style-type: none"> - правильное оформление ссылок на используемую литературу; - грамотность и культура изложения; - владение терминологией и понятийным аппаратом проблемы; - соблюдение требований к объему реферата; - культура оформления: выделение абзацев. |
| 5. Грамотность Макс. - 15 баллов | <ul style="list-style-type: none"> - отсутствие орфографических и синтаксических ошибок, стилистических погрешностей; - отсутствие опечаток, сокращений слов, кроме общепринятых; - литературный стиль. |

Оценивание реферата

Реферат оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

- 86 – 100 баллов – «отлично»;
- 70 – 75 баллов – «хорошо»;
- 51 – 69 баллов – «удовлетворительно»;

- мене 51 балла – «неудовлетворительно».

Баллы учитываются в процессе текущей оценки знаний программного материала.

Раздел 3. Отладка, тестирование и оптимизация программных модулей

Задание 1. Перечень контрольных вопросов по разделу

1. Что такое данные?
2. Что такое символьные строки?
3. Что такое директива `#define`?
4. Дайте определение понятию «операции»
5. Что такое операторы?
6. Какие операторы вы знаете?
7. Что такое преобразование типов?
8. Для чего нужно переключение ввода-вывода?
9. Что такое выбор вариантов?
10. Что такое цикл?
11. Какие вы знаете управляющие средства?
12. Что такое структурное программирование?
13. Что такое препроцессор языка C++?
14. Что такое файлы ввода-вывода?
15. Что такое символьные строки?
16. Какие функции можно совершить над символьными строками?
17. Возможности C++ в системном программировании

Шкала оценки контрольных вопросов:

Оценка 5 «**отлично**» выставляется студенту, если он правильно ответил на поставленный вопрос.

Оценка 4 «**хорошо**» выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка 3 «**удовлетворительно**» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки.

Оценка 2 «**неудовлетворительно**» выставляется студенту, который не знает значительной части заданного вопроса.

Задание 2. Самостоятельная работа

Самостоятельная работа №4

Разработка и проектирование баз данных

Цель работы: освоить технологию создания базы данных и наполнения ее информацией в системе DELPHI.

Задание 1. Разработка проекта «База данных»

Напишите программу работы с локальной базой данных.

Вариант 1. Учет авиапассажиров.

Таблица авиапассажиров: код авиапредприятия, код города назначения, номер рейса, дата вылета, фамилия, адрес пассажира.

Справочники: авиапредприятия (код, наименование, адрес, телефон), города (код и наименование).

Отчет по авиапредприятиям и городам с числом пассажиров по авиапредприятиям и городам и в целом по отчету.

Вариант 2. Учет выпуска изделий.

Таблица выпущенных изделий: код изделия, код предприятия, количество выпущенных изделий (шт.), их себестоимость, отпускная цена, затраты на электроэнергию, затраты на зарплату, дата выпуска.

Справочники: изделия (код, наименование), предприятия (код, наименование, адрес, телефон).

Отчет по изделиям и предприятиям со средними затратами по каждому изделию.

Вариант 3. Учет поставок товаров.

Таблица поставленных товаров: код поставщика, код товара, количество, цена, единица измерения, дата поставки.

Справочники: поставщики (код, наименования, адрес, телефон), товары (код, наименование).

Отчет по товарам и поставщикам с итоговой стоимостью по товарам, поставщикам и по отчету.

Вариант 4. Учет уволившихся с предприятия.

Таблица уволенных: код подразделения, код причины увольнения, дата увольнения, фамилия, номер приказа.

Справочники: подразделения (код, наименование), причины увольнения (код и наименование).

Отчет по причинам увольнения и подразделениям с итоговым числом уволившихся по причинам увольнения, подразделениям и по отчету.

Вариант 5. Учет рождаемости.

Таблица новорожденных: код населенного пункта, код национальности, вес, рост новорожденного, дата рождения, фамилия, имя, отчество.

Справочники: населенные пункты (код, наименование), национальности (код, наименование).

Отчет по населенным пунктам и национальностям с итоговым числом новорожденных по населенным пунктам, национальностям и по отчету.

Вариант 6. Учет участников Олимпиады.

Таблица олимпийцев: код страны, код вида спорта, фамилия участника, дата рождения, рост, вес, пол.

Справочники: страны (код, наименование, начальник команды, телефон), виды спорта (код, наименование).

Отчет по странам и видам спорта с итогами по странам и видам спорта: число участников, средний вес, средний рост.

Вариант 7. Учет проданных товаров.

Таблица проданного товара: код торгового предприятия, код товара, стоимость проданного товара за последние 3 года (три поля).

Справочники: торгующие предприятия (код, наименование, директор, телефон), товары (код, наименование).

Отчет по товарам и торгующим предприятиям с итогами по товарам и предприятиям.

Вариант 8. Учет малых предприятий.

Таблица малых предприятий: код населенного пункта, код основного вида деятельности МП, численность работающих, адрес, объем выполняемых работ за год (всего) и по основному виду деятельности, директор, дата создания.

Справочники: населенные пункты (код, наименование), виды деятельности (код, наименование).

Отчет по населенным пунктам и видам деятельности с итогами по видам деятельности, населенным пунктам и по отчету в целом.

Вариант 9. Учет больных в больнице.

Таблица больных: фамилия больного, код диагноза, дата начала лечения в больнице, адрес, код подразделения, стоимость лечения за один день, дата выписки, дата рождения больного.

Справочники: диагнозы (код, наименования), подразделения (код наименования, заведующий).

Отчет стоимости лечения по подразделениям и диагнозам с итогами стоимости лечения по подразделениям, диагнозам и по больнице в целом.

Вариант 10. Учет движения общественного транспорта.

Таблица движения общественного транспорта: код типа транспорта, номер и длина маршрута, табельный номер водителя, дата выхода.

