**ГБПОУ «Дзержинский педагогический колледж»**

*Утверждаю*

*Зам. директора ГБОУ СПО ДПК*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Н.Л.Абрамова /*

*«\_\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_ г.*

**Фонд оценочных средств**

*для проведения промежуточнойи итоговой аттестации по\_\_ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем  
 09.02.03 Программирование в компьютерных системах*

**Форма проведения оценочной процедуры***\_\_дифференцированный зачет, экзамен*

**Дзержинск, 2014**

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности СПО *09.02.03 Программирование в компьютерных системах* программы *профессионального модуля ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем*

**Разработчики:**

\_ГБОУ СПО ДПК \_\_\_\_преподаватель\_\_\_Н.А.Руденко\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место работы) (занимаемая должность) (инициалы, фамилия)

\_ГБОУ СПО ДПК \_\_\_\_преподаватель\_\_\_Н.В.Семенова\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место работы) (занимаемая должность) (инициалы, фамилия)

**Эксперты от работодателя**[[1]](#footnote-1)**:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место работы) (занимаемая должность) (инициалы, фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(место работы) (занимаемая должность) (инициалы, фамилия)

|  |
| --- |
| Рассмотрено на заседании предметно-цикловой комиссии  Спец. Информатика  Протокол №\_\_\_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.  Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ |
| Одобрено Экспертным советом колледжа  Протокол №\_\_\_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_\_г. |

**Содержание**

1.Паспорт фонда оценочных средств для профессионального модуля

1.1. Результаты освоения программы профессионального модуля, подлежащие проверки

1.1.1 Вид профессиональной

1.1.2. Показатели сформированности общих и профессиональных компетенций

2. Комплект оценочных средств для оценки освоения междисциплинарых курсов 7

2.1. Область применения. Показатели оценки освоенных знаний и умений

2.2 Организация контроля и оценки

2.3 Контрольно- измерительные материалы для оценки освоенных знаний и умений

2.4 Пакет экзаменатора.

3. Комплект оценочных средств для оценки по учебной и производственной практике

3.1 Паспорт комплекта оценочных средств

3.2.Оценка по учебной и производственной практикам практике………………………………………………….

4. Комплект оценочных средств для организации промежуточного контроля и оценки освоения программы профессионального модуля в ходе экзамена (квалификационного)

5. Контрольно-измерительные материалы для текущего контроля

1. .**Паспорт фонда оценочных средств для профессионального модуля**
2. **1.1. Результаты освоения программы профессионального модуля, подлежащие проверки**
3. **1.1.1 Вид профессиональной**
4. Результатом освоения профессионального модуля является готовность студента к выполнению вида профессиональной деятельности «Разработка программных модулей программного обеспечения для компьютерных систем.
5. **1.1.2. Показатели сформированности общих и профессиональных компетенций**

В результате освоения программы профессионального модуля у студентов должны быть сформированы следующие компетенции.

# Таблица 1.

# Показатели оценки сформированности ПК

|  |  |  |
| --- | --- | --- |
| Результаты(освоенные профессиональные компетенции) | Основные показатели оценки результата | Формы и методы контроля и оценки |
| ПК 1.1. Выполнять разработку спецификаций отдельных компонент. | Разработка приложений производится в соответствии с основными этапами разработки программного обеспечения;Разработка приложений осуществляется с применением основных принципов технологии структурного и объектно-ориентированного программирования;Оформление документации проводится в соответствии с требованиями, предъявляемыми к оформлению документации на программные средства. | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике. |
| ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля. | При разработке кода применяются основные принципы технологии структурного и объектно-ориентированного программирования;Разработка кода программного модуля осуществляется на современных языках программирования;Осуществляется создание программы по разработанному алгоритму как отдельного модуля;Осуществляется разработка кода программного продукта на основе готовой спецификации на уровне модуля; | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике. |
| ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. | Отладка программных модулей осуществляется с применением основных принципов отладки и тестирования программных продуктов;Осуществляется использование инструментальных средств на этапе отладки программного продукта;Осуществляется проведение отладки и тестирования программы на уровне модуля; | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике.. |
| ПК 1.4. Выполнять тестирование программных модулей. | Осуществляется проведение тестирования программного модуля по определенному сценарию;Осуществляется выполнение отладки и тестирование программы на уровне модуля. | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике. |
| ПК 1.5. Осуществлять оптимизацию программного кода модуля. | Осуществляется проведение оптимизации программного кода модуля по определенному сценарию;Осуществляется выполнение отладки и тестирования программы на уровне модуля;Осуществляется использование инструментальных средств на этапе отладки программного продукта | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике. |
| ПК 1.6. Разрабатывать компоненты проектной и техническойДокументации с использованием графических языков спецификаций. | Осуществляется использование инструментальных средства для автоматизации оформления документации;Определение и использование методов и средств разработки технической документации осуществляется в соответствии с поставленными задачами | Экспертная оценка выполненных практических работ.  Квалификационный экзамен.  Экзамен по МДК.  Дифференцированный зачет. Зачеты по производственной и учебной практике. |

**Таблица 2.**

**Показатели оценки сформированностиОК**

|  |  |  |
| --- | --- | --- |
| **Результаты**  **(освоенные общие компетенции)** | **Основные показатели оценки результата** | **Формы и методы контроля и оценки** |
| ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес | демонстрация интереса к будущей профессии | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество | * обоснование выбора и применения методов и способов решения профессиональных задач в области разработки технологических процессов * демонстрацияэффективности и качества выполнения профессиональных задач; | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 3. Решать проблемы, оценивать риски и принимать решенияв нестандартных ситуациях. | * Демонстрация способности принимать решение в стандартных и нестандартных ситуациях и нести за них ответственность. | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 4. Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения задач, профессионального и личностного развития. | * поиск и использование необходимой информации для эффективного выполнения профессиональных задач, профессионального и личностного развития, включая электронные. | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 5. Использовать информационно-коммуникационные технологии для совершенствованияпрофессиональной деятельности. | * Демонстрация навыков использования информационно-коммуникационных технологий в профессиональной деятельности | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 6. Работать в коллективе икоманде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями. | * взаимодействие с обучающимися, преподавателямив ходе обучения | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 7. Ставить цели, мотивировать деятельность подчиненных, организовывать и контролировать их работу с принятием на себя ответственности зарезультат выполнения заданий. | * Проявление ответственности за работу подчиненных, результат выполнения заданий. | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации | * Планирование обучающимися повышения личностного и квалификационного уровня. | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 9. Быть готовым к смене технологий в профессиональной деятельности. | * Проявление интереса к инновациям в области профессиональной деятельности | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |
| ОК 10. Исполнять воинскую обязанность, в том числе с применением полученныхпрофессиональных знаний (для юношей). | * Демонстрация готовности к исполнению воинской обязанности | Экспертное наблюдение и оценка на практическихзанятиях при выполнении работ по учебной и производственной практике  Дифференцированный зачет по МДК.  Квалификационный экзамен. |

1. **Формы промежуточной аттестации по профессиональному модулю**

Обязательной формой аттестации по итогам освоения программы профессионального модуля является экзамен (квалификационный). Результатом этого экзамена является однозначное решение: «вид профессиональной деятельности освоен / не освоен».

Для составных элементов профессионального модуля предусмотрена промежуточная аттестация.

**Таблица 3.**

**Формы промежуточной аттестации при освоении профессионального модуля**

|  |  |
| --- | --- |
| **Элементы модуля, профессиональный модуль** | **Формы промежуточной аттестации** |
| МДК 01.01. Системное программирование | Дифференцированный зачет |
| МДК 01.02. Прикладное программирование | Экзамен |
| Учебная практика | зачет |
| Производственная практика | Дифференцированный зачет |
| **ПМ. 01** | **Экзамен квалификационный** |

1. **2. Комплект оценочных средств для оценки освоения междисциплинарых курсов**
2. **Комплект измерительных материалов для оценки МДК 01.01 Системное программирование**
3. **2.1. Область применения.**

Комплект контрольно-оценочных средств предназначен для проверки результатов освоения МДК 01.01 Системное программирование по специальности 09.02.03 Программирование в компьютерных системах

Комплект контрольно-оценочных средств позволяет оценивать освоение умений и усвоение знаний.

**Таблица 4.**

**Показатели оценки освоенных умений и знаний**

|  |  |  |
| --- | --- | --- |
| **Результаты (освоенные умения и усвоенные знания)** | **Основные показатели** | **Формы, методы контроля и оценки** |
| **Умения:** |  |  |
| осуществлять разработку кода программного модуля на современных языках программирования; | Разработку кода программного модуля осуществляет на современных языках программирования | Практические работы, лабораторные работы |
| создавать программу по разработанному алгоритму как отдельный модуль; | Осуществляет создание программы по разработанному алгоритму как отдельного модуля;  Осуществляет разработку кода программного продукта на основе готовой спецификации на уровне модуля; | Практические работы, лабораторные работы |
| выполнять отладку и тестирование программы на уровне модуля; | Производит отладку программы на уровне модуля в соответствии с требованиями к приложению;  Отладку программных модулей осуществляет с применением основных принципов отладки программных продуктов;  Тестирование программных модулей осуществляет с применением основных принципов тестирования программных продуктов;  Применяет инструментальные средства на этапах отладки и тестирования программного продукта | Практические работы, лабораторные работы |
| оформлять документацию на программные средства; | Документацию оформляет в соответствии с требованиями | Лабораторные работы |
| использовать инструментальные средства для автоматизации оформления документации; | Для оформления документации использует инструментальные средства. | Лабораторные работы |
| **знать:** |  |  |
| основные этапы разработки программного обеспечения; | Воспроизводитэтапы разработки программного обеспечения; | Дифференцированный зачет по МДК  Квалификационный экзамен |
| основные принципы технологии структурного и объектно-ориентированного программирования; | Называет принципы технологии структурного и объектно-ориентированного программирования; | Практические работы, лабораторные работы  Дифференцированный зачет по МДК  Квалификационный |
| основные принципы отладки и тестирования программных продуктов; | Воспроизводит принципы отладки и тестирования программных продуктов | Устный опрос, практические работы, лабораторные работы  Квалификационный экзамен |
| методы и средства разработки технической документации | Воспроизводит методы и средства разработки технической документации | Устный опрос, лабораторные работы  Квалификационный экзамен |

* 1. **Организация контроля и оценки**

Контроль освоения МДК 01.01 Системное программирование осуществляется в форме дифференцированного зачета.

Зачет проводится в виде выполнения электронных тестов на сайте test.dpk.su/moodle и демонстрации результатов проектной работы. При неудовлетворительном оценивании проекта, студент не допускается к выполнению теста.

В комплекте тестовых заданий используются следующие формы тестовых заданий: выбор одного или нескольких вариантов ответа из предложенного множества, задание на установление правильной последовательности, задание на соответствие.

Условием положительной аттестации на зачете является положительная оценка усвоения всех знаний и освоения всех умений по всем контролируемым показателям.

**Контрольно- измерительные материалы для оценки освоенных знаний и умений**

**Тестовые задания**

1. Поставьте этапы программирования в порядке их появления

1-объектный подход к программированию

2-структурный подход к программированию

3-компонентный подход и CASE-технологии

4-стихийное» программирование

1. Выберите объектно-ориентированные языки программирования:

* Ада
* Ассемблер
* С++
* Delphi
* Java
* SQL
* HTML
* Фортран

1. Какие утверждения относительно понятия «Операционная система» являются справедливыми?

* ОС для своей работы использует различные структуры данных
* ОС – это набор взаимодействующих подпрограмм
* ОС полностью реализуется на аппаратном уровне
* ОС организует взаимодействие прикладных программ с аппаратурой компьютера

1. Какие состояния определяют состояния потока?

* состояние файловой системы
* состояние операционной системы
* состояние процессора
* состояние программы

1. Как называется программа, которая занимается распределением квантов времени между потоками?

* инспектор потоков
* менеджер потоков
* обслуживатель потоков
* менеджер квантов

1. Что называют потоком?

* содержимое памяти, к которой поток имеет доступ во время выполнения
* исполняемое на компьютере приложение вместе со всеми ресурсам
* последовательность выполнения инструкций программы
* различные данные программы

7. Что  называют процессом?

* содержимое памяти, к которой поток имеет доступ во время выполнения
* различные данные программы
* исполняемое на компьютере приложение вместе со всеми ресурсами
* последовательность выполнения инструкций программы

8. Какие состояния потоков являются основными?

* состояние выполнения кода потока процессором
* состояния подвешивания и блокирования потока
* состояние готовности к выполнению
* состояние изменения кода потока

9.Какие потоки будут обслуживаться первыми?

* обслуженные потоки
* приостановленный потоки
* имеющие одинаковый приоритет
* имеющие наивысший приоритет

10.Выберите ресурсы, принадлежащие каждому потоку:

* набор регистров процессора
* виртуальное адресное пространство
* маркер доступа, содержащий информацию для системы безопасности
* код исполняемой функции
* стек для работы приложения
* таблица для хранения дескрипторов объектов ядра
* стек для работы операционной системы

11.Соотнесите название функций с их назначением

* возобновление исполнения потока-Resume Thread
* создание потоков-Create Thread
* задержание исполнения потока-Sleep
* завершение работы потока-ExitThread
* приостановление исполнения потока-SuspendThread
* создание процессов-CreateProcess

завершение работы процесса-ExitProcess

12. Задан оператор: printf(“%d”,a); Какого типа должна быть переменная а?

* float
* double
* char
* integer

13. Указатель – это

* имя переменной
* количество памяти, занимаемой переменной
* идентификатор
* адрес поля памяти

14. Команды MS DOS. Выберите команду, которая копирует файл fail1.txt, расположенный на диске С, в каталог FFF на диске D:

* copy С:\fail1.txt D:\FFF
* copy: fail1.txt D:\FFF
* copy D:\FFF to С:\fail1.txt
* copy fail1.txt + D:\FFF

15. Команды MS DOS.  Выберите команду, с помощью которой получается файл  f3.txt  путем слияния файлов  f1.txt  и  f2.txt, расположенных на диске D.

* copy\_D:\f1.txt+D:\f2.txt\_D:\f3.txt
* copy D:\f1.txt+D:\f2.txt+D:\f3.txt
* copy D:\f1.txt+D:\f2.txt D:\f3.txt
* copy D:\f1.txt D:\f2.txt D:\f3.txt

16. Команды MS DOS. Какая команда делает каталог lesson текущим?

* cd.. lesson
* cdlesson
* copyconlesson
* copylesson

17. Выберите правильно записанные имена файлов а ОС Windows:

* “colledge”.dat
* Gruppa?.txt
* Primer1.doc
* 1+2+3.txt

18. Какие вопросы являются основными при реализации файловых систем?

* физическая организация хранения данных на дисках
* способы определения местонахождения запрошенных данных на дисках
* организация основной памяти
* алгоритмы выполнения операций с файлами

19. Что описывает тип HANDLE?

* дескриптор объекта
* дескриптор динамического обмена данными
* дескриптор открытого файла
* дескриптор данных

20. Соотнесите название функций с их назначением

|  |  |
| --- | --- |
| удаление файлов | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| чтение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| запись в файл | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| перемещение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| копирование файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| замещение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |
| создание и открытие файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile |

21. Какое значение параметра dwAccess функции CreateFile должно быть установлено, чтобы в файл можно было записывать информацию?

* GENERIC\_WRITE
* GENERIC\_NEW
* GENERIC\_READ
* OPEN\_EXISTING

22. Какое значение параметра dwCreate функции CreateFile должно быть установлено, чтобы в файл можно было открыть файл, а если файл не существует, то функция создаст его?

* OPEN\_ ALWAYS
* CREATE \_NEW
* OPEN\_EXISTING
* CREATE\_ALWAYS

23. Какие значения может возвращать функция CreateFile?

* INVALID\_HANDLE\_VALUE
* дескриптор открытого файла
* lpFileSizeHigh
* hFile

24. Выберите функцию, которая создаёт каталог:

* MoveFile
* CreateDirectory
* RemoveDirectory
* CreateDirectoryEx

25. Что удаляет функция RemoveDirectory?

* пустой файл
* любой каталог
* любой файл
* пустой каталог

**Пакет экзаменатора**

**Условия выполнения задания**

Зачет проводится по подгруппам в количестве 10 человек, путем выполнения тестовых заданий на компьютере. К зачету допускаются студенты, которые выполнили все лабораторные работы и сдали необходимые отчеты по лабораторным работам. Выполнение зачета оценивается на основании отметки за тест. Ответы на вопросы теста предоставляются в электронном виде на сайте test.dpk.su/moodle.

Фонд заданий состоит из 25 вопросов по разделам курса: основные принципы технологии структурного и объектно-ориентированного программирования, системное программирование в операционной системе Windows, потоки и процессы в операционной системе Windows. Каждый студент отвечает на 20 вопросов, выбранных системой случайным образом. Вопросы оцениваются от 1 до 2 баллов. Вопросы с одним вариантом ответа оцениваются в 1 балл, остальные в 2 балла. Если студент отвечает правильно только на часть вопроса, то получает соответствующее количество баллов. Система автоматически их просчитывает. Максимальное количество баллов за весь тест – 20. Для положительного результата необходимо набрать больше 10,5 баллов. Это оценивается отметкой «3». Отметкой «4» оцениваются результаты от 15 баллов, отметкой «5» - от 19 баллов.

Время выполнения теста - 1 академический час (45 минут) без перерыва.