Справочники: типы транспорта (код, наименования), водители (табельный номер, фамилия).

Отчет по видам транспорта и маршрутам с итогами (километраж) по видам транспорта, маршрутам и по отчету.

Вариант 11. Учет платежных поручений в банке.

Таблица платежных поручений: код клиента, код вида платежа, дата, сумма, номер поручения.

Справочники: клиенты банка (код, наименование, адрес, директор, телефон), виды платежей (код, наименование).

Отчет по клиентам и видам платежа с итогами по клиентам, видам платежа и в целом по отчету.

Вариант 12. Учет договоров займа.

Таблица договоров займа: код организации, номер договора, дата заключения, дата окончания, сумма, процент, сумма возврата, возвращенная сумма, особые условия, процент штрафа за каждый просроченный день, код денежной единицы.

Справочники: организации (код, наименование, адрес, директор, телефон); денежные единицы (код, обозначение, наименование).

Отчет по видам валют и дням возврата с итогами по видам валют и дням возврата.

Вариант 13. Учет проданных ценных бумаг.

Таблица проданных ценных бумаг: код эмитента, код типа ценной бумаги, число, номинальная стоимость, курсовая стоимость, дата выпуска, номер регистрации выпуска ценных бумаг.

Справочники: эмитенты (код, наименование, адрес, директор, телефон), типы ценных бумаг (код, наименование).

Отчет по типам ценных бумаг и эмитентам с итогами по типам ценных бумаг и эмитентам: суммарное число бумаг, средняя стоимость.

14. Учет кадров.

Таблица сотрудников предприятия: табельный номер работника, фамилия, дата рождения, код специальности, код подразделения, дата приема, дата начала трудовой деятельности, оклад, образование.

Справочники: специальности (код, наименование); подразделения (код, наименование, фамилия начальника, телефон).

Отчет по подразделениям и специальностям с итогами (число принятых) по подразделениям, специальностям и по предприятию.

Вариант 15. Учет очередников на получение жилья.

Таблица очередников: фамилия очередника, номер очереди, номер приказа, код организации, код основания включения в очередь, дата, паспортные данные, число членов семьи, адрес.

Справочники: организации (код, наименование, адрес, директор, телефон); основания включения в очередь (код, наименование).

Список очередников по предприятиям и основаниям включения в очередь и числом очередников по предприятиям и основаниям включения в очередь.

Вариант 16. Учет покупки и продажи валюты.

Таблица продажи и покупки валюты: дата, фамилия, паспортные данные, код валюты, количество валюты, проданной и купленной, количество рублей, код обменного пункта.

Справочники: валюта (код, наименование, текущий курс) и обменные пункты (код, наименование).

Отчет по видам валют и обменным пунктам с итогами по видам валют и обменным пунктам (сумма проданной и купленной валюты).

Вариант 17. Учет объектов строительства.

Таблица объектов строительства: код населенного пункта, сметная стоимость, стоимость выполненных работ, дата начала строительства, наименование объекта, код заказчика.

Справочники: населенные пункты (код, наименование), заказчики (код, наименование, адрес, телефон).

Отчет по заказчикам и населенным пунктам с итогами по заказчикам, населенным пунктам и по отчету в целом (сметная стоимость и объем выполненных работ).

Вариант 18. Учет книг в библиотеке.

Таблица книг: код темы, наименование книги, авторы, дата, число экземпляров, адрес хранения, код издательства.

Справочники: темы (код, наименование), издательства (код, наименование, адрес, телефон).

Отчет по темам и издательствам с итоговым числом книг по темам, издательствам и по отчету в целом.

Вариант 19. Учет успеваемости студентов.

Таблица успеваемости студентов: код предмета, дата, оценка, номер зачетной книжки.

Справочники: предметы (код и наименование), студенты (группа, номер зачетной книжки, фамилия).

Отчет по группам и студентам со средней оценкой по группам, студентам и в целом по ВУЗу.

Вариант 20. Каталог почтовых марок.

Концептуальные требования:

Номер Год издания Тематика Ф.И.О. художника Сюжет

Номинал Стоимость Тираж Способ печати Размеры Фото

Режимы работы:

Сортировка по полям – Номер, Год издания, Тематика, Способ печати.

Поиск по полям - ФИО художника, Номер, Номинал.

Фильтрация по полям на диапазон значений- ФИО художника, Год издания, Способ печати.

Вариант 21. Каталог монет России.

Концептуальные требования:

Номер Номинал Материал Размер Вес Год выпуска

Ф.И.О. гравера Монетный двор Тираж Редкость Фото

Режимы работы:

Сортировка по полям – Номер, Год выпуска, Материал, Монетный двор.

Поиск по полям - ФИО гравёра, Номер, Номинал, Редкость.

Фильтрация по полям на диапазон значений - ФИО гравёра, Год выпуска, Материал, Номинал

Вариант 22. Каталог сыров.

Концептуальные требования:

Наименование Категория Способ хранения Срок хранения Употребление

Форма выпуска Завод изгот. Ф.И.О. директора Адрес завода Фото

Режимы работы:

Сортировка по полям – Наименование, Завод изготовитель, Срок хранения.

Поиск по полям - ФИО директора, Наименование.

Фильтрация по полям на диапазон значений - Наименование, Категория, Срок хранения, Завод изготовитель, Форма выпуска.

Вариант 23. Автомобили завода ГАЗ.

Концептуальные требования:

Наименование Тип Кузов Двигатель Ходовые мосты

Дата начала вып. Кол-во Ф.И.О. инженера Масса Фото

Режимы работы:

Сортировка по полям – Наименование, Двигатель, Дата начала выпуска

Поиск по полям - ФИО инженера, Наименование, Масса.

Фильтрация по полям на диапазон значений - Тип, Кузов, Двигатель, Дата начала выпуска.

Вариант 24. Исторические здания г. Москвы.

Концептуальные требования:

Наименование Адрес Дата постройки Ф.И.О. архитект. Владелец

Размеры Этажность Фото Дата ремонта

Режимы работы:

Сортировка по полям – Наименование, Дата постройки, Адрес.

Поиск по полям - ФИО архитектора, Дата постройки, Наименование.

Фильтрация по полям на диапазон значений- ФИО архитектора, Дата постройки, Владелец.

Вариант 25. Каталог периодических изданий

Концептуальные требования:

Наименование Ф.И.О. гл. редакт. Вид Цена 1 экз. Периодичность

Объем Тираж Подп. Индекс Объем подписки

Режимы работы:

Сортировка по полям – Наименование, Объём, Тираж, Подписной индекс.

Поиск по полям - Наименование, ФИО главного редактора, Цена.

Фильтрация по полям на диапазон значений - Вид, Периодичность, Тираж,

Самостоятельная работа №5

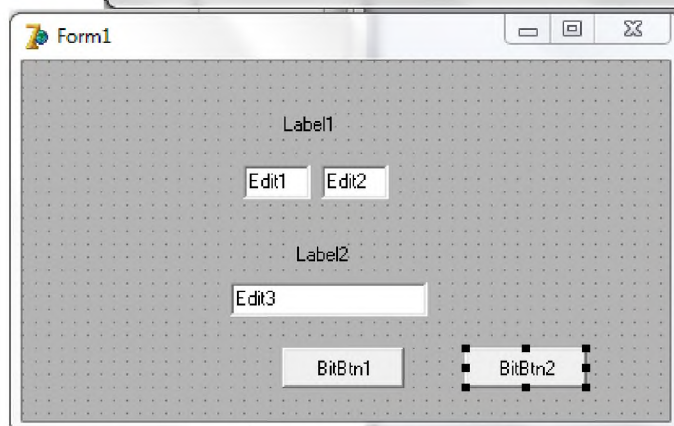
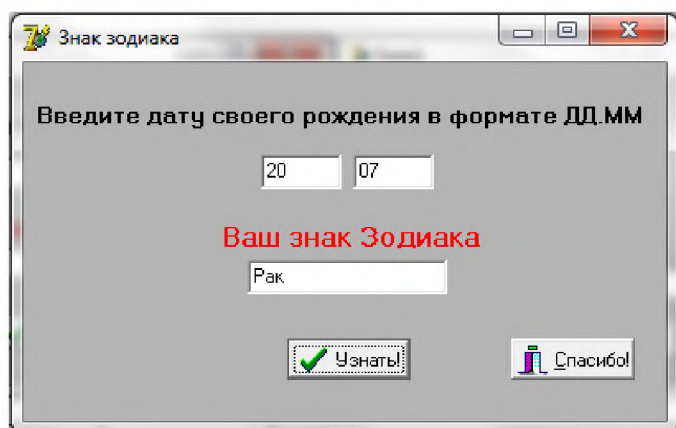
Использование базовых компонентов для решения задач

Цель работы

- рассмотреть практическое применение компонента TEdit (поле ввода данных);
- на примере рассмотреть использование оператора условия.

Задание 1. Разработка проекта «Знак зодиака»

Пользователь вводит в поля Edit1 и Edit2 дату своего дня рождения в формате ДД.ММ. После нажатия кнопки «Узнать!» в поле Edit3 появляется название соответствующего знака Зодиака.



| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/ имя события | Действие |
|-------------------|-------------------------------|---------------------------|---|
| Form1 | Properties | Caption | Название формы «Знак Зодиака» |
| | | Color | blActiveBorder |
| | | Position | poScreenCenter |
| Label1 | Properties | Caption | Название метки «Введите дату своего дня рождения в формате ДД.ММ» |

| | | | |
|----------------|-------------------|----------|--|
| | | Font | 10, полужирный |
| Edit1 | Properties | Text | |
| | | Hint | Текст подсказки «Введите день» |
| | | ShowHint | True |
| Edit2 | Properties | Text | |
| | | Hint | Текст подсказки «Введите месяц» |
| | | ShowHint | True |
| Label2 | Properties | Caption | Название метки «Ваш Зодиак» |
| | | Font | 12, полужирный, красный |
| BitBtn1 | Properties | Kind | bkOK |
| | | Caption | Название кнопки «Узнать!» |
| | Events | OnClick | <pre> var d,m,k:byte; s:string; begin m:=strtoint(Edit2.Text); if not (m in [1..12]) then begin Showmessage('Неверно введен месяц! Повторите ввод. '); Edit2.Clear; Edit2.SetFocus; end; k:=0; d:=strtoint(Edit1.Text); if ((m=2)and(d>29))or((m in [1,3,5,7,8,10,12])and(d>31)) or((m in [4,6,9,11])and(d>30))or(d<1)then begin k:=1; Showmessage('Неверно введен день! Повторите ввод. '); Edit1.Clear; Edit1.SetFocus; end; if (m in [1..12])and(k=0)then begin case m of 1:case d of 1..19:s:='Козерог'; 20..31:s:='Водолей'; </pre> |

| | | |
|--|--|--|
| | | end; 2:case d of 1..18:s:='Водолей'; 19..29:s:='Рыбы'; end; 3:case d of 1..20:s:='Рыбы'; 21..31:s:='Овен'; end; 4:case d of 1..21:s:='Овен'; 22..30:s:='Телец'; end; 5:case d of 1..2:s:='Телец'; 21..31:s:='Близнецы'; end; 6:case d of 1..21:s:='Близнецы'; 22..30:s:='Рак'; end; 7:case d of 1..22:s:='Рак'; 23..31:s:='Лев'; end; 8:case d of 1..22:s:='Лев'; 23..31:s:='Дева'; end; 9:case d of 1..22:s:='Дева'; 23..30:s:='Весы'; end; 10:case d of 1..22:s:='Весы'; 23..31:s:='Скорпион'; end; 11:case d of 1..22:s:='Скорпион'; 23..30:s:='Стрелец'; end; 12:case d of 1..21:s:='Стрелец'; 22..31:s:='Козерог'; end; end; |
|--|--|--|