Оборудование: бумага, ручки, компьютеры с выходом в Интернет.

|  |  |  |
| --- | --- | --- |
| **Задания (номер)** | **Предмет(ы) оценивания** | **Критерии оценки: количество баллов за задание, правильный ответ: 1балл** |
| 1.Поставьте этапы программирования в порядке их появления  объектный подход к программированию  структурный подход к программированию  компонентный подход и CASE-технологии  «стихийное» программирование | Знание этапов программирования и порядка их появления. Знание специализированной терминологии | 3-объектный подход к программированию  2-структурный подход к программированию  4-компонентный подход и CASE-технологии  1-«стихийное» программирование  (2 балла) |
| 2.Выберите объектно-ориентированные языки программирования:   * Ада * Ассемблер * С++ * Delphi * Java * SQL * HTML * Фортран | Знание объектно-ориентированных языков программирования | С++(0,5 балла)  Delphi(0,5 балла)  Java(0,5 балла)  Ада (0,5 балла) |
| 3. Какие утверждения относительно понятия «Операционная система» являются справедливыми?   * ОС для своей работы использует различные структуры данных * ОС – это набор взаимодействующих подпрограмм * ОС полностью реализуется на аппаратном уровне * ОС организует взаимодействие прикладных программ с аппаратурой компьютера | Знание определения и назначения операционной системы. Знание специализированной терминологии | * ОС для своей работы использует различные структуры данных (0,5 балла) * ОС – это набор взаимодействующих подпрограмм (0,5 балла) * ОС полностью реализуется на аппаратном уровне (0,5 балла) * ОС организует взаимодействие прикладных программ с аппаратурой компьютера (0,5 балла) |
| 4.Какие состояния определяют состояния потока?   * состояние файловой системы * состояние операционной системы * состояние процессора * состояние программы | Знание состояний понятия потока в Windows. Знание специализированной терминологии | * состояние процессора (1.0 балл) * состояние программы (1.0 балл) |
| 5. Как называется программа, которая занимается распределением квантов времени между потоками?   * инспектор потоков * менеджер потоков * обслуживатель потоков * менеджер квантов | Знание специализированной терминологии. Умение создавать программу | менеджер потоков (1.0 балл) |
| 6. Что называют потоком?   * содержимое памяти, к которой поток имеет доступ во время выполнения * исполняемое на компьютере приложение вместе со всеми ресурсам * последовательность выполнения инструкций программы * различные данные программы | Знание понятие потока в Windows. Знание специализированной терминологии | последовательность выполнения инструкций программы (1.0 балл) |
| 7. Что  называют процессом?   * содержимое памяти, к которой поток имеет доступ во время выполнения * различные данные программы * исполняемое на компьютере приложение вместе со всеми ресурсами * последовательность выполнения инструкций программы | Знание понятие процесса в Windows. Знание специализированной терминологии | исполняемое на компьютере приложение вместе со всеми ресурсами (1.0 балл) |
| 8. Какие состояния потоков являются основными?   * состояние выполнения кода потока процессором * состояния подвешивания и блокирования потока * состояние готовности к выполнению * состояние изменения кода потока | Знание состояний понятия процесса в Windows. Знание специализированной терминологии | * состояние выполнения кода потока процессором (0.66 балла) * состояния подвешивания и блокирования потока (0.66 балла) * состояние готовности к выполнению (0.66 балла) |
| 9.Какие потоки будут обслуживаться первыми?   * обслуженные потоки * приостановленный потоки * имеющие одинаковый приоритет * имеющие наивысший приоритет | Знание работы потоков в Windows. Умение разрабатывать приложения | имеющие наивысший приоритет (1.0 балл) |
| 10.Выберите ресурсы, принадлежащие каждому потоку:   * набор регистров процессора * виртуальное адресное пространство * маркер доступа, содержащий информацию для системы безопасности * код исполняемой функции * стек для работы приложения * таблица для хранения дескрипторов объектов ядра * стек для работы операционной системы | Знания для работы с потоками в Windows. Знание специализированной терминологии. | * набор регистров процессора (0.4 балла) * маркер доступа, содержащий информацию для системы безопасности (0.4 балла) * код исполняемой функции (0.4 балла) * стек для работы приложения (0.4 балла) * стек для работы операционной системы (0.4 балла) |
| 11.Соотнесите название функций с их назначением   * возобновление исполнения потока-Resume Thread * создание потоков-Create Thread * задержание исполнения потока-Sleep * завершение работы потока-ExitThread * приостановление исполнения потока-SuspendThread * создание процессов-CreateProcess * завершение работы процесса-ExitProcess | Знание специализированной терминологии. Умение разрабатывать приложения с использованием системных функций Windows | * возобновление исполнения потока-ResumeThread (0.28 балла) * создание потоков-CreateThread (0.28 балла) * задержание исполнения потока-Sleep (0.28 балла) * завершение работы потока-ExitThread (0.28 балла) * приостановление исполнения потока-SuspendThread (0.28 балла) * создание процессов-CreateProcess (0.28 балла)   завершение работы процесса-ExitProcess (0.28 балла) |
| 12. Задан оператор: printf(“%d”,a); Какого типа должна быть переменная а?   * float * double * char * integer | Знание специализированной терминологии. Умение разрабатывать приложенияна языке С++ | Integer (1 балл) |
| 13. Указатель – это   * имя переменной * количество памяти, занимаемой переменной * идентификатор * адрес поля памяти | Знание специализированной терминологии. Знание понятия указатель. | адрес поля памяти (1 балл) |
| 14. Команды MS DOS. Выберите команду, которая копирует файл fail1.txt, расположенный на диске С, в каталог FFF на диске D:   * copy С:\fail1.txt D:\FFF * copy: fail1.txt D:\FFF * copy D:\FFF to С:\fail1.txt * copy fail1.txt + D:\FFF | Знание специализированной терминологии. Умение работать с командной строкой Windows | copy С:\fail1.txt D:\FFF(1 балл) |
| 15. Команды MS DOS.  Выберите команду, с помощью которой получается файл  f3.txt  путем слияния файлов  f1.txt  и  f2.txt, расположенных на диске D.   * copy\_D:\f1.txt+D:\f2.txt\_D:\f3.txt * copy D:\f1.txt+D:\f2.txt+D:\f3.txt * copy D:\f1.txt+D:\f2.txt D:\f3.txt * copy D:\f1.txt D:\f2.txt D:\f3.txt | Знание специализированной терминологии. Умение работать с командной строкой Windows | copyD:\f1.txt+D:\f2.txtD:\f3.txt (1 балл) |
| 16. Команды MS DOS. Какая команда делает каталог lesson текущим?   * cd.. lesson * cdlesson * copyconlesson * copylesson | Знание специализированной терминологии. Умение работать с командной строкой Windows | cdlesson(1 балл) |
| 17. Выберите правильно записанные имена файлов а ОС Windows:   * “colledge”.dat * Gruppa?.txt * Primer1.doc * 1+2+3.txt | Знание специализированной терминологии. Умение создавать программы для работы с файловой системой Windows | Primer1.doc (1 балл)  1+2+3.txt (1 балл) |
| 18. Какие вопросы являются основными при реализации файловых систем?   * физическая организация хранения данных на дисках * способы определения местонахождения запрошенных данных на дисках * организация основной памяти * алгоритмы выполнения операций с файлами | Знание специализированной терминологии. Умение создавать программы для работы с файловой системой Windows | физическая организация хранения данных на дисках (0,66 балла)  способы определения местонахождения запрошенных данных на дисках (0,66 балла)  алгоритмы выполнения операций с файлами (0,66 балла) |
| 19. Что описывает тип HANDLE?   * дескриптор объекта * дескриптор динамического обмена данными * дескриптор открытого файла * дескриптор данных | Знание специализированной терминологии. Знание объектов и дескрипторов в Windows | дескриптор объекта (1 балл) |
| 20. Соотнесите название функций с их назначением   |  |  | | --- | --- | | удаление файлов | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | чтение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | запись в файл | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | перемещение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | копирование файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | замещение файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | создание и открытие файла | MoveFile  ReadFile  CloseFile  WriteFile  DeleteFile  ReplaceFile  CopyFile  CreateFile | | Знание специализированной терминологии. Знание функций для работы с файловой системой Windows | удаление файлов - DeleteFile  чтение файла - ReadFile  запись в файл - WriteFile  перемещение файла - MoveFile  копирование файла - CopyFile  замещение файла- ReplaceFile  создание и открытие файла–CreateFile  (2 балла) |
| 21. Какое значение параметра dwAccess функции CreateFile должно быть установлено, чтобы в файл можно было записывать информацию?   * GENERIC\_WRITE * GENERIC\_NEW * GENERIC\_READ * OPEN\_EXISTING | Знание специализированной терминологии. Умение использовать функций и соответствующих параметров. | GENERIC\_WRITE (1 балл) |
| 22. Какое значение параметра dwCreate функции CreateFileдолжно быть установлено, чтобы в файл можно было открыть файл, а если файл не существует, то функция создаст его?   * OPEN\_ ALWAYS * CREATE \_NEW * OPEN\_EXISTING * CREATE\_ALWAYS | Знание специализированной терминологии. Умение использовать функций и соответствующих параметров. | OPEN\_ ALWAYS (1 балл) |
| 23. Какие значения может возвращать функция CreateFile?   * INVALID\_HANDLE\_VALUE * дескриптор открытого файла * lpFileSizeHigh * hFile | Знание специализированной терминологии. Знание функций для работы с файловой системой Windows | INVALID\_HANDLE\_VALUE (1 балл)  дескриптор открытого файла(1 балл) |
| 24. Выберите функцию, которая создаёт каталог:   * MoveFile * CreateDirectory * RemoveDirectory * CreateDirectoryEx | Знание специализированной терминологии. Знание функций для работы с файловой системой Windows | CreateDirectory (1 балл) |
| 25. Что удаляет функция RemoveDirectory?   * пустой файл * любой каталог * любой файл * пустой каталог | Знание специализированной терминологии. Знание функций для работы с файловой системой Windows | пустой каталог (1 балл) |

**Комплект измерительных материалов для оценки МДК 01.02 Прикладное программирование**

**1. Область применения**

Комплект контрольно-оценочных средств предназначен для проверки результатов освоения МДК 01.02 Прикладное программирование *09.02.03Программирование в компьютерных системах*

Комплект контрольно-оценочных средств позволяет оценивать освоение умений и усвоение знаний.

**Таблица 5.**

## Показатели оценки освоенных знаний и умений

|  |  |  |
| --- | --- | --- |
| **Результаты освоенния (объекты оценивания)** | **Основные показатели оценки результата** | **Средства проверки** |
| **Умения:** |  |  |
| осуществлять разработку кода программного модуля на современных языках программирования | Разработку кода программного модуля осуществляет на современных языках программирования | Экспертная оценка выполненных практических работ.  Экзамен по МДК. Выполнение заданий в ходе производственной и учебной практики. |
| создавать программу по разработанному алгоритму как отдельный модуль | Осуществляет создание программы по разработанному алгоритму как отдельного модуля;  Осуществляет разработку кода программного продукта на основе готовой спецификации на уровне модуля; | Экспертная оценка выполненных практических работ.  Экзамен по МДК. .Выполнение заданий в ходе производственной и учебной практики. |
| выполнять отладку и тестирование программы на уровне модуля | Производит отладку программы на уровне модуля в соответствии с требованиями к приложению;  Отладку программных модулей осуществляет с применением основных принципов отладки программных продуктов;  Тестирование программных модулей осуществляет с применением основных принципов тестирования программных продуктов;  Применяет инструментальные средства на этапах отладки и тестирования программного продукта | Экспертная оценка выполненных практических работ.  Экзамен по МДК. Выполнение заданий в ходе производственной и учебной практики. |
| оформлять документацию на программные средства | Оформляет документацию в соответствии с требованиями | Экспертная оценка выполненных практических работ.  Экзамен по МДК. Выполнение заданий в ходе производственной и учебной практики. |
| использовать инструментальные средства для автоматизации оформления документации | Для оформления документации использует инструментальные средства. | Экспертная оценка выполненных практических работ.  Экзамен по МДК. Выполнение заданий в ходе производственной и учебной практики. |
| **Знания:** |  |  |
| основные этапы разработки программного обеспечения | Воспроизводитэтапы разработки программного обеспечения; | Экзамен по МДК. Тестирование |
| основные принципы технологии структурного и объектно-ориентированного программирования | Называет принципы технологии структурного и объектно-ориентированного программирования; | Экзамен по МДК.  Тестирование |
| основные принципы отладки и тестирования программных продуктов | Воспроизводит принципы отладки и тестирования программных продуктов | Экзамен по МДК.  Тестирование |
| методы и средства разработки технической документации | Воспроизводит методы и средства разработки технической документации | Экзамен по МДК.  Тестирование |

**Организация контроля и оценки**

Промежуточный контроль освоения МДК 01.02 Прикладное программирование осуществляется в форме экзамена.

Экзамен проводится в виде выполнения электронных тестов на сайте test.dpk.su/moodle, а так же практических заданий на компьютере. В комплекте тестовых заданий используются следующие формы тестовых заданий: выбор одного варианта ответа из предложенного множества, задание на соответствие, задания с кратким ответом, задания с ответом верно/неверно. Фонд заданий состоит из вопросов по разделам курса. На каждый проверяемый учебный элемент предусмотрено не менее одного тестового задания.

**Контрольно- измерительные материалы для оценки освоенных знаний и умений**

**Тестовые задания**

**Программное обеспечение**

1Найдите соответствие  между характеристиками программного модуля и их значением

|  |  |
| --- | --- |
| размер модуля  прочность модуля  сцепление модуля  рутинность модуля  связность модулей | измеряется числом, содержащихся в нем операторов  мера его внутренних связей  мера его зависимости по способу передачи данных от других модулей  зависимость модуля от предыстории обращений к нему  мера прочности соединения функциональных и информационных объектов внутри модуля |

2Выберите классы программного обеспечения

* системное ПО
* прикладное ПО
* инструментарий технологии программирования
* общее ПО

3 Модульным программированием называют

* метод создания программ, когда программу разрабатывают несколько человек
* метод создания программ, когда программу разрабатывают по частям
* метод создания программ, когда готовую программу разбивают на части

4 Выберите программное обеспечение, относящееся к прикладному

* автоматизированные системы управления
* издательские системы
* графические системы
* авторские системы
* антивирусные программы
* операционная система
* системы автоматизированного проектирования
* средства создания программ для компьютера

5Выберите основные процессы жизненного цикла в соответствии с ISO/IEC 12207

* процесс документирования
* процесс приобретения
* процесс верификации
* процесс поставки
* процесс разработки
* процесс сопровождения
* процесс эксплуатации

**Типы даных**

6Соотнесите имя типа данных с типом

|  |  |
| --- | --- |
| long  long double  ansiChar  boolean  ansiString | целый тип  вещественный тип  символьный тип  логический тип  строковый тип |

7 Для определения переменной нужно указать

* адрес переменной
* тип переменной
* значение переменной
* имя переменной

8 Какая последовательность символов является именем переменной

* F?
* Klava3
* 1Km
* AbCd
* \_Tri\_2
* Mel.1
* A\_c
* void

9 Выберите те выражения, которые могут быть константами

* Enum
* 'E'
* 'ddddd'
* Ox7
* 250
* 359E-4
* truefalse
* 5,25

10Какой тип данных имеет диапазон от -32768 до 32767?

* shortint
* char
* long
* int

11 Пусть имеется множество типа перечисления. enumd{two=2, three,five, six=6 };

Чему равно значение переменной five?

* 2
* 3
* 4
* 5
* 6

12 Выберите тип данных, у которого диапазон значений от 0 до 255

* unsignedshort
* unsignedint
* unsignedlong
* unsignedchar

13 Выберите только те типы данных, которые относятся к вещественным

* int
* float
* char
* long
* longdouble
* double

14 Какой тип данных имеет диапазон от -128 до 127?

**Тестирование**

15Выберите уровни тестирования

* системное
* прикладное
* модульное
* отладочное
* интеграционное
* приемочное
* выходное

16В ходе какого тестирования проверяют связи между модулями, их совместимость и функциональность?

* выходного
* приемочного
* модульного
* интеграционного
* прикладного
* отладочного
* системного

17Какой вид тестирования провидится организацией, отвечающей за инсталляцию, сопровождение программной системы и обучение конечного пользователя?

* выходное
* модульное
* интеграционное
* приемочное
* отладочное
* системное

18 Что называется отладкой?

* процесс обнаружения ошибки
* способ улучшения качества ПО
* процесс подсчета количества ошибок
* процесс обнаружения причин возникновения ошибки и ее последующего исправления

19 Техническая документация используется для определения и описания

* алгоритмов
* структур данных
* описание рабочей среды
* руководства пользователей
* API

20Какие программы относятся к генераторам документации?

* Java
* PHPDoc
* NDoc
* Doxygen
* Aris
* Javadoc

21К какому типу ошибок  относятся ошибки в логике и алгоритмах программы?

* Невоспроизводимые ошибки
* Ошибки многопользовательского доступа
* Ошибки во время выполнения
* Ошибки логики взаимосвязанных CGI-программ
* Ошибки синтаксиса языка
* Ошибки инструментария и других компонентов системы

**С++ Builder**

22Выберите объекты у которых есть следующие свойства

|  |  |
| --- | --- |
| Name | Edit  Form  Timer  Label  Всеобъекты |
| Text | Edit  Form  Timer  Label  Всеобъекты |
| Canvas | Edit  Form  Timer  Label  Всеобъекты |
| WordWrap | Edit  Form  Timer  Label  Всеобъекты |
| Top | Edit  Form  Timer  Label  Всеобъекты |

23 Какой объект можно использовать для того, чтобы пользователь мог указать свое решение да/нет?

* RadioButton
* CheckBox
* Label
* Timer

24 Какой объект призапуске приложения не отображается на форме?