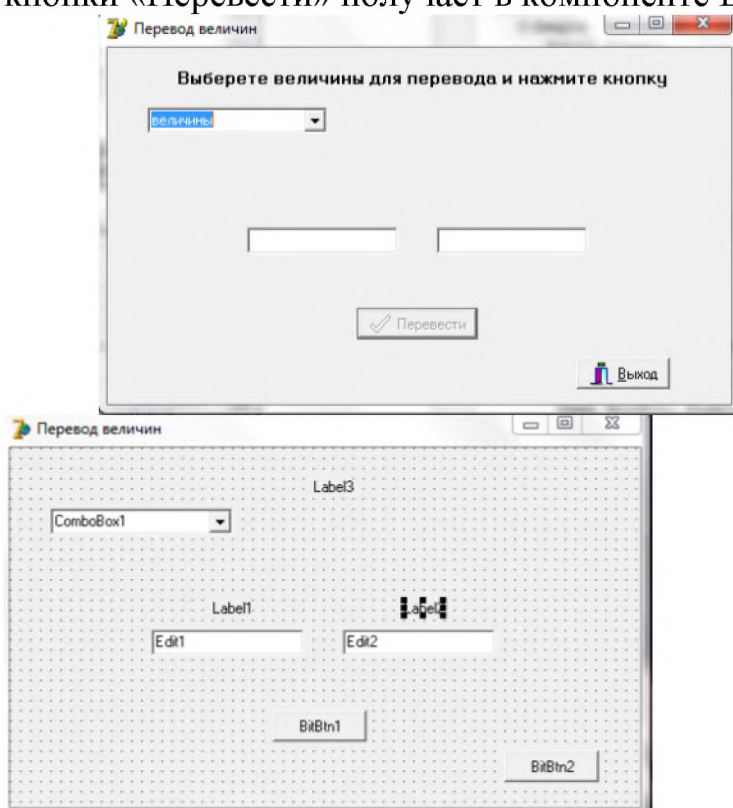
| | | | |
|----------------|-------------------|---------|--------------------------------|
| | | | Edit3.Text:=s; end; end; |
| BitBtn2 | Properties | Kind | bkClose |
| | | Caption | Название кнопки «Спасибо!» |

Задание 2. Разработка проекта «Быстрое умножение»

Программа вычисляет и показывает произведение двух целых чисел. Числа вводит пользователь с клавиатуры. В программе осуществляется контроль ввода данных.

Задание 3. Разработка проекта «Перевод величин»

Пользователь выбирает из компонента ComboBox1 нужные для перевода величины, затем вводит исходные данные в компонент Edit1 и после нажатия кнопки «Перевести» получает в компоненте Edit2 результат.

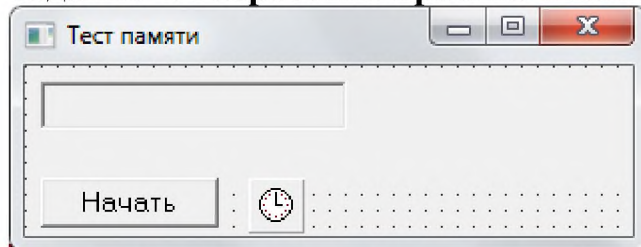


| Выделенный объект | Вкладка окна Object Inspector | Имя свойства/ имя события | Действие |
|-------------------|---|------------------------------|------------------------------------|
| Form1 | Properties | Caption | Название формы «Перевод величин» |
| | | Position | poScreenCenter |
| Label1 | Properties | Caption | |
| Edit1 | Properties | Text | |
| | Events | OnChange | begin if Length(Edit1.Text) = 0 |

| | | | |
|------------------|-------------------|-----------|--|
| | | | then BitBtn1.Enabled := False else BitBtn1.Enabled := True; end; |
| Edit2 | Properties | Text | |
| Label2 | Properties | Caption | |
| Label3 | Properties | Caption | Название метки «Выберите величины для перевода нажмите кнопку» |
| ComboBox1 | Properties | Items | Величины Кг – фунты См – дюймы Цельсий – фаренгейт Рубли - доллары |
| | | ItemIndex | 0 |
| | Events | OnChange | begin case ComboBox1.ItemIndex of 1:begin Edit1.Clear; Edit2.Clear; Label1.Caption := 'кг'; Label2.Caption := 'фунты'; end; 2:begin Edit1.Clear; Edit2.Clear; Label1.Caption := 'см'; Label2.Caption := 'дюймы'; end; 3:begin Edit1.Clear; Edit2.Clear; Label1.Caption := 'C'; Label2.Caption := 'F'; end; 4:begin Edit1.Clear; Edit2.Clear; Label1.Caption := 'руб.'; Label2.Caption := '\$'; end; end; end; |
| BitBtn1 | Properties | Kind | bkOK |
| | | Caption | Название кнопки «Перевести |

| | | | |
|----------------|-------------------|---------|--|
| | | Enabled | False |
| | Events | OnClick | <pre> var od:real; d:real; begin if Length(Edit1.Text) = 0 then begin // в поле Edit1 нет исходной информации ShowMessage('Введите величину'); Edit1.Clear; Edit1.SetFocus; end; case ComboBox1.ItemIndex of 1: begin od := StrToFloat(Edit1.Text); d := od * 2.205; Edit2.Text:= FloatToStr(d); end; 2:begin od := StrToFloat(Edit1.Text); d := od * 0.3937; Edit2.Text:= FloatToStr(d); end; 3:begin od := StrToFloat(Edit1.Text); d := od * 1.8 + 32; Edit2.Text:= FloatToStr(d); end; 4: begin od := StrToFloat(Edit1.Text); d := od / 40; Edit2.Text:= FloatToStr(d); end; end; end; end; end; </pre> |
| BitBtn2 | Properties | Kind | bkClose |
| | | Caption | Название кнопки «Выход» |

Задание 4. Разработка проекта «Память»



Задание 5. Разработка игры «Камень, ножницы, бумага»

Игроки считают вместе вслух «Камень... Ножницы... Бумага... Раз... Два... Три», одновременно качая кулаками. На счёт «Три» они одновременно показывают при помощи руки один из трёх знаков: камень, ножницы или бумагу.

Победитель определяется по следующим правилам:

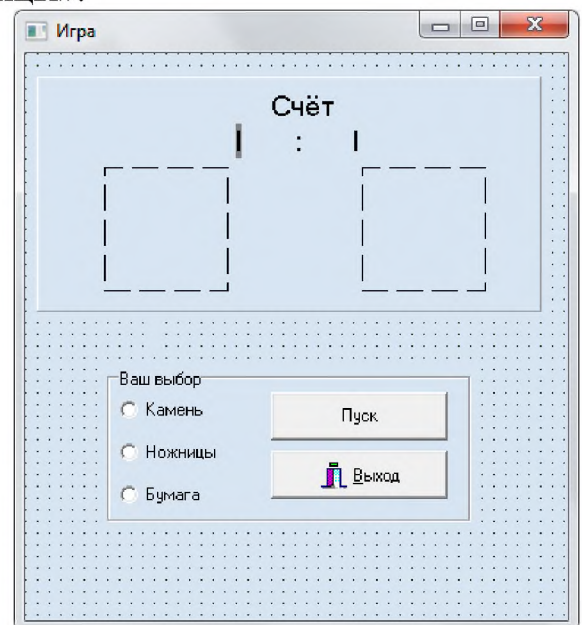
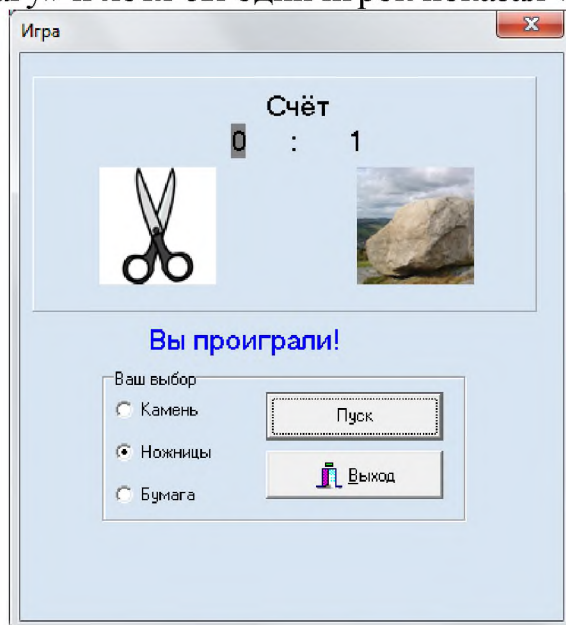
Камень побеждает ножницы («камень затупляет или ломает ножницы»)

Ножницы побеждают бумагу («ножницы разрезают бумагу»)

Бумага побеждает камень («бумага заворачивает камень»)

Если игроки показали одинаковый знак, то засчитывается ничья и игра переигрывается.

В классическом варианте в игру играют вдвоём, однако возможна игра большего количества участников. При этом ничья засчитывается в ситуации, когда одновременно хотя бы один игрок показал «камень», хотя бы один игрок показал «бумагу» и хотя бы один игрок показал «ножницы».



Шкала оценки самостоятельной работы:

оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

Задание 3. Тестовые задания к разделу

Список тестовых вопросов

1. Дополнить предложение. После подключения файла описания формы следует...
 - a. Раздел описания типов
 - b. Атрибут видимости
 - c. Описание размеров используемых компонентов
 - d. Реализация методов обработки событий**
2. Установить метод обработки события Button1Click: *(несколько вариантов ответа)*
 - a. Двойной щелчок по компоненту Button1 на форме**
 - b. Выделить Button1 ->Events -> OnClick**
 - c. Выделить BitBtn1 ->Events -> OnClick
3. Программный файл проекта – это...
 - a. Программист

- b. Главная часть приложения, с которой начинается выполнение программы и которая обеспечивает инициализацию других модулей
- c. Пустая секция
4. К комментариям не относятся: *(несколько вариантов ответа)*
- a. **Program**
- b. {Form1}
- c. {\$R *.res}
- d. //выполнение команды
5. Дополнить предложение. Сочетание end со следующей за ней точкой называется...
- a. Терминалом программной единицы
- b. **Терминатором программной единицы**
- c. Андроидом программной единицы
6. Содержимое DPR-файла вызывается: *(несколько вариантов ответа)*
- a. Ctrl+alt
- b. File -> New->Application
- c. **Project -> View Source**
- d. **Ctrl+F12**
7. Дополнить предложение. Зарезервированное слово uses сообщает компилятору о...
- a. **Модулях, которые должны рассматриваться как необъемлемые части программы и которые содержаться в других файлах**
- b. Начале работы приложения
- c. Ошибках, которые могут возникнуть при компилировании
8. Модуль Forms – это...
- a. Дополнительный модуль, содержащий неиспользуемые компоненты
- b. Модуль для построения более красочного приложения
- c. **Обязательный для всех программ модуль, который содержит определение объекта Application**
9. Запись {\$R *.res} означает:
- a. Комментарий, указывающий программисту где искать ресурсы
- b. **Директиву, подключающую к результирующему выполняемому файлу ресурсы**
- c. Директиву, подключающую к результирующему выполняемому файлу форму
10. Установить соответствие:

| | |
|----------------------------|---|
| Application.Initialize | загружает и инициализирует форму |
| Application.CreateForm | активизирует форму и начинает выполнение приложения |
| Application.Run | подготавливает приложение к работе |
| 1 – c; 2 – a; 3 – b | |