* Timer
* Edit
* Shape
* Image

25 Выберите функцию преобразования, значением которой является дробное число, изображением которого является строка s

* StrToFloat(s)
* StrToInt(s)
* FloatToStr(s)
* IntToStr(s)

26 Какое событие происходит при создании объекта

* OnEnter
* OnPaint
* OnClick
* OnCreate

27 Какое событие происходит при нажатии кнопки мыши

* OnMouseMove
* OnMouseDown
* OnMouseUp
* OnMousePress

28 Указатель – это

* адрес участка памяти, выделенный для переменной
* имя переменной
* операция определения значения переменной
* переменная, содержащая адрес другой переменной

29 Используя какое свойство, можно заполнить комбинированный список ComboBox?

* Text
* Count
* Items
* ItemIndex

C:\Documents and Settings\Наталья\Рабочий стол\кнопка с изображением.JPG

30Впишите название объекта, изображенного на рисунке

31 Что будет выведено в объект Label1 после выполнения кода программы, если для значения чисел берутся значения по умолчанию

***void \_\_fastcall TForm1::Button1Click(TObject \*Sender)***

***{ inta,b;***

***a=StrToInt(InputBox("Слагаемые","Введитепервоечисло", "3"));***

***b=StrToInt(InputBox("Слагаемые","Введите первое число", "4"));***

***Label1->Caption=IntToStr(++a+b);***

***}***

32 Верно ли утверждение, что компонент StringGrid нельзя использовать в задачах на одномерные массивы?

* Верно
* Неверно

**Практические задания:**

1. Создайте приложение, которое считает количество положительных элементов двумерного массива, стоящих на главной и побочной диагоналях. Массив заполнить случайным образом. Размерность матрицы – четное число.
2. Создайте приложение, которое вычисляет стоимость поездки на автомобиле, если известно расстояние (км), цена на бензин (рубль/литр) и потребление бензина (литров на 100 км). Должна предусматриваться возможность выбора подсчета стоимости поездки туда или туда и обратно (использовать один флажок).
3. Создайте приложение, которое рисует на форме 3 ряда по 5 квадратов с зеленым контуром и желтой заливкой (между квадратами в ряду и между рядами должно быть расстояние).
4. Создайте приложение, которое выполняет арифметические действия над двумя числами. В программе используйте меню следующей структуры:

Файл Действия О программе

Выход Сложение

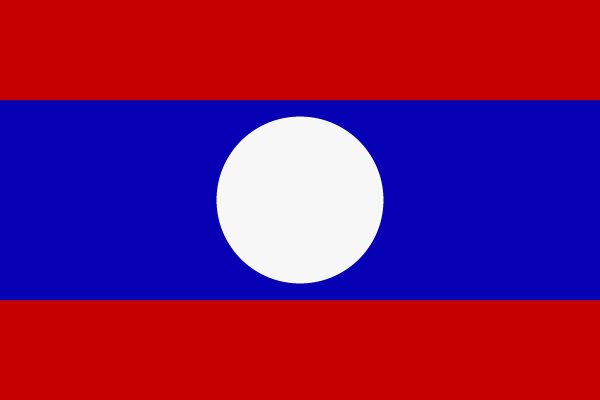
Вычитание

Умножение

Деление

Возвести в степень

1. Создайте приложение, которое определяет минимальный положительный элемент двумерного массива, заполненного с клавиатуры.
2. Создайте приложение, которое на поверхности формы рисует флаг Лаоса. Подпишите рисунок с помощью функции:



Белый

Синий

Красный

Красный

1. Создайте приложение, в котором пользователь вводит 5 чисел с помощью диалоговых окон и находит максимальное из них. Результат вывести через диалоговое окно.
2. Создайте приложение, которое считает сумму элементов, кратных 3 одномерного массива. Массив и его размерность вводятся с клавиатуры.
3. Создайте приложение, которое позволяет менять параметры шрифта (видимость, курсив и полужирный). Параметры шрифта можно выбрать с помощью флажков, а результаты изменения должны отображаться в метке.
4. Напишите программу для расчета сопротивления резисторов, соединенных параллельно или последовательно. Выбор соединения должен осуществляется с помощью переключателей. Сопротивление для последовательного соединения вычисляется по формуле R=R1+R2; а для параллельного R=(R1\*R2)/(R1+R2).
5. Практическое ***задание.*** Создайте приложение, которое вычисляет площадь кольца, если дан радиус внешней и радиус внутренней окружностей. Площадь кольца вычисляется по формуле: πr2
6. Создайте приложение, которое выводит самое длинное слово в символьном массиве. Массив и его размерность вводятся с клавиатуры.

**Организация экзамена**

Промежуточный контроль освоения профессионального модуля осуществляется в форме экзамена.

Экзамен проводиться в виде теста и практического задания. Вопросы в тестах охватывают темы курса: «Основные понятия прикладного программного обеспечения», «Тестирование и отладка программ»

Практический вопрос представляет собой задачу, которую необходимо решить в среде программирования.

Условием положительной аттестации на экзамене является положительная оценка усвоения всех знаний и освоения всех умений по всем контролируемым показателям.

**Пакет экзаменатора**

**Условия выполнения задания**

Экзамен проводится по подгруппам в количестве 10 человек, путем выполнения электронных тестов на сайте test.dpk.su/moodle, а так же практических заданий на компьютере. Каждому студенты предлагается ответить на 20 вопросов, выбранные системой случайным образом.

К выполнению практической работы допускаются студенты, получившие за тест выше 70%.

За выполнение теста студент получает 2 балл. За правильно спроектированное, разработанное приложение, которое протестировано и отлажено получает 2 балла. Преподаватель может оценить практическое задание, если студент допустил неточности при проектировании приложения, при тестировании уточнены на все возможные варианты использования приложения, не все необходимые функции приложения, заданные в условии задачи, работают в программе.

Максимальное количество баллов за ответ –4. Оценка «3» ставиться, если студент набрал 2 балл, «4» - 3 балла, «5» - 4 балла.

Время выполнения задания - 1 академический час (45 минут) без перерыва.

Оборудование: бумага, ручки, компьютеры с программойС++ Builder

|  |  |  |  |
| --- | --- | --- | --- |
| **Задания (номер)** | **Предмет(ы) оценивания** | | **Критерии оценки: количество баллов за задание** |
| 1 Найдите соответствие  между характеристиками программного модуля и их значением  **размер модуля-** измеряется числом, содержащихся в нем операторов  **прочность модуля-** мера его внутренних связей  **сцепление модуля-** мера его зависимости по способу передачи данных от других модулей  **рутинность модуля-** зависимость модуля от предыстории обращений к нему  **связность модулей-** мера прочности соединения функциональных и информационных объектов внутри модуля | Знание понятия модуля и его характеристик.  Знание специализированной терминологии | | **размер модуля-** измеряется числом, содержащихся в нем операторов  **прочность модуля-** мера его внутренних связей  **сцепление модуля-** мера его зависимости по способу передачи данных от других модулей  **рутинность модуля-** зависимость модуля от предыстории обращений к нему  **связность модулей-** мера прочности соединения функциональных и информационных объектов внутри модуля (2.0 балла) |
| 2Выберите классы программного обеспечения   * системное ПО * прикладное ПО * инструментарий технологии программирования * общее ПО | Знание специализированной терминологии. Знание понятия виды программное обеспечение. | | * системное ПО(0.33 балла) * прикладное ПО(0.33 балла) * инструментарий технологии программирования(0.33 балла) |
| 3 Модульным программированием называют   * метод создания программ, когда программу разрабатывают несколько человек * метод создания программ, когда программу разрабатывают по частям * метод создания программ, когда готовую программу разбивают на части | Знание специализированной терминологии. Знание понятия модульного программирования. | | метод создания программ, когда программу разрабатывают по частям(1.0 балл) |
| 4 Выберите программное обеспечение, относящееся к прикладному   * автоматизированные системы управления * издательские системы * графические системы * авторские системы * антивирусные программы * операционная система * системыавтоматизированного проектирования * средства создания программ для компьютера | Знание специализированной терминологии. Знание понятия прикладного программное обеспечение. | | * авторские системы (0.4 балла) * мыавтоматизированного проектирования (0.4 балла) * автоматизированные системы управления (0.4 балла) * издательские системы (0.4 балла) * графические системы (0.4 балла) |
| 5Выберите основные процессы жизненного цикла в соответствии с ISO/IEC 12207   * процесс документирования * процесс приобретения * процесс верификации * процесс поставки * процесс разработки * процесс сопровождения * процесс эксплуатации | Знание специализированной терминологии. Знание стандартов по программному обеспечению. | | * процесс поставки (0.4 балла) * процесс разработки (0.4 балла) * процесс сопровождения (0.4 балла) * процесс приобретения (0.4 балла) * процесс эксплуатации (0.4 балла) |
| 6Соотнесите имя типа данных с типом   |  |  | | --- | --- | | long  long double  ansiChar  boolean  ansiString | строковый тип  целый тип  символьный тип  вещественный тип  логический тип | | Знание типов данных. Знание специализированной терминологии. Умение разработки программного кода. | | |  |  | | --- | --- | | long  longdouble  ansiChar  boolean  ansiString | целый тип  вещественный тип  символьный тип  логический тип  строковый тип |   (2.0 балла) |
| 7 Для определения переменной нужно указать   * адрес переменной * тип переменной * значение переменной * имя переменной | Знание специализированной терминологии. Знание понятия переменной, правила ее описания. | | * тип переменной (0.5 балла) * имя переменной (0.5 балла) |
| 8 Какая последовательность символов является именем переменной   * F? * Klava3 * 1Km * AbCd * \_Tri\_2 * Mel.1 * A\_c * void | Знание специализированной терминологии. Знание понятия переменной, правила записи имени переменной. Умение разработки программного кода. | | * Klava3 (0.5 балла) * AbCd (0.5 балла) * A\_c (0.5 балла) * \_Tri\_2 (0.5 балла) |
| 9 Выберите те выражения, которые могут быть константами   * Enum * 'E' * 'ddddd' * Ox7 * 250 * 359E-4 * truefalse * 5,25 | Знание специализированной терминологии. Знание понятия константы, правила записи имени константы. Умение разработки программного кода. | | * 250 (0.5 балла) * 359E-4 (0.5 балла) * Ox7 (0.5 балла) * 'E' (0.5 балла) |
| 10Какой тип данных имеет диапазон от -32768 до 32767?   * shortint * char * long * int | Знание типов данных. Знание специализированной терминологии. Умение разработки программного кода. | | shortint (1.0 балл) |
| 11 Пусть имеется множество типа перечисления. enumd{two=2, three,five, six=6 };  Чему равно значение переменной five?   * 2 * 3 * 4 * 5 * 6 | Знание перечисляемого типа данных. Знание правил записи имен переменных. Знание специализированной терминологии | | 4 (1.0 балл) |
| 12 Выберите тип данных, у которого диапазон значений от 0 до 255   * unsignedshort * unsignedint * unsignedlong * unsignedchar | Знание типов данных, диапазона типов данных. Умение разработки программного кода. Знание специализированной терминологии | | unsignedchar (1.0 балл) |
| 13 Выберите только те типы данных, которые относятся к вещественным   * int * float * char * long * longdouble * double | Знание типов данных. Умение разработки программного кода. Знание специализированной терминологии | | * longdouble * float * double   (0.6 балла) |
| 14 Какой тип данных имеет диапазон от -128 до 127? | Знание типов данных. Знание специализированной терминологии. Умение разработки программного кода. | | Char (1.0 балл) |
| 15Выберите уровни тестирования   * системное * прикладное * модульное * отладочное * интеграционное * приемочное * выходное | Знание понятия тестирования программного модуля, уровней тестирования. Знание специализированной терминологии. | | * интеграционное (0.4 балла) * приемочное (0.4 балла) * системное (0.4 балла) * прикладное (0.4 балла) * выходное (0.4 балла) |
| 16В ходе какого тестирования проверяют связи между модулями, их совместимость и функциональность?   * выходного * приемочного * модульного * интеграционного * прикладного * отладочного * системного | Знание понятия тестирования программного модуля. Знание специализированной терминологии. | | интеграционного(1.0 балл) |
| 17Какой вид тестирования провидится организацией, отвечающей за инсталляцию, сопровождение программной системы и обучение конечного пользователя?   * выходное * модульное * интеграционное * приемочное * отладочное * системное | Знание понятия тестирования программного обеспечения, виды тестирования. Знание специализированной терминологии. | | Приемочное (1.0 балл) |
| 18 Что называется отладкой?   * процесс обнаружения ошибки * способ улучшения качества ПО * процесс подсчета количества ошибок * процесс обнаружения причин возникновения ошибки и ее последующего исправления | Знание понятия отладки программного обеспечения. Знание специализированной терминологии | | процесс обнаружения причин возникновения ошибки и ее последующего исправления (1.0 балл) |
| 19 Техническая документация используется для определения и описания   * алгоритмов * структур данных * описание рабочей среды * руководства пользователей * API | Знание технической документации на программное обеспечение. Знание специализированной терминологии | | * Алгоритмов (0.33 балла) * структур данных (0.33 балла) * API (0.33 балла) |
| 20Какие программы относятся к генераторам документации?   * Java * PHPDoc * NDoc * Doxygen * Aris * Javadoc | Знание программ для разработки технической документации на программное обеспечение. Знание специализированной терминологии | | * NDoc (0.5 балла) * Doxygen (0.5 балла) * PHPDoc (0.5 балла) * Javadoc (0.5 балла) |
| 21К какому типу ошибок  относятся ошибки в логике и алгоритмах программы?   * Невоспроизводимые ошибки * Ошибки многопользовательского доступа * Ошибки во время выполнения * Ошибки логики взаимосвязанных CGI-программ * Ошибки синтаксиса языка * Ошибки инструментария и других компонентов системы | Знание понятия отладки программного обеспечения, типов ошибок, допускаемых при разработки кода программного обеспечения. Знание специализированной терминологии | | Ошибки во время выполнения (1.0 балл) |
| 22Выберите объекты у которых есть следующие свойства   |  |  | | --- | --- | | Name | Edit  Form  Timer  Label  Всеобъекты | | Text | Edit  Form  Timer  Label  Всеобъекты | | Canvas | Edit  Form  Timer  Label  Всеобъекты | | WordWrap | Edit  Form  Timer  Label  Всеобъекты | | Top | Edit  Form  Timer  Label  Всеобъекты | | Знание объектов и их свойств. Умение выделение спецификаций отдельных компонент. Знание специализированной терминологии | | * Name- Всеобъекты (0.4 балла) * Text- Edit (0.4 балла) * Canvas- Form (0.4 балла) * Top- Всеобъекты (0.4 балла) * WordWrap- Label (0.4 балла) |
| 23 Какой объект можно использовать для того, чтобы пользователь мог указать свое решение да/нет?   * RadioButton * CheckBox * Label * Timer | Знание объектов и их свойств. Назначение объектов. Знание специализированной терминологии | | CheckBox (1.0 балл) |
| 24 Какой объект призапуске приложения не отображается на форме?   * Timer * Edit * Shape * Image | Знание объектов и их свойств. Знание специализированной терминологии | | Timer (1.0 балл) |
| 25 Выберите функцию преобразования, значением которой является дробное число, изображением которого является строка s   * StrToFloat(s) * StrToInt(s) * FloatToStr(s) * IntToStr(s) | Знание функций преобразования. Умение правильного выбора функции при написании программного кода. | | StrToFloat(s) (1.0 балл) |
| 26 Какое событие происходит при создании объекта   * OnEnter * OnPaint * OnClick * OnCreate | Знание основных событий. Умение использования соответствующего события при разработки приложения. | | OnCreate( 1.0 балл) |
| 27 Какое событие происходит при нажатии кнопки мыши   * OnMouseMove * OnMouseDown * OnMouseUp * OnMousePress | Знание основных событий. Умение использования соответствующего события при разработки приложения | | OnMouseDown (1.0 балл) |
| 28 Указатель – это   * адрес участка памяти, выделенный для переменной * имя переменной * операция определения значения переменной * переменная, содержащая адрес другой переменной |  | | переменная, содержащая адрес другой переменной (1.0 балл) |
| 29 Используя какое свойство, можно заполнить комбинированный список ComboBox?   * Text * Count * Items * ItemIndex | Знание объектов и их свойств. Знание специализированной терминологии. Умение работать в визуальной среде программирования | | Items (1.0 балл) |
| C:\Documents and Settings\Наталья\Рабочий стол\кнопка с изображением.JPG30Впишите название объекта, изображенного на рисунке | Знание имен объектов. Умение работать в визуальной среде программирования. Знание специализированной терминологии | | BitBtn (1.0 балл) |
| 31 Что будет выведено в объект Label1 после выполнения кода программы, если для значения чисел берутся значения по умолчанию  ***void \_\_fastcall TForm1::Button1Click(TObject \*Sender)***  ***{ inta,b;***  ***a=StrToInt(InputBox("Слагаемые","Введитепервоечисло", "3"));***  ***b=StrToInt(InputBox("Слагаемые","Введите первое число", "4"));***  ***Label1->Caption=IntToStr(++a+b);***  ***}*** | Умение разработки программного кода, тестирования программного кода. Умение работать в визуальной среде программирования. Знание специализированной терминологии | | 8 (1.0 балл) |
| 32 Верно ли утверждение, что компонент StringGrid нельзя использовать в задачах на одномерные массивы?   * Верно * Неверно | Знание объектов и их свойств. Умение работать в визуальной среде программирования.Знание специализированной терминологии | | Неверно(1.0 балл) |
| Практические задания | | | |
| 1. Создайте приложение, которое считает количество положительных элементов двумерного массива, стоящих на главной и побочной диагоналях. Массив заполнить случайным образом. Размерность матрицы – четное число. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое вычисляет стоимость поездки на автомобиле, если известно расстояние (км), цена на бензин (рубль/литр) и потребление бензина (литров на 100 км). Должна предусматриваться возможность выбора подсчета стоимости поездки туда или туда и обратно (использовать один флажок). | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое рисует на форме 3 ряда по 5 квадратов с зеленым контуром и желтой заливкой (между квадратами в ряду и между рядами должно быть расстояние). | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое выполняет арифметические действия над двумя числами. В программе используйте меню следующей структуры:   Файл Действия О программе  Выход Сложение  Вычитание  Умножение  Деление  Возвести в степень | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое определяет минимальный положительный элемент двумерного массива, заполненного с клавиатуры. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое на поверхности формы рисует флаг Лаоса. Подпишите рисунок с помощью функции:   C:\Documents and Settings\Админ\Рабочий стол\laos.gif  Белый  Синий  Красный  Красный | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, в котором пользователь вводит 5 чисел с помощью диалоговых окон и находит максимальное из них. Результат вывести через диалоговое окно. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое считает сумму элементов, кратных 3 одномерного массива. Массив и его размерность вводятся с клавиатуры. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое позволяет менять параметры шрифта (видимость, курсив и полужирный). Параметры шрифта можно выбрать с помощью флажков, а результаты изменения должны отображаться в метке. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Напишите программу для расчета сопротивления резисторов, соединенных параллельно или последовательно. Выбор соединения должен осуществляется с помощью переключателей. Сопротивление для последовательного соединения вычисляется по формуле R=R1+R2; а для параллельного R=(R1\*R2)/(R1+R2). | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Практическое задание. Создайте приложение, которое вычисляет площадь кольца, если дан радиус внешней и радиус внутренней окружностей. Площадь кольца вычисляется по формуле: πr2 | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |
| 1. Создайте приложение, которое выводит самое длинное слово в символьном массиве. Массив и его размерность вводятся с клавиатуры. | | работать в среде программирования, разрабатывать код программного модуля на языке программирования;создавать программу как отдельный модуль;  выполнять отладку и тестирование программы на уровне модуля;оформлять документацию на программные средства. | 1 балл – спроектирована форма в соответствии с заданием, программа работает без ошибок, программа соответствует всем требованиям задания, текст программы оформлен в соответствии с стандартом (ГОСТ 19.401-78)  1 балл – программа протестирована и отлажена, учитывая задание. Интерфейс помогает пользователю пользоваться приложением. |