11. Включение множественного выбора элементов компонента ListBox1
- a. ListBox2.MultiSelect:=false;
 - b. ListBox1.MultiSelect:=true;**
 - c. ListBox1.MultiSelect:=false;
 - d. ListBox1.Checked:=true;
12. Запись ListBox1.Selected[3]:=true означает что:
- a. Необходимо установить три дополнительных элемента
 - b. Необходимо удалить три лишних элемента
 - c. Выделен четвертый элемент в компоненте**
 - d. Выделен третий элемент в компоненте
13. Простейшая функция вывода на экран текстовых сообщений:
- a. ListBox1.Items.Add();
 - b. ShowMessage();**
 - c. Case...of;
 - d. CheckBox1.Checked:=true;
14. Полная очистка списка:
- a. ListBox1.Clear;**
 - b. ListBox1.Items.Add();
 - c. RadioGroup1.ItemIndex:=0;
15. Снятие «флажка» компонента CheckBox1
- a. ListBox1.Checked:=true;
 - b. CheckBox1.Checked:=false;**
 - c. CheckBox1.Checked:=true;
 - d. CheckBox1.Caption := 'Снять флажок';
16. Свойство Align=AllBottom:
- a. Компонент перемещается в центр формы
 - b. Компонент перемещается в верхнюю часть контейнера
 - c. Компонент перемещается в нижнюю часть контейнера и его ширина становится равной ширине контейнера**
17. Объект ComboBox - ...
- a. Многострочный редактор
 - b. Комбинированный список**
 - c. Функциональная кнопка
18. Установка в ComboBox1 первой строки массива Item:
- a. ItemIndex=0**
 - b. ItemIndex=-1
 - c. ItemIndex=1
 - d. ItemIndex=356
19. Объект RadioGroup - ...
- a. Команда «Авторadio»
 - b. Панель группы радиокнопок**
 - c. Метка
20. Дополнить предложение. Компонент-флажок, который используется для обозначения включения или отключения какой либо опции, называется...

- a. RadioGroup
- b. Функциональная кнопка
- c. **CheckBox**
- d. ListBox

Шкала оценки тестовых вопросов:

«2» – от 0 до 40%

«3» – от 41% до 60%

«4» – от 61% до 80%

«5» – от 81% до 100%

Задание 4. Практические работы

Перечень тем практических работ:

Практическая работа №17 «Условный оператор»

Практическая работа №18 «Оператор выхода из цикла break»

Практическая работа №19 «Оператор continue»

Практическая работа №20 «Метки»

Практическая работа №21 «Работа с клавиатурой»

Практическая работа №22 «Работа с экраном»

Практическая работа №23 «Возможности C++ в системном программировании»

Практическая работа №24 «Синхронизация потоков с использованием мониторов»

Практическая работа №25 «Синхронизация потоков с использованием семафоров»

Практическая работа №26 «Работа со структурами Дата-Время. Работа со структурами-строками»

Практическая работа №27 «Работа со структурами-окнами. Работа со структурами-массивами»

Практическая работа №28 «Использование классов в С++. Использование статических членов классов в С++»

Практическая работа №29 «Операции для работы с классами. Использование классов при работе с отладчиком»

Практическая работа №30 «Реализация иерархии классов. Наследование и полиморфизм. Обработка исключений на С++»

Практическая работа №31 «Использование шаблонных функций»

Практическая работа №32 «Разработка шаблонов стандартных структур данных»

Практическая работа №33 «Реализация иерархии объектов на основе механизмов наследования»

Практическая работа №34 «Пользовательский интерфейс прикладных программ»

Практическая работа №35 «Организация разработки прикладного программного обеспечения»

Шкала оценки практических работ:

оценка «5» ставится, если:

- учащийся самостоятельно выполнил все этапы решения задач на ЭВМ;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с ЭВМ в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но учащийся владеет основными навыками работы на ЭВМ, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что учащийся не владеет обязательными знаниями, умениями и навыками работы на ЭВМ или значительная часть работы выполнена не самостоятельно.

Задание 5. Рефераты

Перечень тем рефератов:

1. Указатели и массивы в языке C++
2. Функции в языке C++
3. Строки в языке C++
4. Файлы в языке C++
5. Динамические структуры данных в языке C++
6. Многомодульные программы на языке C++
7. Объединение.
8. Принципы и средства работы с файлами.
9. Выделение памяти для переменных.
10. Свободная память.
11. Использование директив препроцессора для создания гибких и мобильных программ.
12. Технология разработки многомодульных программ.