**3. Комплект оценочных средств для оценки по учебной и производственной практике**

**3.1 Паспорт комплекта оценочных средств**

Фонд оценочных средств предназначен для проверки результатов освоения следующего вида деятельности — разработка программных модулей программного обеспечения для компьютерных систем в ходе учебной и производственной практики

**3.2.Оценка по учебной практике и производственной практике**

**Показатели оценки приобретенного опыта, освоенных умений**

|  |  |  |
| --- | --- | --- |
| **Результаты**  **(опыт и освоенные умения)** | **Основные показатели** | **Формы, методы контроля и оценки** |
| **Уметь:** |  |  |
| осуществлять разработку кода программного модуля на современных языках программирования; | Программный код программного модуля создает на современном языке программирования | Выполнение заданий в ходе учебной практики |
| создавать программу по разработанному алгоритму как отдельный модуль; | Осуществляет создание программы по разработанному алгоритму как отдельного модуля;  Осуществляет разработку кода программного продукта на основе готовой спецификации на уровне модуля; | Выполнение заданий в ходе учебной практики. Выполнение заданий в ходе производственной практики |
| выполнять отладку и тестирование программы на уровне модуля; | Производит отладку программы на уровне модуля в соответствии с требованиями к приложению;  Отладку программных модулей осуществляет с применением основных принципов отладки программных продуктов;  Тестирование программных модулей осуществляет с применением основных принципов тестирования программных продуктов;  Применяет инструментальные средства на этапах отладки и тестирования программного продукта | Выполнение заданий в ходе учебной практики |
| оформлять документацию на программные средства; | Оформляет документацию в соответствии с требованиями | Выполнение заданий в ходе учебной практики. |
| использовать инструментальные средства для автоматизации оформления документации; | Для оформления документации использует инструментальные средства. | Выполнение заданий в ходе учебной практики |
| **Иметь опыт:** |  |  |
| разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования | Разрабатывает алгоритм поставленной задачи и реализует его средствами автоматизированного проектирования | Выполнение заданий в ходе производственной практики |
| разработки кода программного продукта на основе готовой спецификации на уровне модуля | Разрабатывает код программного продукта на уровне модуля | Выполнение заданий в ходе производственной практики  Выполнение заданий в ходе учебной практики |
| использования инструментальных средств на этапе отладки программного продукта | Производит отладку программы на уровне модуля в соответствии с требованиями к приложению;  Отладка программных модулей осуществляет с применением основных принципов отладки программных продуктов;  Применяет инструментальные средства на этапе отладки программного продукта | Выполнение заданий в ходе производственной практики  Выполнение заданий в ходе учебной практики |
| проведения тестирования программного модуля по определенному сценарию | Тестирование программных модулей осуществляет с применением основных принципов тестирования программных продуктов;  Применяет инструментальные средства на этапетестирования программного продукта | Выполнение заданий в ходе производственной практики  Выполнение заданий в ходе учебной практики |

**Учебная практика** проводится в форме учебных занятий в системе программирования по проектированию, разработке, отладке и тестированию приложений.

В ходе учебных практик формируются умения:

* осуществлять разработку кода программного модуля на современных языках программирования
* создавать программу по разработанному алгоритму как отдельный модуль
* выполнять отладку и тестирование программы на уровне модуля
* оформлять документацию на программные средства

В ходе учебных практик приобретается практический опыт:

* 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

Виды работ

Учебная практика 1:

Разработка алгоритма и кода приложений:

* Использование функций API в разработке приложений
* Разработка приложений на работу с сообщениями
* Разработка приложений на использование связи между сообщениями и событиями

Отладка и тестирование приложений

Учебная практика 2:

Разработка алгоритма и кода приложений:

* + Проверка знаний
  + Горячие клавиши
  + Тест по С++
  + Текстовый редактор
  + Калькулятор
  + Просмотрщик картинок
  + Графический редактор

Отладка и тестирование приложений

Подготовка технической документации

Отметка «зачтено» ставится, если

* выполнены все виды работ
* выполнена отладка и тестирование приложения,
* код программы оформлен в соответствии со стандартом оформления документации.

АТТЕСТАЦИОННЫЙ ЛИСТ ПО УЧЕБНОЙ ПРАКТИКЕ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

*ФИО*

обучающийся(аяся) на III курсе по специальности СПО

09.02.03 Программирование в компьютерных системах,

успешно прошел(ла) учебную практику по профессиональному модулю

ПМ 01 «Разработка программных модулей программного обеспечения для компьютерных систем»

в объеме \_18\_ часов с «\_\_\_» \_\_\_\_\_\_\_20\_\_\_г. по «\_\_\_» \_\_\_\_\_\_\_20\_\_\_г.,

в объеме \_54\_ часов с «\_\_\_» \_\_\_\_\_\_20\_\_\_г. по «\_\_\_» \_\_\_\_\_\_\_20\_\_\_г.

в организации \_\_ГБПОУ «Дзержинский педагогический колледж»

Виды и качество выполнения работ

|  |  |  |
| --- | --- | --- |
| **ПК, ОК** | **Виды и объем работ** | **Качество выполнения** |
| ПК 1.1,ПК 1.2, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Разработка алгоритма и кода приложений:   1. Использование функций API в разработке приложений 2. Разработка приложений на работу с сообщениями 3. Разработка приложений на использование связи между сообщениями и событиями |  |
| ПК 1.3, ПК 1.4, ПК 1.5, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Отладка и тестирование приложений |  |
| ПК 1.1,ПК 1.2, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Разработка алгоритма и кода приложений:  Проверка знаний  Горячие клавиши  Тест по С++  Текстовый редактор  Калькулятор  Просмотрщик картинок  Графический редактор |  |
| ПК 1.3, ПК 1.4, ПК 1.5, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Отладка и тестирование приложений |  |

Итог учебной практики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Характеристика учебной и профессиональной деятельности обучающегося во время учебной практики (*дополнительные критерии по выбору ОУ*)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата «\_\_\_» \_\_\_\_\_20\_\_ г. Подпись руководителя практики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО руководителя практики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Производственная практика** предназначена для овладения студентами первичным профессиональным опытом и закрепления учебных умений и навыков по профессиональному модулю «Разработка программных модулей программного обеспечения для компьютерных систем».

В ходе производственной практики продолжается работа по приобретению практического опыта

* + - 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..

А так же формируются следующие умения:

* + - осуществлять разработку кода программного модуля на современных языках программирования;
    - создавать 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.

Практика по получению первичных профессиональных навыков проводится в сроки, предусмотренные учебным планом и графиком учебного процесса.

Руководство осуществляют преподаватели колледжа.

По результатам практики оформляется портфолио практики, которое включает в себя:

1. Дневник
2. Отчет диск с электронным вариантом выполненных заданий.
3. Отзыв-характеристика руководителя с отметкой по 5-бальной шкале, подписью руководителя, заверенной печатью.

**1.** Все виды и объем работ фиксируются студентом и заверяются подписью руководителя практики от предприятия в **дневнике практики.**

Дневник заполняется практикантом ежедневно рукописно в соответствии с задачами практики.

**Заполнение дневника**

|  |  |  |
| --- | --- | --- |
| Дата | Задание руководителя | Выполненные работы (подробное описание) |
|  |  |  |

**2.**  По окончании практики студентом составляется**ОТЧЕТ** следующей структуры:

Титульный лист отчета

Содержание отчета

1. **Анализ деятельности предметной области предприятия.**
   1. **Предварительная информация о предприятии**
      * Полное наименование предприятия
      * Организационно-правовая форма
      * Юридический адрес
      * Фио руководителя предприятия
      * Основная деятельность предприятия
      * Фио руководителя практики
   2. **Структура предприятия**
      * Организационная структура предприятия (провести сбор первичной информации о структуре предприятия и оформить организационную диаграмму в word)
   3. **Уровень автоматизации**

* Перечень используемого оборудования (рабочее место): сведения о системе, процессор, монитор, видеокарта, модем, маршрутизатор и др. с кратким описанием.
* Перечень программного обеспечения (рабочее место): операционная система, программы с кратким описанием.
  1. **Должностные обязанности** специалистов по обслуживанию технического**,** программного и информационного обеспечения и требования техники безопасности при работе за компьютером.

1. **Разработка приложения**

* Ознакомление с системным программным обеспечением на предприятии. Выполнение заданий на рабочем месте по системному программированию
* Ознакомление с прикладным программных обеспечением на предприятии. Выполнение заданий на рабочем месте по прикладному программированию:
* Выделение спецификаций и проектирование решаемой задачи
* Описание структуры решаемой задачи и ее модулей
* Описание алгоритмов реализации модулей.Разработка и написание кода программного модуля
* Тестирование и отладка программного продукта
* Документирование программного продукта
* Изучение законодательных и нормативных документов в области стандартизации и сертификации ПО
* Создать приложение по заданию руководителя практики.

1. **Самоанализ практики с выводами и предложениями**
   1. Объем выполнения (полностью, частично и т.д.).
   2. Время выполнения (с соблюдением сроков или нет).
   3. Качество выполненной работы (достигнута ли поставленная цель).
   4. Самостоятельность выполнения.
   5. Какие возникли трудности при выполнении работы.
   6. Рекомендации по самосовершенствованию профессиональных знаний и навыков (при наличии трудностей))

Отчет оформляется в электронном и печатном виде в соответствии с требованиями:

1. Шрифт Times New Roman, 14 пт,
2. межстрочный интервал - 1,5,
3. поля: 3 см левое, 1,5 см правое, 2 см верхнее, 2 см нижнее.
4. Нумерация страниц – арабскими цифрами внизу по центру.
5. Содержание в начале документа.
6. Каждый пункт – с новой страницы; подпункты не должны начинаться с новой страницы, следующий подпункт отделяется от предыдущего пустой строкой.

По окончании практики руководитель от предприятия составляет отзыв-характеристику, который заверяется подписью руководителя предприятия и печатью (вложить в отчет).

Производственная практика засчитывается по пятибалльной шкале на основании критериев:

1. Ведение документации (сдан отчет, дневник и отзыв характеристика).
2. Соблюдение сроков сдачи документации и прохождения практики.
3. Теоретическая подготовка – оценивается на основании отчета (реализации задач практики)
4. Профессиональная компетентность - оценивается на основании дневника (оценка работ руководителем практики от предприятия) и характеристики.
5. Творческие способности - оценивается качество решения практического задания.
6. Социально-личностные качества - оценивается на основании характеристики.

Оценка за практику выставляется на основании аттестационного листа.

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРАКТИКЕ

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРАКТИКЕ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

*ФИО*

обучающийся(аяся) на III курсе по специальности СПО

09.02.03 Программирование в компьютерных системах

успешно прошел(ла) производственную практику по профессиональному модулю   
ПМ 01 «Разработка программных модулей программного обеспечения для компьютерных систем»

в объеме \_144\_ часа с «\_\_\_» \_\_\_\_\_\_\_20\_\_\_г. по «\_\_\_» \_\_\_\_\_\_\_20\_\_\_г.,

в организации \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*наименование организации, юридический адрес*

Виды и качество выполнения работ

|  |  |  |
| --- | --- | --- |
| **ПК, ОК** | **Виды и объем работ** | **Качество выполнения** |
| ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Ознакомление с местом проведения практики. Инструктаж по технике безопасности  Сбор информации о структуре предприятия  Построение организационной структуры предприятия  Анализ уровня автоматизации предприятия  Изучение должностных обязанностей программиста на предприятии |  |
| ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Ознакомление с системным программным обеспечением на предприятии  Выполнение заданий на рабочем месте по системному программированию |  |
| ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Ознакомление с прикладным программных обеспечением на предприятии  Выполнение заданий на рабочем месте по прикладному программированию |  |
| ПК 1.1, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Выделение спецификаций и проектирование решаемой задачи |  |
| ПК 1.1, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Описание структуры решаемой задачи и ее модулей  Описание алгоритмов реализации модулей |  |
| ПК 1.2, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Разработка и написание кода программного модуля |  |
| ПК 1.3, ПК 1.4, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Тестирование и отладка программного продукта |  |
| ПК 1.6, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 9, ОК 10 | Изучение законодательных и нормативных документов в области стандартизации и сертификации ПО  Документирование программного продукта |  |
| ОК 1-9 | Дифференцированный зачет |  |

Характеристика учебной и профессиональной деятельности обучающегося во время производственной практики (*дополнительные критерии по выбору ОУ*)

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Итоговая отметка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата « » \_\_\_\_\_\_\_\_\_ 20 г.

Подпись руководителя практики\_\_\_\_\_\_\_\_\_\_\_\_\_ ФИО руководителя практики\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Комплект оценочных средств для организации промежуточного контроля и оценки освоения программы профессионального модуля в ходе экзамена (квалификационного)**

В качестве итоговой аттестации по модулю после завершения обучения проводится экзамен (квалификационный), на котором представители работодателей и учебного заведения проверяют готовность обучающегося к выполнению соответствующего вида профессиональной деятельности и сформированность у него профессиональных (ПК) и общих компетенций (ОК).

Экзамен (квалификационный) проводится после изучения МДК 01.01, МДК 01.02 модуля, прохождения учебных и производственной практик.

Экзамен (квалификационный) учитывает: оценку освоения МДК, прохождение учебных и производственной практик.

Экзамен (квалификационный) определяет уровень и качество освоения образовательной программы, проверяет готовность обучающегося к выполнению соответствующего вида профессиональной деятельности и сформированность у него компетенций.

Итогом проверки является однозначное решение «вид профессиональной деятельности освоен / не освоен».

Экзамен квалификационный проводится в форме защиты портфолио с предоставлением презентации портфолио.

**Требования к портфолио**

Портфолио - способ фиксирования, накопления и оценки индивидуальных достижений обучающегося (подборка сертифицированных достижений, наиболее значимых работ и отзывов на них).

Содержание портфолио на бумажном носителе:

1. Титульный лист
2. Аннотирование содержания с перечислением основных элементов портфолио:
3. Портфолио по МДК 01.01:
   1. - тестовый контроль,
   2. - лабораторные и практические работы,
   3. - контрольные работы,
   4. - самостоятельные работы (творческие проекты, рефераты, презентации, доклады, расчеты, эссе, сообщения - по выбору)
   5. - грамоты, дипломы, сертификаты
4. Аттестационный лист по учебной практике 1
5. Портфолио по МДК 01.02:
   1. - тестовый контроль,
   2. - лабораторные и практические работы,
   3. - контрольные работы,
   4. - самостоятельные работы (творческие проекты, рефераты, презентации, доклады, расчеты, эссе, сообщения - по выбору)
   5. - грамоты, дипломы, сертификаты
6. Аттестационный лист по учебной практике 2
7. Оценочный лист по производственной практике и портфолио
   1. Дневник
   2. Отчет
   3. Отзыв-характеристика
   4. Прилагаемые работы
   5. Аттестационный лист по производственной практике
8. Оценочный лист ПМ
9. Самоанализ освоения программы и прогноз на будущее.

Портфолио носит смешанный характер, состоящий из документов, работ обучающегося, отзывов.

Перечень прилагаемых документов: дипломы, грамоты, сертификаты олимпиад, конкурсов профессионального мастерства, творчества; документы по тестированию; удостоверения и сертификаты по курсам дополнительной профессиональной подготовки и т.п.

Перечень прилагаемых работ: творческие, проектные, исследовательские работы; ведомость или зачетная книжка о прохождении курсов по выбору, электронные документы, фотографии, видеозаписи, проекты, отчеты о практиках и т.п.

Перечень прилагаемых отзывов: тексты заключений, рецензии, отзывы, резюме, эссе, рекомендательные письма.

Лица, проводящие оценку – члены экзаменационной комиссии (работодатели, представители учебного заведения) фиксируют представленные доказательства освоения ПК в итоговой оценочной ведомости ПК через оценочные листы МДК модуля и учебной практики, документов производственной практики (дневника, производственной характеристики с указанием рекомендуемого разряда), портфолио.

При возникновении вопроса о степени освоения ПК данного модуля по представленным выше документам, комиссия может уточнить путем постановки устных вопросов степень освоения ПК и ОК. Формулировки устных вопросов должны быть четкими, ясными доступными для понимания студентов.

Итоговая оценочная ведомость ПК экзамена (квалификационного), включает в себя оценку МДК модуля по оценочным листам МДК (освоил, не освоил ОТМЕТКА), оценку учебной практики по оценочным листам (освоен, не освоен ЗАЧТЕНО/НЕЗАЧТЕНО) и производственной практики модуля (по представленному дневнику (отработал, не отработал ОТМЕТКА), производственной характеристики руководителя предприятия (с указанием степени освоения ПК), отзыва руководителя практики от учебного заведения (преподавателя), анализа достижений студента (портфолио), оценки сдачи экзамена (сдал, не сдал), заключение комиссии (освоил ПК, ОК; не освоил).

**Пакет экзаменатора**

**Условия проведения экзамена**

Экзамен квалификационный проводится в форме защиты портфолио. Проходит в кабинете теоретического обучения в соответствии с графиком экзаменов и приказом директора. Кабинет должен быть оснащен мультимедийной техникой.

Студент входит и называет свою фамилию. Преподаватель дает краткую характеристику данному студенту по степени освоения МДК модуля, затем руководитель учебной и производственной практики дает характеристику по освоению программы практики.

После этого студент представляет подготовленную презентацию своего портфолио.

Содержание презентации портфолио:

1. Титульный лист.
2. Перечень работ ПМ.
3. Лучшие работы (3-10 шт.), которые показывают умения обучающегося в соответствии с программой:
   1. Программы по системному программированию
   2. Программы по прикладному программированию
4. Лучшие творческие практические работы опережающего характера (1-2).
5. Работы, выполненные в ходе учебной практики
6. Работы, выполненные в ходе производственной практики (от работодателя)
7. Рекомендательное письмо (если есть)
8. Документы достижений: грамоты, свидетельства, дипломы, удостоверения.
9. Самоанализ освоения программы ПМ.

Квалификационная комиссия просматривает содержание портфолио студента, проверяет подготовленные документы: оценочный лист МДК ПМ, аттестационный лист учебной практики ПМ, аттестационный лист по производственной практике, итоговую ведомость оценки ПМ.

Комиссия может задать устные вопросы по изученному профессиональному модулю для уточнения степени освоения ПК данного модуля.

После защиты портфолио, студент выходит из аудитории, комиссия принимает решение и дает заключение «данный ПМ освоен / не освоен», и заполняет итоговую ведомость оценки ПМ, оценочные листы.

**Приложение**

**Форма оценочного листа**

|  |  |  |
| --- | --- | --- |
| **Коды проверяемых компетенций** | Показатели оценки результата | **Оценка (да / нет)** |
| ПК 1.1. Выполнять разработку спецификаций отдельных компонент. | Разработка приложений производится в соответствии с основными этапами разработки программного обеспечения;Разработка приложений осуществляется с применением основных принципов технологии структурного и объектно-ориентированного программирования;Оформление документации проводится в соответствии с требованиями, предъявляемыми к оформлению документации на программные средства. |  |
| ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля. | При разработке кода применяются основные принципы технологии структурного и объектно-ориентированного программирования;Разработка кода программного модуля осуществляется на современных языках программирования;Осуществляется создание программы по разработанному алгоритму как отдельного модуля;Осуществляется разработка кода программного продукта на основе готовой спецификации на уровне модуля; |  |
| ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств. | Отладка программных модулей осуществляется с применением основных принципов отладки и тестирования программных продуктов;Осуществляется использование инструментальных средств на этапе отладки программного продукта;Осуществляется проведение отладки и тестирования программы на уровне модуля; |  |
| ПК 1.4. Выполнять тестирование программных модулей. | Осуществляется проведение тестирования программного модуля по определенному сценарию;Осуществляется выполнение отладки и тестирование программы на уровне модуля. |  |
| ПК 1.5. Осуществлять оптимизацию программного кода модуля. | Осуществляется проведение оптимизации программного кода модуля по определенному сценарию;Осуществляется выполнение отладки и тестирования программы на уровне модуля;Осуществляется использование инструментальных средств на этапе отладки программного продукта |  |
| ПК 1.6. Разрабатывать компоненты проектной и техническойДокументации с использованием графических языков спецификаций. | Осуществляется использование инструментальных средства для автоматизации оформления документации;Определение и использование методов и средств разработки технической документации осуществляется в соответствии с поставленными задачами |  |

**Форма оценочной ведомости**

*(заполняется на каждого обучающегося)*

Все части ведомости до пункта «Итоги экзамена (квалификационного)» должны быть заполнены до начала очной части экзамена (квалификационного).

**ГБОУ СПО «Дзержинский педагогический колледж»**

**Оценочная ведомость по профессиональному модулю**

**ПМ 01Разработка программных модулей программного обеспечения   
для компьютерных систем**

Ф.И.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Обучающийся на \_III\_ курсе по специальности \_09.02.03 «Программирование в компьютерных системах» освоил программу профессионального модуля \_в объеме\_\_720\_\_ часов с \_\_\_\_\_\_\_\_\_\_\_\_\_\_ по \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Результаты промежуточной аттестации по элементам модуля**

|  |  |  |
| --- | --- | --- |
| Элементы модуля (код и наименование МДК, код практик) | Формы промежуточной аттестации | Оценка |
| МДК.01.01 Системное программирование | Дифференцированный зачет |  |
| МДК.01.02 Прикладное программирование | Экзамен |  |
| УП. 01 Учебная практика 1 | Зачет |  |
| УП. 01 Учебная практика 2 | Зачет |  |
| ПП. 01 Производственная практика | Дифференцированный зачет |  |

Оценка освоения общих и профессиональных компетенций

|  |  |  |  |
| --- | --- | --- | --- |
| Коды проверяемых компетенций | Наименование общих и профессиональных компетенций | Оценка  (да/нет) | Если нет, то что должен обучающийся сделать дополнительно (указать сроки) |
| 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 |  |  |
| 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 | Решать проблемы, оценивать риски и принимать решения в нестандартных ситуациях. |  |  |
| ОК 4 | Осуществлять поиск, анализ и оценку информации, необходимой для постановки и решения задач, профессионального и личностного развития. |  |  |
| ОК 5 | Использовать информационно-коммуникационные технологии для совершенствования профессиональной деятельности. |  |  |
| ОК 6 | Работать в коллективе и команде, обеспечивать ее сплочение, эффективно общаться с коллегами, руководством, потребителями. |  |  |
| ОК 7 | Ставить цели, мотивировать деятельность подчиненных, организовывать и контролировать их работу с принятием на себя ответственности за результат выполнения заданий. |  |  |
| ОК 8 | Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации |  |  |
| ОК 9 | Быть готовым к смене технологий в профессиональной деятельности. |  |  |

Результат оценки: вид профессиональной деятельности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Всего часов на проведение\_\_\_\_\_\_\_\_ час \_\_\_\_\_\_\_ мин

Подписи членов экзаменационной комиссии \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)

\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)

\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_)

**Контрольно-измерительные материалы для текущего контроля   
по МДК 01.01 Системное программирование**

*Лабораторная работа №1*

***«ПРИЕМЫ РАБОТЫ С КОМАНДНОЙ СТРОКОЙ И  
 ПЕРЕМЕННЫМИ СРЕДЫ WINDOWS» (2ч.)***

***Цель работы:*** *познакомиться с параметрами командной строки и переменными среды Windows.*

Стандарты языка C++ пакета С++Builder позволяют создавать программное обеспечение не только прикладного, но и системного уровня. Например, возможна обработка параметров командной строки.

Число параметров командной строки может быть прочитано из переменной **ParamCount**, причем доступ к i-тому параметру командной строки осуществляется как **ParamStr(i)**, где **i** изменяется от **0** до **ParamCount-1**.

Нижеследующий фрагмент кода заносит в компонент Memo1 содержимое параметров командной строки, а в компонент Memo2 содержимое переменных среды Windows:

Memo1->Clear(); // очистка Memo1

// заполнение списка Memo1 параметрами командной строки

for (i=0; i <= ParamCount(); i++)

Memo1->Lines->Add(ParamStr(i));

Memo2->Clear();// очистка Memo2

// заполнение списка Memo2 переменными среды Windows

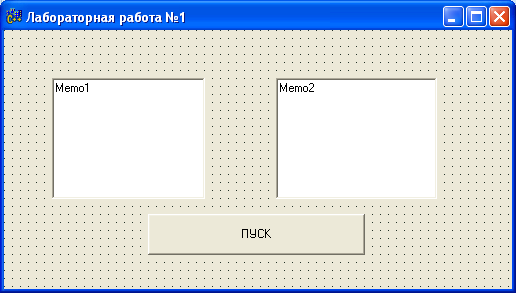
i=0;

while (\_environ[i])

Memo2->Lines->Add(\_environ[i++]);

**Задание.**

1. **Программным путем получите информацию о параметрах командной строки и переменных среды Windows. Для этого в среде С++Builder создайте приложение следующего вида:**

****

1. **Раскройте значение переменных среды Windows.**

Требования к оформлению работы

По каждой лабораторной работе составляется отчет, который должен содержать:

- титульный лист;

- название и цель работы;

- лабораторное задание;

- распечатку формы до запуска приложения и с результатами выполнения приложения;

- программный код с пояснениями;

- выводы, которые должны доказывать или оценивать правильность составленной программы или объяснять допущенные ошибки.

**Лабораторная работа №2**

**«Работа с командами MSDOS»**

**Цель работы: з**накомство с основными командами MS DOS для работы с файлами, каталогами, дисками и приобретение навыков работы с операционной системой MS DOS.

**Постановка задачи работы:**

1. На диске D создайте структуру каталогов и файлов в соответствии с представленным скриншотом структурной схемы

lesson1.txt  
lesson2.txt

FAVORITE  
 favorite1.txt  
 favorite2.txt

LIBRARY  
 ARCTICLE  
 arcticle1.txt  
 arcticle2.txt  
 favorite1.txt

BOOK  
 book1.txt  
 book2.txt  
 lesson1.txt

**D:**

2. Создайте директории первого уровня D:\FAVORITE, D:\LIBRARY и второго уровня D:\LIBRARY\ARTICLE, A:\LIBRARY\BOOK, вложенные в каталог LIBRARY.
3. Текстовые файлы lesson1.txt и lesson2.txt создайте путем ввода текста с клавиатуры.
4. Файлы favorite1.txt и favorite2.txt создайте методом копирования и вставки текста из других файлов.
5. Файлы article1.txt и article2.txt создайте методом копирования файлов favorite1.txt и favorite2.txt в каталог D:\LIBRARY\ARTICLE с изменением имен файлов на article1.txt и article2.txt соответственно.
6. Файлы book1.txt и book2.txt создайте путем слияния файлов lesson1.txt и favorite1.txt, lesson2.txt и favorite2.txt соответственно.
7. Скопируйте файл lesson1.txt без изменения имени в директорию D:\LIBRARY\BOOK.
8. Скопируйте файл favorite1.txt без изменения имени в директорию D:\LIBRARY\ARTICLE.

**Пошаговое выполнение работы:**

1. **Загрузите командную строку и сделайте активным диск D:**
2. **Создайте структуру каталогов на диске D:**

Сначала создайте каталоги первого уровня D:\FAVORITE и D:\LIBRARY, затем в директории D:\LIBRARY создайте вложенные каталоги второго уровня (подкаталоги) D:\LIBRARY\ARTICLE и D:\LIBRARY\BOOK..

1. **В корневом каталоге диска D: создайте файлы lesson1.txt и lesson2.txt**

В файл lesson1.txt введите свою фамилию, в lesson2.txt **-**имя с отчеством. Создание текстовых файлов с клавиатуры осуществляется командой copycon. После ввода этой команды можно вводить текст (текст произвольный). В конце каждой строки надо нажимать клавишу Enter, а после ввода последней – нажать F6 (или Ctrl +Z) и затем Enter. На диске появится файл с указанным именем.

1. **В директории D:\FAVORITE создайте файлы favorite1.txt и favorite2.txt**

Файлы favorite1.txt и favorite2.txt создайте методом копирования и вставки текста из других файлов.   
Командой dir проверьте наличие файлов в каталоге D:\FAVORITE, а командой typeпроверьте наличие текста в них. В файл favorite1.txt введите дату рождения, в favorite2.txt -место рождения

1. **Создайтефайлы article1.txt и article2.txt вкаталог D:\LIBRARY\ARTICLE**

Файлы article1.txt и article2.txt создайтеметодомкопированияфайлов favorite1.txt и favorite2.txt вкаталогD:\LIBRARY\ARTICLE сизменениемименфайловна article1.txt и article2.txt соответственно, используякомандуcopy.Например, D:\>copy D:\favorite\favorite1.txt D:\library\article\article1.txt

1. **Создайтефайлы book1.txt и book1.txt путемслиянияфайлов lesson и favorite**

Файлы book1.txt и book2.txt создайтепутемслиянияфайлов lesson1.txt и favorite1.txt, lesson2.txt и favorite2.txt соответственно. Слияние (объединение файлов в один) осуществляется по следующему алгоритму: copy полное имя 1–го файла + полное имя 2–го файла полное имя 3-го файла, нажать Enter.

1. **Скопируйте файл lesson1.txt без изменения имени в директорию D:\LIBRARY\BOOK**
2. **Скопируйте файл favorite1.txt без изменения имени в директорию D:\LIBRARY\ARTICLE**

Проверьте наличие скопированных файлов командой **tree/f**, а командой **type**проверьте наличие текста в них.

1. **Завершение работы**

Сообщите преподавателю о выполненной работе.

**Лабораторная работа №3. Использование функций API**

API (ApplicationProgrammingInterface) – это набор функций, которые операционная система предоставляет программисту. Эти функции находятся в стандартных динамически компонуемых библиотеках (DLL-файлы), которые содержатся в системном каталоге. Использование функций API позволяет достаточно просто выполнять различного рода задачи.

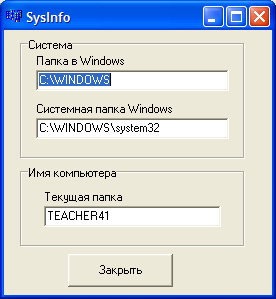
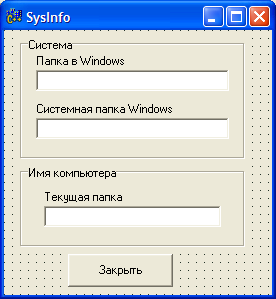
**Информация о системе**

Создайте приложение, в котором будет выводиться информация о системе, используя следующие функции:

* GetWindowsDirectory — возвращает путь к каталогу Windows;
* GetSystemDirectory — возвращает путь к системному каталогу Windows;
* GetComputerName— возвращает имя компьютера.

Разместите на форме следующие объекты и задайте им соответствующие свойства:

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Firm1 | Caption | SysInfo |
| GroupBox1 | Caption | Система |
| GroupBox2 | Caption | Имя компьютера |
| LabeledEdit1 (вкладка Additional) | Text |  |
| EditLabel-> Caption | Папка в Windows |
| LabeledEdit2 | Text |  |
| EditLabel-> Caption | Системная папка Windows |
| LabeledEdit3 | Text |  |
| EditLabel-> Caption | Текущая папка |



Программный код:

#include<vcl.h>

#pragma hdrstop

#include "Unit1.h"

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

void \_\_fastcall TForm1::FormCreate(TObject \*Sender)

{

charWindowsDirectory[MAX\_PATH];GetWindowsDirectory(WindowsDirectory, MAX\_PATH);

LabeledEdit1->Text=WindowsDirectory;charSystemDirectory[MAX\_PATH];

GetSystemDirectory(SystemDirectory, MAX\_PATH);LabeledEdit2->Text=SystemDirectory;

unsigned long Size = MAX\_COMPUTERNAME\_LENGTH + 1;char \*Buffer = new char[Size];

GetComputerName(Buffer, &Size);LabeledEdit3->Text=Buffer;

delete [] Buffer;}

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{Close();}

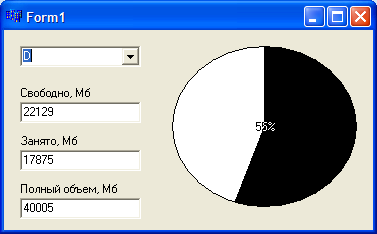
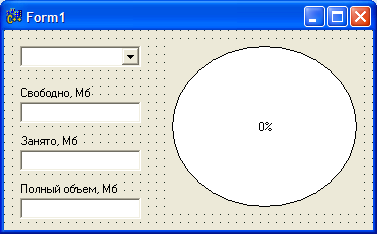
**Информация о дисках**

Создайте приложение, в котором будет выводиться системная информация о дисках, используя следующие функции:

* DiskSize — возвращает размер диска в байтах. В качестве входного параметра используется целое число, которое означает: 0 — текущий диск, 1 — А, 2 — В и т
* DiskFree — возвращает размер в байтах свободного пространства на диске.