Шкала оценки рефератов:

Критерии и показатели, используемые при оценивании учебного реферата

| Критерии | Показатели |
|---|--|
| 1. Новизна реферированного текста Макс. - 20 баллов | <ul style="list-style-type: none">- актуальность проблемы и темы;- новизна и самостоятельность в постановке проблемы, в формулировании нового аспекта выбранной для анализа проблемы;- наличие авторской позиции, самостоятельность суждений. |
| 2. Степень раскрытия сущности проблемы Макс. - 30 баллов | <ul style="list-style-type: none">- соответствие плана теме реферата;- соответствие содержания теме и плану реферата;- полнота и глубина раскрытия основных понятий проблемы;- обоснованность способов и методов работы с материалом;- умение работать с литературой, систематизировать и структурировать материал;- умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы. |
| 3. Обоснованность выбора источников Макс. - 20 баллов | <ul style="list-style-type: none">- круг, полнота использования литературных источников по проблеме;- привлечение новейших работ по проблеме (журнальные публикации, материалы сборников научных трудов и т.д.). |

| | |
|---|---|
| 4. Соблюдение требований к оформлению Макс. - 15 баллов | <ul style="list-style-type: none"> - правильное оформление ссылок на используемую литературу; - грамотность и культура изложения; - владение терминологией и понятийным аппаратом проблемы; - соблюдение требований к объему реферата; - культура оформления: выделение абзацев. |
| 5. Грамотность Макс. - 15 баллов | <ul style="list-style-type: none"> - отсутствие орфографических и синтаксических ошибок, стилистических погрешностей; - отсутствие опечаток, сокращений слов, кроме общепринятых; - литературный стиль. |

Оценивание реферата

Реферат оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

- 86 – 100 баллов – «отлично»;
- 70 – 75 баллов – «хорошо»;
- 51 – 69 баллов – «удовлетворительно»;
- менее 51 балла – «неудовлетворительно».

Оценивание студента на экзамене

Оценка 5 **«отлично»** выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач по формированию общепрофессиональных компетенций.

Оценка 4 **«хорошо»** выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а так же имеет достаточно полное представление о значимости знаний по дисциплине.

Оценка 3 **«удовлетворительно»** выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой.

Оценка 2 **«неудовлетворительно»** выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине.

Раздел 5. Промежуточный контроль (в форме экзамена)

ФОС для промежуточной (семестровой) аттестации обучающихся по дисциплине предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения дисциплины в установленной учебным планом форме и позволяет определить качество усвоения изученного материала.

Итоговой формой контроля сформированности компетенций у студентов по дисциплине является экзамен.

ФОС промежуточной аттестации состоит из вопросов и задач к экзамену по дисциплине.

Оценивание студента на экзамене

Оценка 5 **«отлично»** выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач по формированию общепрофессиональных компетенций.

Оценка 4 **«хорошо»** выставляется студенту, если он грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а так же имеет достаточно полное представление о значимости знаний по дисциплине.

Оценка 3 **«удовлетворительно»** выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой.

Оценка 2 **«неудовлетворительно»** выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине.

Вопросы к билетам

1. Расскажите о технологии программирования
2. Цель модульного программирования. Основные характеристики модульного программирования
3. Методы разработки структуры программы
4. Функциональная спецификация
5. Условные операторы и операторы цикла
6. Прототип функции. Библиотечные файлы. Директива препроцессора `#include`.
7. Задачи и особенности прикладного программирования
8. Основные инструменты прикладного программиста
9. Выбор языка программирования
10. Принципы объектно-ориентированного анализа
11. Компиляция программы и сборка исполняемого модуля
12. Структура программы на языке C++. Примеры. Этапы создания исполняемой программы.
13. Состав языка C++. Константы и переменные C++.
14. Типы данных в C++.
15. Выражения. Знаки операций. Постфиксные и префиксные операции
16. Сводка операций: скобки, порядок вычислений, инкремент и декремент, преобразование типа.
17. Основные операторы C++ (присваивание, составные, выбора, циклов, перехода). Синтаксис, семантика, примеры.
18. Массивы (определение, инициализация, способы перебора).
19. Сортировка массивов (простой обмен, простое включение, простой выбор).
20. Указатели. Операции с указателями. Примеры.
21. Одномерные массивы и указатели. Примеры.

22. Многомерные массивы и указатели. Примеры.
23. Символьная информация и строки. Функции для работы со строками (библиотечный файл string.h).
24. Функции в C++. Рекурсия. Примеры.
25. Место языков ассемблера среди языков программирования.
26. Структура МП Intel 80x86: используемые регистры.
27. Структура МП Intel 80x86: операционное устройство и шинный интерфейс.
28. Размещение данных в памяти. Сегментация памяти.
29. Структура регистра флагов. Команды установки флагов.
30. Структура и форматы команд МП Intel 80x86. Команды пересылки данных.
31. Способы адресации в командах МП Intel 80x86.
32. Система команд МП: команды сложения и вычитания.
33. Команды умножения и деления чисел с ФТ.
34. Структура команд МП: базовая, индексная и косвенная адресации.
35. Логические команды обработки битов.
36. Команды сдвигов и их использование.
37. Команды передачи управления: безусловные переходы. Адресация в переходах.
38. Команды передачи управления: условные переходы.
39. Команды передачи управления: организация циклов.
40. Стек. Команды работы со стеком.
41. Элементарные конструкции языка ассемблера: алфавит, ключевые слова.
42. Элементарные конструкции языка ассемблера: числа, символьные данные.
43. Элементарные конструкции языка ассемблера: имена, метки.
44. Элементарные конструкции языка ассемблера: выражения и их использование.
45. Предложения языка ассемблера: комментарии.
46. Предложения языка ассемблера: команды.
47. Предложения языка ассемблера: директивы.
48. Структура файла ассемблер-программы. Директивы оформления программы.
49. Структура файла ассемблер-программы: односегментные и многосегментные файлы.
50. Использование прерываний в ассемблер-программах.
51. Операторы в командах языка ассемблера.
52. Блочная структура программы: правила описания и вызова процедур.
53. Блочная структура программы: расположение процедур в исходном файле.
54. Блочная структура программы: внутренние и внешние процедуры.
55. Способы передачи параметров между процедурой и вызывающей программой.