Разместите на форме следующие объекты и задайте им соответствующие свойства:

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Form1 | Caption | hddInfo |
| ComboBox1 | Text |  |
| Items | A B C D E (ввести через Enter) |
| LabeledEdit1 (вкладка Additional) | Text |  |
| EditLabel-> Caption | Свободно, Мб |
| LabeledEdit2 | Text |  |
| EditLabel-> Caption | Занято, Мб |
| LabeledEdit3 | Text |  |
| EditLabel-> Caption | Полный объем, Мб |
| CGauge1 (вкладка Samples) | Kind | gkPie |
| BorderStyle | bsNone |



Программный код:

#include<vcl.h>

#pragma hdrstop

#include "Unit1.h"

#pragma package(smart\_init)

#pragma link "CGAUGES"

#pragma resource "\*.dfm"

TForm1 \*Form1;

\_\_int64 Size;

\_\_int64 FreeSpace;

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{ }

void \_\_fastcall TForm1::ComboBox1Change(TObject \*Sender)

{

if (ComboBox1->ItemIndex>-1)

{

Size=DiskSize(ComboBox1->ItemIndex+1);

FreeSpace = DiskFree(ComboBox1->ItemIndex+1);

if (Size>0)

{ LabeledEdit3->Text=IntToStr(Size/1024/1024);

LabeledEdit1->Text=IntToStr(FreeSpace/1024/1024);

LabeledEdit2->Text=IntToStr((Size-FreeSpace)/1024/1024);

CGauge1->Progress= FreeSpace\*100/Size;

} else

{

ShowMessage("Диска нет! \n Вставьте другой диск.");} } }

**Дополнительное задание**

Измените приложение так, чтобы информация о дисках выводилась в Гигабайтах

**Лабораторная работа №4**

**Копирование файлов с использованием Win32**

Последовательная обработка файлов – самая простая, наиболее обычная и наиболее необходимая функция любой файловой системы. Копирование файлов, часто с обновлением, и объединением отсортированных файлов – распространенные формы последовательной обработки. Простое копирование позволяет определить сильные и слабые стороны разных систем и перейти к Win32.

Цель работы – изучить копирование файлов с использованием стандартной библиотеки на языке С, с использованием интерфейса программирования приложений (API) Win32, а также с использованием функции-полуфабриката Win32.

**1. Копирование файлов с использованием библиотеки С. Программа Срс**.

Обращение к программе производится из командной строки: **Срс файл1, файл2**

#include <vcl.h>

#pragma hdrstop

#pragma argsused

#include <stdio.h>

#include <errno.h>

#define BUF\_SIZE 256

int main (intargc, char \*argv [])

{

FILE \*in\_file, \*out\_file; char rec

[BUF\_SIZE]; size\_tbytes\_in,

bytes\_out;

if (argc != 3)

{printf("Использование: Cрc file1 File2\n");return 1;}

in\_file = fopen (argv [1],"rb");

if (in\_file == NULL)

{perror (argv [1] ) ;return 2;}

out\_file = fopen (argv [2], "wb");

if (out\_file == NULL)

{perror (argv [2] ) ;return 3;}

/\* Обрабатываем входной файл по одной записи. \*/

while ( (bytes\_in = fread (rec, 1, BUF\_SIZE, in\_file) ) > 0)

{bytes\_out = fwrite (rec, 1, bytes\_in, out\_file) ;

if (bytes\_out != bytes\_in) {perror ("Неисправимаяошибказаписи.");return 4;}}

fclose (in\_file);

fclose (out\_file);}

**2. Копирование файлов с использованием API Win32.Программа Сpw.**

Обращение к программе производится из командной строки: **Cpw файл1, файл2**

#include <vcl.h>

#pragma hdrstop

#pragma argsused

#include <windows.h>

#include <stdio.h>

#define BUF\_SIZE 256

int main (intargc, LPTSTR argv [])

{

HANDLE hIn, hOut;

DWORD nIn, nOut;

CHAR Buffer [BUF\_SIZE];

if (argc != 3)

{printf ("Использование: CpW file1 File2\n");

return 1;}

hIn = CreateFile (argv [1], GENERIC\_READ, 0, NULL, OPEN\_EXISTING, 0, NULL);

if (hIn==INVALID\_HANDLE\_VALUE)

{printf ("Нельзя открыть входной файл.Ошибка: %х\n", GetLastError ());return 2; }

hOut = CreateFile (argv [2], GENERIC\_WRITE, 0, NULL,

CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

if (hOut==INVALID\_HANDLE\_VALUE)

{printf ("Нельзя открыть выходной файл.Ошибка: %х\n", GetLastError () ) ;return 3;}

while (ReadFile (hIn, Buffer, BUF\_SIZE, &nIn, NULL) &&nIn> 0)

{WriteFile (hOut, Buffer, nIn, &nOut, NULL);

if (nIn != nOut)

{printf ("Неисправимаяошибказаписи: %x\n", GetLastError ());

return 4;}}

CloseHandle (hIn);

CloseHandle (hOut);return 0;}

# Копирование файлов с использованием функции полуфабриката Win32. Программа cpСF.

Win32 содержит множество функций-полуфабрикатов, которые объединяют несколько функций для выполнения часто встречающихся задач. В некоторых случаях эти функции-полуфабрикаты могут улучшить быстродействие. Например, CopyFile значительно упрощает программу. В частности, здесь совершенно не нужно искать оптимальный размер буфера, который в двух предыдущих программах был произвольно задан в 256 байт.

Обращение производится из командной строки: сpCF файл1 файл2

#include <vcl.h>

#pragma hdrstop

#pragma argsused

#include <windows.h>

#include <stdio.h>

int main (intargc, LPTSTR argv [])

{

if (argc != 3)

{printf ("Использование: cpCF file1 file2\n");return 1; }

if (!CopyFile (argv [1], argv [2], FALSE))

{printf ("ОшибкаCopyFile: %x\n", GetLastError ());return 2; } return 0;}

Приведенные программы копирования файлов демонстрируют многочисленные различия между библиотекой С и Win32. Примеры с использованием Win32 демонстрируют стиль программирования и соглашения Win32, но дают лишь первое представление о функциональных возможностях, доступных в Win32.

**4.Содержание работы**

1. Изучить программы копирования файлов.
2. Последовательно набрать и отладить программы копирования файлов в среде C++ Builder6.
3. Выполнить задание по копированию файла 1 в файл 2, выбрав произвольные имена файлов.

Программы запустить из командной строки.

1. Подготовить отчет по выполненной работе. Листинги программ записать в тетрадь.
2. В отчете написать название, цель работы и ответить на контрольные вопросы.

**5.Контрольные вопросы**

1. Что такое API Win32?

2. Какие операционные системы обслуживает API Win 32?

3.Какие особенности имеет API Win 32?

4. Какие преимущества программирования дает API 32?

5. Какой основной тип переменных используется в Win 32?

6. Для управления каких систем могут быть написаны программы с использованием Win32?

7. Чтоозначаетстрокаint main (intargc, LPTSTR argv [])?

8. Поясните, какуюфункциювыполняетданныйоператор: hIn = CreateFile (argv [1], GENERIC\_READ, 0, NULL, OPEN\_EXISTING, 0, NULL);

9. Поясните, какуюфункциювыполняетданныйоператор: hOut = CreateFile (argv [2], GENERIC\_WRITE, 0, NULL, CREATE\_ALWAYS, FILE\_ATTRIBUTE\_NORMAL, NULL);

10.Поясните, какуюфункциювыполняетданныйоператор: while (ReadFile (hIn, Buffer, BUF\_SIZE, &nIn, NULL) &&nIn> 0)

11. Поясните, какую функцию выполняет данный оператор: WriteFile (hOut, Buffer, nIn, &nOut, NULL);

12. Поясните, какую функцию выполняет данный оператор: if (!СopyFile (argv [1], argv [2], FALSE)).

**Лабораторная работа № 5**. ***Работа с накопителями информации.***

***Цель работы:*** Ознакомление с возможностями для работы с накопителями информации в среде BorlandC++ Builder 6.

Основные функции для работы с накопителями информации:

***GetLogicalDrives: DWORD; – ф***ункция возвращает битовую маску, в которой хранится информация о доступных накопителях: наличие или отсутствие накопителя. То есть каждый бит отвечает за присутствие определённого логического диска. Либо, если бит не установлен, то это указывает на отсутствие логического диска. DWORD имеет размер 32 бита, этого вполне достаточно, чтобы уместить весь английский алфавит, который содержит 26 букв. Например: нулевой бит это информация о накопителе 'A', первый бит – информация о накопителе 'B', второй бит это информация о накопителе 'C' и.т.д. В случае ошибки функция возвращает нуль.

***GetLogicalDriveStrings(nBufferLength: DWORD; lpBuffer: PAnsiChar): DWORD; – ф***ункция записывает специальную переменную ***lpBuffer*** имена присутствующих в системе накопителей.

Параметры:

***nBufferLength:*** определяет максимальный размер в символах, буфера указанного в парметре***lpBuffer***. Этот размер не включает завершающего нулевого символа.

***lpBuffer***: Указатель на буфер, в которой записываются имена присутсвующих в системе накопителей в следующем формате: 'C:\nullD:\null<null>'. Т.е. на каждый диск отводится по 4 байта.

Возвращаемые значения

Если функция выполнилась без ошибок, то она возвращает длину в символах буфера, не включая завершающего нулевого символа. В случае ошибки возвращает нуль.

***GetDriveType(lpRootPathName:PChar): UINT;*** – функция возвращает тип носителя: сменный, фиксированный, CD-ROM, диск RAM, или сетевой диск.

Параметры:

***lpRootPathName:*** строка определяющая корневой каталог устройства, тип которого надо определить(например 'A:\'– для дисковода). Если ***lpRootPathName*** равен ***NULL***, то возвращается тип текущего накопителя.

Возвращаемые значения

Возвращаемое значение определяет тип накопителя. Может быть:

**0** – Тип накопителя не определен.

**1** – Корневой директорий не существует.

***DRIVE\_REMOVABLE*** – Накопитель может удаляться с накопителя.

***DRIVE\_FIXED*** – Фиксированный диск(не может быть удален).

***DRIVE\_REMOTE*** – Удаленный накопитель(сетевой диск).

***DRIVE\_CDROM*** – диск CD-ROM.

***DRIVE\_RAMDISK*** – Накопитель является виртуальным RAM-диском.

***GetDiskFreeSpaceEx(lpDirectoryName:PChar; &lpFreeBytesAvailableToCaller, &lpTotalNumberOfBytes, &lpTotalNumberOfFreeBytes: ULARGE\_INTEGER): BOOL;*** – функция получает информацию о дисковом пространстве доступного на накопителе: общий объем пространства, общий объем свободного пространства, и общий объем свободного пространства доступного для пользователя.

Параметры:

***lpDirectoryName***: строка определяющая корневой каталог устройства, размер которого надо определить(например 'A:\'– для дисковода). Если ***lpRootPathName*** равен ***NULL***, то выбирается текущий накопитель.

***lpFreeBytesAvailableToCaller***: общий объём свободного пространства доступного для пользователя.

***lpTotalNumberOfBytes***: общее число байтов на диске.

***lpTotalNumberOfFreeBytes***: общее число свободный байтов на диске.

Возвращаемые значения

Если функция выполнилась без ошибки, то она вернёт не нулевое значение.

***GetVolumeInformation(lpRootPathName:PChar; lpVolumeNameBuffer: PChar; nVolumeNameSize: DWORD; lpVolumeSerialNumber: PDWORD; &lpMaximumComponentLength, &lpFileSystemFlags: DWORD; &lpFileSystemNameBuffer: PChar; &nFileSystemNameSize: DWORD): BOOL; –*** возвращает информацию о файловой системе и о накопители информации.

Параметры:

l***pRootPathName***: строка определяющая корневой каталог устройства (например 'A:\'– для дисковода). Если ***lpRootPathName*** равен ***NULL***, то выбирается текущий накопитель.

***lpVolumeNameBuffer***: имя метки тома.

***nVolumeNameSize:*** Содержит длину в символах имени метки тома.

***lpVolumeSerialNumber:*** Указатель на переменную, которая содержит серийный номер тома.

***lpMaximumComponentLength:*** Указатель на переменную, которая возвращает максимальную длину допустимого имени файла на диске.

lpFileSystemFlags: Указатель на переменную, которая возвращает флаги связанные с текущей файловой системой. Этот параметр может быть комбинацией любых флагов кроме ***FS\_FILE\_COMPRESSION*** и ***FS\_VOL\_IS\_COMPRESSED***, которые взаимно исключены:

***FS\_CASE\_IS\_PRESERVED*** –файловая система сохраняет имя файла на диске.

***FS\_CASE\_SENSITIVE*** –файловая система поддерживает контекстно-зависимые имена файлов.

***FS\_UNICODE\_STORED\_ON\_DISK*** –файловая система поддерживает имена файлов в Unicode.

***FS\_PERSISTENT\_ACLS*** –файловая система сохраняет и осуществляет ACL. Например, NTFS сохраняет и осуществляет ACL,а FAT нет.

***FS\_FILE\_COMPRESSION*** – Файловая система поддерживает сжатие файлов.

***FS\_VOL\_IS\_COMPRESSED*** – Томявляетсясжатым; например DoubleSpace.

***lpFileSystemNameBuffer:*** Указатель на переменную, которая возвращает имя файловой системы (как например, FAT или NTFS).

***nFileSystemNameSize:*** Параметр определяет длину переменной имени файловой системы в символах.Возвращаемые значения: если функция выполнилась без ошибки, то она вернёт не нулевое значение.

**Практическая часть**

Запустите *C++Builder 6*. Сохраните проект в новой отдельной папке.

Присвойте основной форме заголовок, в качестве имени которого используйте название дисциплины, номер работы и варианта и свою фамилию.

Установите для формы ограничения изменения размеров максимум 800х600, минимум 600х400. Для этого используйте свойства **MaxHeight**, **MaxWidth** и **MinHeight**, **MinWidth**. Свойства находятся в свойстве **Constraints,** для доступа к ним нажмите на знак плюса напротив свойства **Constraints.**

Установите позицию показа формы по центру экрана (свойство **Position** =*poDesktopCenter*).

Выведите список всех логических дисков используемых системой, сначала с помощью функции ***GetLogicalDrives()***, а затем с помощью – ***GetLogicalDriveStrings()***. Сравните полученные результаты.

Для каждого логического диска определите:

его тип с помощью функции ***GetDriveType***;

размер диска, количество занятого пространства и количество свободного места на диске, количество места доступного пользователю, с помощью функции ***GetDiskFreeSpaceEx***;

тип файловой системы, максимальный размер имени, метку тома, атрибуты файловой системы диска с помощью функции ***GetVolumeInformation***.

Добавьте на форму текстовой поле и кнопку для выполнения индивидуального задания, согласно номеру варианта

|  |  |
| --- | --- |
| №п/п | Вариант задания |
| 1 | При нажатии кнопки должен создаваться каталог путь, к которому указан в текстовом поле. |
| 2 | При нажатии кнопки должен создаваться файл путь, к которому указан в текстовом поле. |
| 3 | Определить дату создания файла путь, к которому указан в текстовом поле. |
| 4 | Определить дату последнего изменения файла путь, к которому указан в текстовом поле. |
| 5 | Определить дату последнего открытия файла путь, к которому указан в текстовом поле. |
| 6 | При нажатии кнопки вывести в текстовое поле текущий каталог |
| 7 | При нажатии кнопки должен удаляться каталог путь, к которому указан в текстовом поле. |
| 8 | При нажатии кнопки должен удаляться файл путь, к которому указан в текстовом поле. |
| 9 | Определить размер файла путь, к которому указан в текстовом поле. |
| 10 | Определить архивный ли файл путь, к которому указан в текстовом поле. |
| 11 | Определить путь к системному каталогу Windows |
| 12 | Определить используется ли файл только для чтения, к которому указан в текстовом поле. |
| 13 | Установить для файла путь, к которому указан в текстовом поле атрибут только для чтения. |

Созданный проект покажите преподавателю.

Самостоятельная работа по теме  
 «Операционная система»

Вариант 1

1. Какое системное ПО называют сервисным?
2. Что называется операционной системой?
3. Перечислите на какие виды делятся ОС?
4. Что такое инструментарий технологии программирования?
5. Какие средства входят в инструментарий технологии программирования?

Вариант 2

1. Какое системное ПО называют базовым?
2. Что называется сетевой операционной системой?
3. Перечислите функции ОС.
4. Что такое язык программирования?
5. На какие классы делятся языки программирования?

Самостоятельная работа по теме «Потоки и процессы»

Вариант №1

1. Что называется объектом в Windows? Приведите примеры объектов.
2. Запишите прототип функции, которая используется для завершения работы с объектом.
3. Дайте определение контекста потока.
4. Что такое адресное пространство?
5. Опишите операции Suspend и Resume.
6. Пусть потоки имеют одинаковый приоритет. Тогда опишите, как они будут выполняться?

Вариант №2

1. Что представляет собой дескриптор потока? Для чего он необходим.
2. Опишите работу функции CloseHandle.
3. Дайте определение потока.
4. Какая программа называется многопоточной? Как называются потоки в такой программе?
5. Опишите операцию Sleep.
6. Пусть потоки имеют разные приоритеты. Тогда опишите, как они будут выполняться?

Практическая работа по работе с файловой системой Windows

Создайте файловую структуру по индивидуальному заданию преподавателя:

D:\\ PKS\\document.txt

D:\\Gruppa

Текстовый файл заполните информацией. Скопируйте (или перенесите) файл в папку Gruppa

**Контрольно-измерительные материалы для текущего контроля   
по МДК 01.02 Прикладное программирование**

Самостоятельная работа по теме «Программное обеспечение»

|  |  |
| --- | --- |
| **1 вариант**  1. Программа – это…  2. Технология программирования – это…  3. Системное ПО – это…  4. Назвать 4 вида программ, относящихся к программам специального назначения.  5. Охарактеризовать вид пользователей «прикладные программисты» | **2 вариант**  1. Программное обеспечение – это…  2. Программирование – это…  3. Прикладное ПО – это…  4. Назвать 4 вида программ, относящихся к программам общего назначения.  5. Охарактеризовать вид пользователей «системные программисты» |

|  |  |
| --- | --- |
| **Самостоятельная работа**  **«Работа с элементами выбора»**  **1 вариант**   1. Какое назначение компонента Флажок. 2. Запишите название компонента Переключатель 3. Напишите название свойства, определяющее установлен или сброшен флажок 4. Опишите свойство AllowGrayed для CheckBox. 5. Какие компоненты можно использовать, если в программе требуется создать несколько групп переключателей? В чем их отличие? | **Самостоятельная работа**  **«Работа с элементами выбора»**  **2 вариант**   1. Какое назначение компонента Переключатель. 2. Запишите название компонента Флажок 3. Напишите название свойства, определяющее установлен или сброшен переключатель 4. Опишите свойство Items для RadioGroup. 5. Какие компоненты можно использовать, если в программе требуется создать несколько групп переключателей? В чем их отличие? |

Самостоятельная работа «Обработка одномерного массива»

**Вариант 1**

Массив А из 12 элементов заполнить целыми случайными числами из интервала ( -13,20). Полученный массив распечатать. Найти сумму значений первой половины. Подсчитать количество значений> 1 (блок-схема и программа)

**Вариант 2**

Массив В из 15 элементов заполнить целыми случайными числами из интервала ( -5, 21). Полученный массив распечатать. Найти произведение последних 7 значений. Найти сумму значений < 12. (блок-схема и программа)

**Вариант 3**

Массив С из 13 элементов заполнить целыми случайными числами из интервала ( -1 , 17). Полученный массив распечатать. Найти сумму значений, взятых с четных мест. Найти произведение значений >6 (блок-схема и программа)

**Вариант 4**

Массив D из 22 элементов заполнить целыми случайными числами из интервала ( 2, 30). Полученный массив распечатать. Найти произведение значений, взятых с нечетных мест. Подсчитать количество значений < 21. (блок-схема и программа)

Самостоятельная работа «Поиск максимального и минимального одномерного массива»

**Вариант 1**

Массив А из 27 элементов заполнить целыми случайными числами из интервала ( -5, 17). Полученный массив распечатать. Найти максимальное значение и его порядковый номер в массиве, Найти минимальное значение и его номер в первой трети массива.

**Вариант 2**

Массив В из 35 элементов заполнить целыми случайными числами из интервала (-15, 35). Полученный массив распечатать. Найти минимальное значение и его порядковый номер в массиве. Найти максимальное значение и его номер среди значений, стоящих на четных местах.

**Вариант 3**

Массив С из 30 элементов заполнить целыми случайными числами из интервала (-10, 8). Полученный массив распечатать. Найти максимальное значение и его порядковый номер в массиве. Найти минимальное значение и его порядковый, номер во второй половине массива.

**Вариант 4**

Массив D из 27 элементов заполнить целыми случайными числами из интервала (-2, 25). Полученный массив распечатать. Найти минимальное значение и его порядковый номер в массиве. Найти максимальное значение и его номер среди значений, стоящих на нечетных местах.

Самостоятельная работа «Обработка двумерного массива»

|  |  |
| --- | --- |
| Вариант 3   1. Дана матрица размера *МхN.* Для каждой строки матрицы найти сумму ее элементов. 2. Дана матрица размера *МхМ.* Найти среднее арифметическое элементов побочной диагонали. | Вариант 4   1. Дана матрица размера МхN. Для каждого столбца матрицы найти произведение его элементов. 2. Дана матрица размера NхN. Найти минимальный среди элементов главной диагонали. |
| Вариант 5   1. Дана матрица размера МхN. Для каждой строки матрицы с нечет­ным номером (1, 3, …) найти среднее арифметическое ее элементов. Условный оператор не использовать. 2. Дана матрица размера NхN. Найти минимальный среди элементов побочной диагонали. | Вариант 6   1. Дана матрица размерам МхN. Для каждого столбца матрицы с четным номером (2, 4, …) найти сумму его элементов. Условный оператор не ис­пользовать. 2. Дана матрица размера МхМ. Найти среднее арифметическое элементов главной диагонали. |
| Вариант 7   1. Дана матрица размера МхN . В каждой ее строке найти количество элементов, меньших среднего арифметического всех элементов этой стро­ки. 2. Дана матрица размера М х N. Найти номер ее столбца с наименьшим произведением элементов и вывести данный номер, а также значение наи­меньшего произведения. | Вариант 8   1. Дана матрица размера МхN . В каждом ее столбце найти количество элементов, больших среднего арифметического всех элементов этого столбца. 2. Дана матрица размера Мх N . Найти номер ее строки с наибольшей суммой элементов и вывести данный номер, а также значение наибольшей суммы. |
| Вариант 9   1. Дана целочисленная матрица размера **Мх***N.* Найти номер первого из ее столбцов, содержащих только нечетные числа. Если таких столбцов нет, то вывести 0. 2. Дана матрица размера **Мх***N.* Найти максимальный среди минимальных элементов ее строк. | Вариант 10   1. Дана целочисленная матрица размера **Мх***N.* Найти номер последней из ее строк, содержащих только четные числа. Если таких строк нет, то вывести 0. 2. Дана матрица размера **Мх***N.* Найти минимальный среди максимальных элементов ее столбцов. |
| Вариант 1   1. Дана целочисленная матрица размера **Мх***N.* Найти номер последнего из ее столбцов, содержащих равное количество положительных и отрица­тельных элементов (нулевые элементы матрицы не учитываются). Если та­ких столбцов нет, то вывести 0. 2. Дана квадратная матрица***А***порядка ***М****.* Повернуть ее на угол 90° в отрицательном направлении, то есть по часовой стрелке. Вспомогательную матрицы не использовать. | Вариант 2   1. Дана целочисленная матрица размера МхN. Найти номер первой из ее строк, содержащих равное количество положительных и отрицательных элементов (нулевые элементы матрицы не учитываются). Если таких строк нет, то вывести 0. 2. Дана квадратная матрица А порядка М. Повернуть ее на угол 90° в положительном направлении, то есть против часовой стрелки Вспомогательную матрицу не использовать. |

Практические работы «Работа с компонентомListBox»

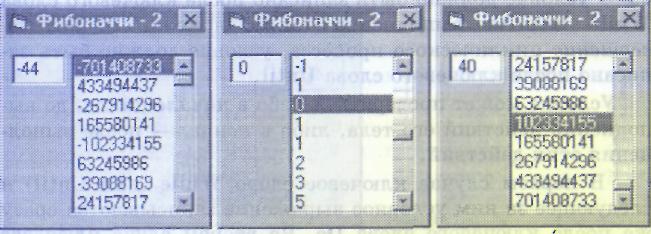
1. **Приложение «Фибоначчи-1».**Напечатайте в объекте ListBox числа из ряда Фибоначчи. В этом ряду каждое число, начиная с третьего, является суммой двух предыдущих чисел (1, 1, 2, 3,5,8,13,21,…). Ограничить значения номером 45.

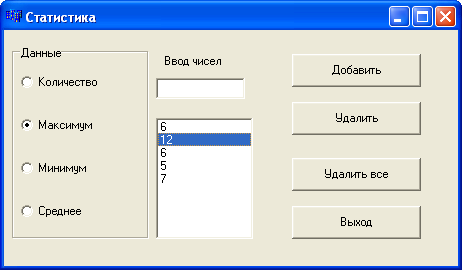
Номер выбранного числа из списка должен появиться в текстовом поле.

1. **Приложение «Фибоначчи-2 »**Обобщим понятие ряда Фибоначчи, – будем считать, что нумерация чисел начинается не 1, а с произвольного числа, например, -7

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер числа | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 |  |
| Число | 13 | -8 | 5 | -3 | 2 | -1 | 1 | 0 | 1 | 1 |  |

Числа с положительными номерами должны быть такими же, что и в обычном ряду Фибоначчи. Напишите программу, которая должна работать следующим образом:

После её запуска должно появиться *окно ввода,* с помощью которого пользователь введёт число *N.* Вслед за вводом числа программа создаст *обобщённый* ряд Фибоначчи, состоящий изКчисел (К = 2N + 1, от -Nдо N). Числа должны быть помещены в объект ListBox,а щелчок мышью любого элемента списка должен вызывать появление номера числа Фибоначчи в текстовом поле.

1. **Приложение "Статистика".** Приложение позволяет добавлять числовые данные в список при помощи текстового поля, удалять элементы списка, а также обрабатывать имеющиеся значения, вычисляя для них один из следующих параметров, определяемый состоянием переключателей:

* количество введенных чисел;
* максимальный элемент;
* минимальный элемент;
* среднее значение для введенных чисел.

1. **Приложение «Рост вклада в банке**». Допустим, некто поместил в банк S рублей (например, S — 200000). За год этот банк увеличивал сумму помещённых в него денег на Р % (например, Р = 10). Напишите программу, которая определит, за какой срок сумма вложенных денег увеличится не менее чем в kраз (например, k = 5, то есть сумма до­стигнет миллиона рублей). Для задания исходных данных *(S, P, k)* и выдачи результата используйте текстовые поля и список.

**Лабораторная работа №1 по теме «Модули»**

## Дисциплина: Прикладное программирование

**Тема:** Модульное программирование

**Цель:** Получить представление об создании программ с использованием модулей.

**Задание:** Используя модуль, составить программу, которая перемножает матрицы A(nxn), B(nxn). Для полученной матрицы вывести на экран нижнюю треугольную матрицу.

|  |  |
| --- | --- |
| **1. Созданиемодуля**  **Unit matric;**  **Interface**  Const m=100;  Vari,j:integer; a,b,c: array [1..m,1..m]of integer;  Procedure vvod (n:integer);  Procedure umn (n:integer);  Procedure vivod (n:integer);  **Implementation**  Procedure vvod(n:integer);  Vari,j: integer;  Begin  For i:=1 to n do  For j:=1 to n do begin  a[i,j]:=random(15)-3;  b[i,j]:=random(20)-5;  end;  end;  Procedure umn (n:integer);  Vari,j,k,s: integer;  Begin  For i:=1 to n do  For j:=1 to n do begin  s:=0;  for k:=1 to n do s:=s+a[i,k]\*b[k,j];  c[I,j]:=s;  end;  end;  Procedure vivod (n:integer);  Vari,j: integer;  Begin  Writeln;  Writeln (‘massiv A’);  For i:=1 to n do begin  Writeln;  For j:=1 to n do Write (a[i,j]:4);  end;  Writeln;  Writeln (‘massiv B’);  For i:=1 to n do begin  Writeln;  For j:=1 to n do Write (b[i,j]:4);  end;  Writeln;  Writeln (‘massiv C’);  For i:=1 to n do begin  Writeln;  For j:=1 to i do Write (c[i,j]:4);  end;  readln  end;**end.** | **2. Созданиепрограммы**  Program primer;  Uses **matric**;  Var n: integer;  Begin  Write (‘vveditekolichestvoelementovmassiva’);  Readln (n);  Vvod (n);  Umn (n);  Vivod (n);  End.  Для создания на диске файла с именем **matric.tpu** нужно выполнить компиляцию этого модуля (клавиши **Alt+F9**), предварительно установив опцию **Compile/Distination/Disk**. |

**Лабораторная работа №2 по теме «Модули»**

## Дисциплина: Прикладное программирование

Ввод матриц

Умножение матриц

Вывод матриц

Выход

Окно вывода матриц

(появляется только при выборе третьего пункта меню)

**Тема:** Модульное программирование

**Цель:** научиться создавать программы с использованием модулей.

**Задание:**

1. Создать меню, позволяющее работать с двумерными массивами (на основе модуля **matric**). Выбор пунктов меню осуществляется клавишами управления курсора (↑,↓) и клавишей <Enter>. Курсор должен двигаться по кругу, то есть переходить от последнего элемента к первому и наоборот.
2. Выполнить индивидуальное задание.

program menu;

usescrt, **matric;**

vari,j,iq,n:integer;

tx: array [1..4] of string[35];

c:char;

**proceduretextmenu;**

begin

tx[1]:='vvod matric';

tx[2]:='umnogenie matric';

tx[3]:='pechat matric';

tx[4]:='vihod';

end;

**procedureokno (x1,y1,x2,y2:integer;cv,phon:byte);**

begin

textbackground(phon);

textcolor(cv);

window(x1,y1,x2,y2);

clrscr;

end;

**procedurestartmenu(x:integer);**

begin

textbackground(13);

window(1,1,80,25);

clrscr;

i:=1;

okno(2,20+i,35,20+i,yellow,red);

write(tx[i]);

for i:=2 to x do begin

okno(2,20+i,35,20+i,black, lightgray);

write(tx[i]);

end;

end;

**procedure cursor(iq,i:integer);**

begin

okno(2,20+iq,35,20+iq,black, lightgray);

write(tx[iq]);

okno(2,20+i,35,20+i,yellow,red);

write(tx[i]);

end;

**procedurepech (n:integer);**

begin

okno(30,5,70,5+3+n\*3,0,2);

vivod(n);

readln;

okno(30,5,70,5+3+n\*3,yellow,13);

end;

**begin**

textbackground(13);

clrscr;

randomize;

writeln('kol-voelementovmassiva');

readln(n);

textmenu;

startmenu(4);

i:=1;

repeat

c:=readkey;

if c=#0 then c:=readkey;

case c of

#72: begin

iq:=i;

i:=i-1;

if i=0 then i:=4;

cursor(iq,i);

end;

#80: begin

iq:=i;

i:=i+1;

if i=5 then i:=1;

cursor(iq,i);

end;

#13: case i of

1: vvod(n);

2: umn(n);

3: pech(n);

4: exit;

end; end;

until false;

readln;

**end.**

**Лабораторная работа «Использование системы DragandDrop»**

**Цель работы:** познакомить с возможностью использования .системы *DragandDrop* для списков.

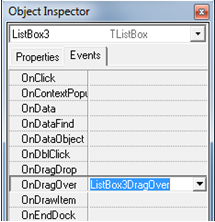
**Задание:**

1. разработать программу «Любимые напитки», используя рекомендации к выполнению;
2. переписать в тетрадь программный код, связанный с системой DragandDrop (пункт 3);
3. оформить лабораторную работу и сдать преподавателю.

**Спецификация программы:**

На пользовательской форме должны быть расположены 3 списка. Один из списков содержит названия напитков. Пользователь может переносить элементы из этого списка в списки «Любимые» и «Нелюбимые» и обратно. При этом перемещаемый элемент должен удалятся из списка-источника. Таким образом общее количество элементов всех трех списков остается постоянным. Перенос элементов между списками должен осуществляться по нажатию на соответствующие кнопки, либо мышью (система DragandDrop).

Система DragandDrop позволяет напрямую перетаскивать объекты между разными источниками, например, из одного списка в другой. «Перетаскивание» представляет собой нажатие и удерживание левой кнопки мыши на объекте и дальнейшее его перемещение за курсором в желаемую область.

Списки «Любимые» и «Нелюбимые» должны сохраняться в текстовые файлы.

**Рекомендации для выполнения лабораторной работы:**

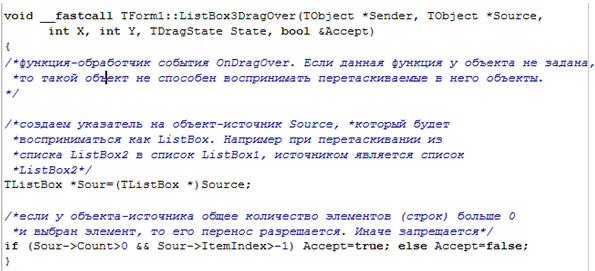
1. Написать код для кнопки переноса строки из списка ListBox2 в ListBox1 и для других кнопок.
2. Сохранить список ListBox1 в файле. Пример кода кнопки сохранения списка ListBox1 в файле:

http://informatics.ssga.ru/_/rsrc/1322739916559/app-inform/pi-practics/pi-practics-lab-8/10.png

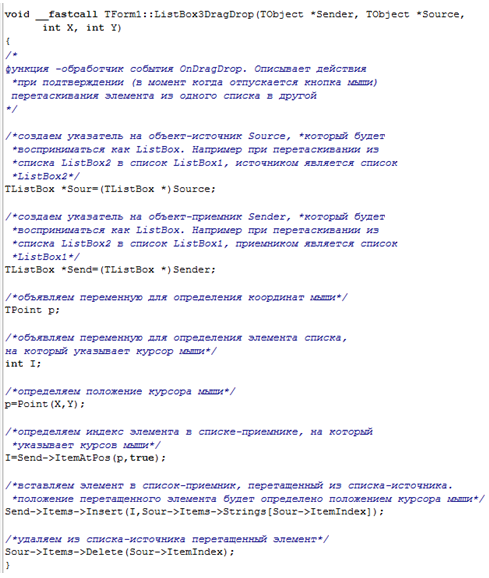
По аналогии написать код для сохранения списка ListBox3.

1. Перенос элементов между списками с помощью мыши можно организовать следующим образом:
2. Для каждого списка задать свойству DragMode значение dmAutomatic.
3. Для списка ListBox3 назначить обработчик событие OnDragOver. Для этого нужно выбрать список ListBox3 и в окне ObjectInspector на вкладке Events найти событие OnDragOver и дважды щелкнуть по его полю.