56. Передача параметров между процедурой и вызывающей программой. Проблема сохранения регистров.
57. Программные пакеты MASM и TASM: этапы обработки задания (подготовка исходного файла и его трансляция).
58. Программные пакеты MASM и TASM: этапы обработки задания (компоновка объектного модуля и отладка программы).
59. Программные пакеты MASM и TASM: общие функции и различия.
60. Модели памяти и их использование в TASM.
61. Макросы: макроопределения и их использование.
62. Макросы: использование параметров и комментариев.
63. Требования к программному проекту
64. Требования к оформлению программной документации
65. Составление эскизного проекта

Дополнительные (устные) вопросы

1. Дайте определение понятию «Программирование»
2. Дайте определение понятию «Модуль»
3. Дайте определение понятию «Модульное программирование»
4. Жизненный цикл программы.
5. Представление данных в IBM PC: целые числа.
6. Представление данных в IBM PC: двоично-десятичные числа.
7. Представление данных в IBM PC: алфавитно-цифровые данные.
8. Представление данных в IBM PC: вещественные данные.

Примерные задания для подготовки к экзамену:

1. Задание: составить программу вычисления X в зависимости от значения символа ch по формуле, соответствующей варианту. Значение символа ch вводится с клавиатуры. Для контроля за правильностью работы программы использовать модуль IO.ASM. Формулы для расчета:

$(8g)/(5m)$, если $ch > 'c'$

$X = d + g - 6$, если $ch = 'c'$

$3v - 4d + 100$, если $ch < 'c'$

2. Задание: составить программу вычисления X в зависимости от значения символа ch по формуле, соответствующей варианту. Значение символа ch вводится с клавиатуры. Для контроля за правильностью работы программы использовать модуль IO.ASM. Формулы для расчета:

$3 + vg - m$, если $ch > 't'$

$X = (10d)/(7m)$, если $ch = 't'$

$m - 7 + 3d$, если $ch < 't'$

3. Составить программу вычисления y по формуле: $y = 2m - 7g + 4 - d * v$

4. Составить программу вычисления y по формуле: $y = 2m + 8 - g * d + 5v$

5. Найти в DSEG: 1) 3-й по порядку нулевой байт; 2) 4-й по порядку код CR (0Dh); 3) 4-й байт из числа тех, которые ниже 20h; 4) 3-й по порядку код '\$'(24h);

6. Найти в DSEG: 1) байт, следующий за 3-м кодом ';' (3Bh); 2) 4-й байт из числа больших, чем 29h; 3) байт, следующий за 3-м отрицательным байтом;

7. Найти в DSEG: 1) байт, являющийся 4-м нечетным; 2) байт, следующий за 3-м кодом пробела (20h); 3) 3-й байт из числа тех, которые выше 10h;

8. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Если символ в нижнем регистре, перевести его в верхний регистр; если в верхнем – в нижний

9. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Вывести строку в обратном порядке

10. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Вывести строку, в закодированном виде, от каждого кода символа строки отнимается число 10.

11. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Удалить все символы в верхнем регистре;

12. Написать программу с использованием процедур, которая запрашивает строку (ввод с клавиатуры), и затем переводит все символы по следующему алгоритму: Найти позицию символа (вводится с клавиатуры) в строке и вывести позицию (и) в шестнадцатеричном виде.

13. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления у по формуле: $y=5v-d*m+6g+3$

14. Исходные данные:

- дата рождения студента в формате ДД-ММ- ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления у по формуле: $y=5d-g*v+7m-2$

15. Вычислить: $(A*B-C)/D$

16. Найдите ошибку в следующих командах:

MOV DS,@DATA

MOV AX,ES

MOV DS,AX

MOV CS,AX

MOV (ячейка памяти), (ячейка памяти)

17. Найдите ошибку в следующих командах:

XCNG AX,BX

XCHG CX,BL

XCHG ES,DS

XCHG DATA1,AX

18. Вычисление результата выполнения арифметического выражения, в котором некоторые числа постоянны, а другие переменные.

Формула вычислений: $X = (A * 2 + B * C) / (D - 3)$

19. Исходные данные:

- дата рождения студента в формате ДД-ММ-ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления y по формуле: $y = 2v + 6d - m * g + 3$

20. Исходные данные:

- дата рождения студента в формате ДД-ММ-ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления y по формуле: $y = g * m - 7d + v - 20$

21. Исходные данные:

- дата рождения студента в формате ДД-ММ-ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления y по формуле: $y = g * m - 4d + 8v - 7$

22. Исходные данные:

- дата рождения студента в формате ДД-ММ-ГГ - числа – d, m, g (байт)

- возраст студента (количество полных лет) – число v (байт)

Задание: Составить программу вычисления y по формуле: $y = 2 + m * d - 3g + 7v$

Примечание (расшифровка компетенций)

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

ПК 1.1 Выполнять разработку спецификаций отдельных компонент.

ПК 1.2 Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4 Выполнять тестирование программных модулей.

ПК 1.5 Осуществлять оптимизацию программного кода модуля.

ПК 1.6 Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

В результате освоения междисциплинарного курса обучающийся должен иметь **практический опыт:**

- разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования;
- разработки кода программного продукта на основе готовой спецификации на уровне модуля;
- использования инструментальных средств на этапе отладки программного продукта;
- проведения тестирования программного модуля по определенному сценарию;

уметь:

- осуществлять разработку кода программного модуля на современных языках программирования;
- создать программу по разработанному алгоритму как отдельный модуль;
- выполнять отладку и тестирование программы на уровне модуля;
- оформлять документацию на программные средства;
- использовать инструментальные средства для автоматизации оформления документации;

знать:

- основные этапы разработки программного обеспечения;
- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные принципы отладки и тестирования программных продуктов;
- методы и средства разработки технической документации.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

| № п.п. | Содержание изменения | Дата, номер протокола заседания педагогического совета |
|-----------|---|--|
| 1 | 2 | 3 |
| 1 | Внесены изменения в перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы. | решение от 27.08.2020 №7 |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |
| 11 | | |
| 12 | | |
| 13 | | |
| 14 | | |