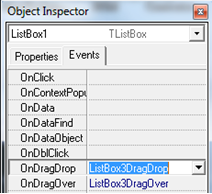
При этом автоматически будет создана заготовка для функции-обработчика. В тело этой функции следует записать следующий код:



1. Для этого же списка ListBox3 назначить обработчик событие OnDragDrop. Для этого нужно создать заготовку функции-обработчика события OnDragDrop и в нее записать ниже приведенный код:

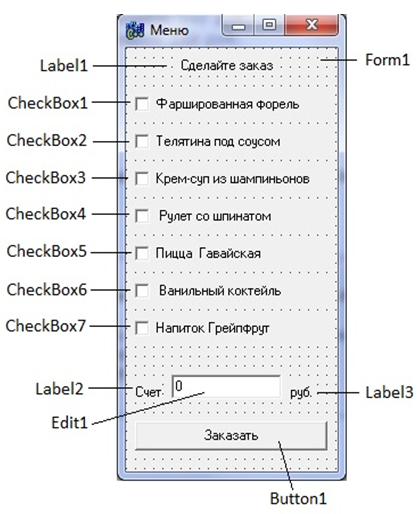


1. Для списков ListBox2 и ListBox1 в обработчиках событий OnDragOver  и OnDragDrop выбрать из выпадающих списков уже готовые функции ListBox3DragOver и ListBox3DragDrop:



1. Сохранить проект нажатием кнопки http://informatics.ssga.ru/_/rsrc/1322971224380/app-inform/pi-practics/pi-practics-lab-8/203.pngна панели инструментов.
2. Провести отладку и тестирование программы

**Задание 1. Разработка приложения "Меню"**

Напишите программу расчета стоимости электронного заказа по меню. Список блюд выполните с помощью компонента CheckBox. При выборе какого-либо блюда должна автоматически высчитываться стоимость заказа (при установлении флажка – счет должен увеличиваться на стоимость блюда, при снятии – уменьшаться на соответствующее значение). При нажатии на кнопку «Заказать» должно появляться сообщение (в виде диалогового окна ) об успешном заказе с общим счетом заказа.

***Спецификация программы:***

1. Создайте форму по образцу.
2. http://informatics.ssga.ru/_/rsrc/1322713155630/app-inform/pi-practics/pi-practics-lab-5/8.pngПоскольку объект Edit1 необходим только для вывода результата (то есть заполняться он будет по нажатию на кнопку, а не вручную), нужно изменить его свойство ReadOnly  на значение true (то есть сделать объект Edit1 только для чтения).
3. Каждое блюдо из меню должно иметь свою стоимость. Допустим, цены на блюда будут следующие:

Фаршированная форель 369 руб.

Телятина под соусом 349 руб.

Крем-суп из шампиньонов 189 руб.

Рулет со шпинатом 179 руб.

Пицца Гавайская 299 руб.

Ванильный коктейль 149 руб.

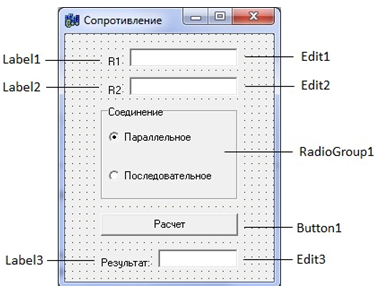
Напиток Грейпфрут 199 руб.

Стоимость каждого блюда должна быть записана в массив price, который должен выглядеть следующим образом:

intprice[7]={369, 349,189,179,299,149,199};

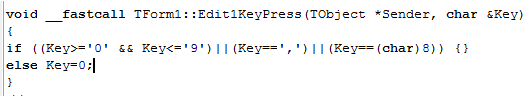
1. Напишите программный код программы.

**Задание 2. Разработка приложения «Расчет сопротивления»**

Напишите программу для расчета сопротивления резисторов, соединенных параллельно или последовательно.

***Спецификация программы:***

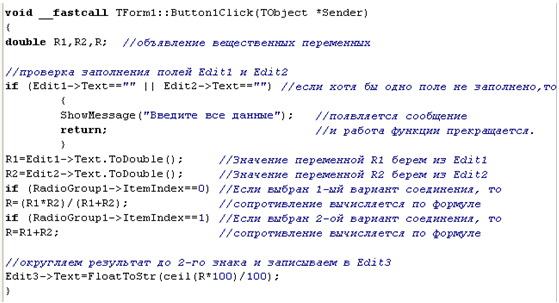
1. Создайте форму по образцу.
2. Измените свойство ReadOnly=True объекта Edit3, чтобы сделать объект Edit3 только для чтения.
3. Объекты Edit1 и Edit2 предназначены для ввода числовых данных, необходимых для расчета сопротивления. Для того чтобы пользователь не смог ввести буквы или другие символы, которые могу вызвать ошибку при расчетах, необходимо сделать обработчик нажатий клавиш на клавиатуре. Для этого необходимо выбрать объект Edit1 и в окне свойств объекта открыть вкладку Events, найти событие OnKeyPress («по нажатию клавиши») и сделать двойной щелчок левой кнопкой мыши по пустому полю. В появившейся заготовке функции написать код обработки нажатия клавиш:



Этим самым все клавиши стали отфильтрованы на клавиатуре, кроме цифровых от 0 до 9, запятой и BackSpase (код клавиши 8). Теперь нажатие ненужных клавиш будет игнорироваться.

http://informatics.ssga.ru/_/rsrc/1322140211602/practics/lab-11/31.pngДля объектов Edit2 на обработку события OnKeyPress следует подключить уже созданную функцию Edit1KeyPress. Для этого необходимо выбрать  эту функцию в выпадающем списке  события OnKeyPress

1. Сопротивление для последовательного соединения вычисляется по формуле R=R1+R2; а для параллельного R=(R1\*R2)/(R1+R2).
2. Для реализации расчета в программе необходимо добавить обработчик нажатия кнопки Button1. Для этого надо дважды щелкнуть по объекту Button1 и в появившейся заготовке функции написать программный код расчета общего сопротивления. Перед расчетами необходимо проверить текстовые поля Edit1 и Edit2 на заполняемость.

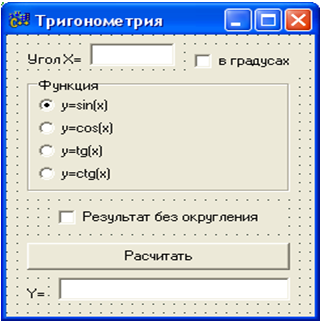


1. Допишите программу самостоятельно.

**Задание 3. Разработка приложения «Тригонометрия»**

Напишите программу расчета значений тригонометрических функций для угла X.

***Спецификация программы:***

1. Создайте форму по образцу.
2. Угол может быть введен как в радианах (по умолчанию), так и в градусах. Для указания того, что введенный угол в градусах на форму поместить переключатель CheckBox.
3. Выбор тригонометрической функции должен осуществляться с помощью объекта RadioGroup.
4. По умолчанию программа должна выдавать результат, округленный до четвертого знака. Для возможности получения неокругленного результата использовать переключатель CheckBox.
5. Поле «Угол X=» сделать с проверкой вводимых данных для исключения ввода нечисловых данных.
6. Программа должна выдавать сообщение об ошибке, если поле «Угол X=» не заполнено.
7. Поле «Y=» должно быть только для чтения

## Практическая работа: «Работа с несколькими формами»

## Задание 1. Колобки

## Осуществить переход между формами по щелчку.

## Для того, чтобы добавить новую форму, необходимо выполнить команду File/New/Form.

## Тогда в разделе программного кода появится модуль Unit2, предназначенный для второй формы.

## Запуск программы будет по умолчанию осуществляться с запуска первой формы. Чтобы осуществить переход между формами по щелчку, необходимо запрограммировать процедуры.

## Задание 2.Загадка.

## Создайте три формы. После запуска приложения на первой форме видны 2 кнопки: «Загадка» и «Выход». После нажатия на кнопку «Загадка» появляется вторая форма с текстом загадки и кнопками «Отгадка» и «Выход». После нажатия на кнопку «Отгадка» появляется третья форма с рисунком и текстом отгадки и кнопкой «Выход».

## 

## Задание 3. День и ночь

## Создайте проект, состоящий из двух форм. На форме1 должен быть показан день, с движением солнца, а на форме2 – ночь, с движением луны. Все переходы должны осуществляться по времени (объекты Timer1 и Timer2). Необходимо останавливать и вновь запускать таймеры при открытии новой формы (свойство Enabled).

## 

## 

## Установите для компонентов Form1, Form2 свойства: ClientHeight=500, Client=600. На каждой форме разместите 2 таймера. На Form1 для Timer1 установите свойство Interval=500, а для Timer2-10000. На Form2 для Timer1-10000, для Timer2-500.

## Разместите на формах 2 объекта Shape (небо и земля) и установите для свойства: Height=250, Width=600.

## Для луны используйте два круга.

## Unit1.cpp

## #include <vcl.h>

## #pragma hdrstop

## #include "Unit2.h"

## #include "Unit1.h"

## #include "math.h"

## #pragma package(smart\_init)

## #pragma resource "\*.dfm"

## TForm1 \*Form1;

## \_\_fastcall TForm1::TForm1(TComponent\* Owner)

## : TForm(Owner)

## {

## }

## void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)

## {

## Shape1->Left=Shape1->Left+25;

## Shape1->Top=floor(-sqrt(250\*250-pow(Shape1 ->Left-250,2))+250);

## }

## void \_\_fastcall TForm1::Timer2Timer(TObject \*Sender)

## {

## Shape1->Left=0;

## Shape1->Top=250;

## Timer1->Enabled=False;

## Timer2->Enabled=False;

## Form1->Hide();

## Form2->Show();

## Form2->Timer1->Enabled=true;

## Form2->Timer2->Enabled=true;

## }

## Unit2.cpp

## #include <vcl.h>

## #pragma hdrstop

## #include "Unit1.h"

## #include "Unit2.h"

## #include "math.h"

## #pragma package(smart\_init)

## #pragma resource "\*.dfm"

## TForm2 \*Form2;

## \_\_fastcall TForm2::TForm2(TComponent\* Owner)

## : TForm(Owner)

## {

## }

## void \_\_fastcall TForm2::Timer1Timer(TObject \*Sender)

## {

## Shape1->Left=0;

## Shape1->Top=250;

## Shape2->Left=25;

## Shape2->Top=250;

## Timer1->Enabled=false;

## Timer2->Enabled=false;

## Form1->Show();

## Form2->Hide();

## Form2->Timer1->Enabled=false;

## Form2->Timer2->Enabled=false;

## Form1->Timer1->Enabled=true;

## Form1->Timer2->Enabled=true;

## }

## void \_\_fastcall TForm2::Timer2Timer(TObject \*Sender)

## {

## Shape1->Left=Shape1->Left+ 25;

## Shape1->Top=floor(-sqrt(250\*250-pow(Shape1 ->Left-250,2))+250);

## Shape2->Left=Shape1->Left+ 25;

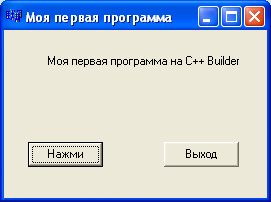
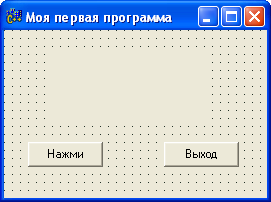
## Shape2->Top=floor(-sqrt(250\*250-pow(Shape1 ->Left-250,2))+250);

## }

***Практическая работа «Первые задачи на С++ Builder»***

1. Проект «Моя первая программа».

Создайте приложение, в котором при нажатии на кнопку «Нажми» на форме в объекте метка выводится сообщение “Моя первая программа на C++ Builder”.



Форма имеет вид: После запуска программы:

Чтобы при нажатии на кнопку «Выход» происходило закрытие формы, необходимо в программном коде для этой кнопки прописать команду Form1->Close();

Замечание: заглавные/маленькие буквы, знаки препинания, пустые скобки имеют значение

1. Создать проект «Вывод сообщения»

* создать форму «Вывод сообщения»;
* разместить на форме объект метка (Label1), разместить на форме объект текстовое поле (Edit1);
* разместить на форме объект командная кнопка (Button1) и создать программный код, который выводит в объекты метка и текстовое поле сообщение «Второе задание выполнено!»
* Для объекта метка установить шрифт «TimesNewRoman, 14, полужирный, красный».
* Для объекта текстовое поле установить шрифт «CourierNew, 16, курсив, зеленый»

1. Создать проект «Привет», в котором с помощью метки на форме печатается текстовое сообщение “Привет!” при нажатии на кнопку “Пуск”. Исправьте программу так, чтобы каждое новое нажатие добавляло в метку слова через пробел, а не заменяло их (сколько раз нажали кнопку, столько слов «Привет» отобразилось в метке).
2. Создать проект «Чистый привет», в котором одиночный щелчок по форме выводит в метке ваше приветствие, а двойной щелчок удаляет его.
3. Создать проект "Вывод сообщений в текстовое поле", в котором два различных варианта текста выводятся в поля ввода (Edit) по щелчку по одной из двух кнопок соответственно.
4. Создайте проект «Кнопки», в котором на форме разместите три кнопки с номерами «1», «2» и «3» и одну метку. При нажатии одной из них о произошедшем событии выводится надпись, например «Кнопка : 1»
5. Создать проект, в котором при одинарном щелчке по красному полю (объект Метка) на этом поле появляется надпись желтого цвета «Один щелчок», при двойном должна появиться надпись желтого цвета «Двойной щелчок».
6. Создать проект «Привет с салютом», в котором создайте 2 формы и по щелчку по ним выводите сообщения «Привет!» и «Салют!» соответственно. Двойной щелчок меняет формы на экране. Предусмотрите очистку форм при их смене.
7. Составьте программу, позволяющую в процессе выполнения автоматически переименовывать окно. Новый заголовок вводится в текстовое поле. Окно переименовывается в результате одинарного щелчка по кнопке.
8. Создать проект "Открой букву", в котором разместите на форме ряд объектов Label, в каждом из которых находится буква загаданного слова. Установите одинаковые значения цвета фона и текста, чтобы задуманное слово было невидимым. Щелчок по каждому объекту Label должен изменять цвет буквы, открывая таким образом соответствующую букву.
9. Составьте программу для пересчета значений температуры из шкалы Цельсия в шкалу Фаренгейта и обратно. C=(F-32)\*5/9; F=C(9/5)+32. Здесь C – значение температуры в градусах Цельсия, F – в градусах Фаренгейта.
10. Создать проект «Печать на форме», в котором с помощью меток на форме печатаются различные строки шрифтами со следующими параметрами:

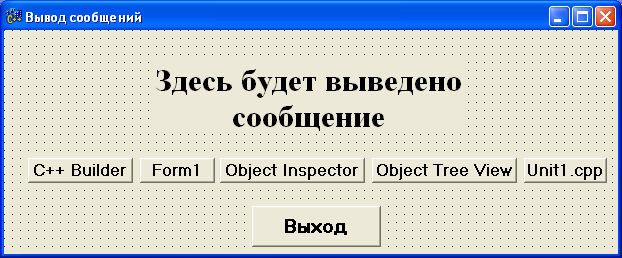
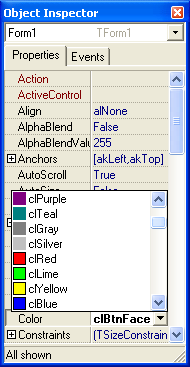
* Times New Roman, 18, курсив, красный;
* Arial, 14, подчеркнутый, синий;
* CourierNew, 12, полужирный, зеленый.

Предусмотреть очистку формы от напечатанного текста.

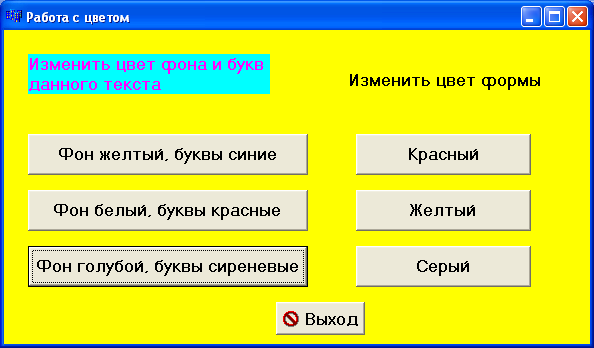
**1. Вывод сообщений**

Создать проект, в результате которого в текстовом поле выводится одно из сообщений в зависимости от того, на какой кнопке пользователь щелкнул мышью. Установите перенос слов, не умещающихся на строке (свойство WordWrap для метки должно иметь значение true).

|  |  |
| --- | --- |
| **Наименование кнопки** | **Отображаемый текст** |
| **С++ Builder** | **Объектно-ориентированный язык программирования** |
| **Form1** | **Заготовка главного окна разрабатываемого приложения** |
| **Object Inspector** | **Окно редактора свойств объекта** |
| **Object Tree View** | **Окно просмотра списка объектов** |
| **Unit1.cpp** | **Окно редактора кодов объекта** |



**2. Работа с цветом**

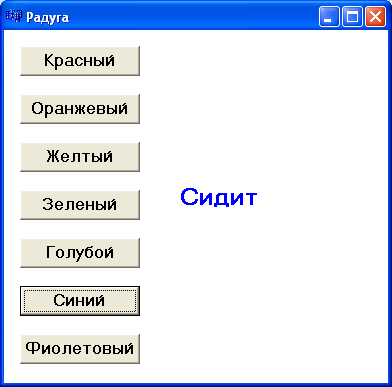
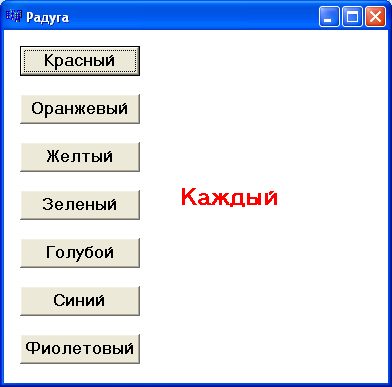
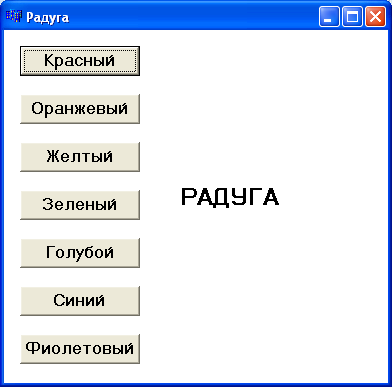


Создать проект, в результате которого можно изменять цвет различных объектов, сообщений (надписей) на них.

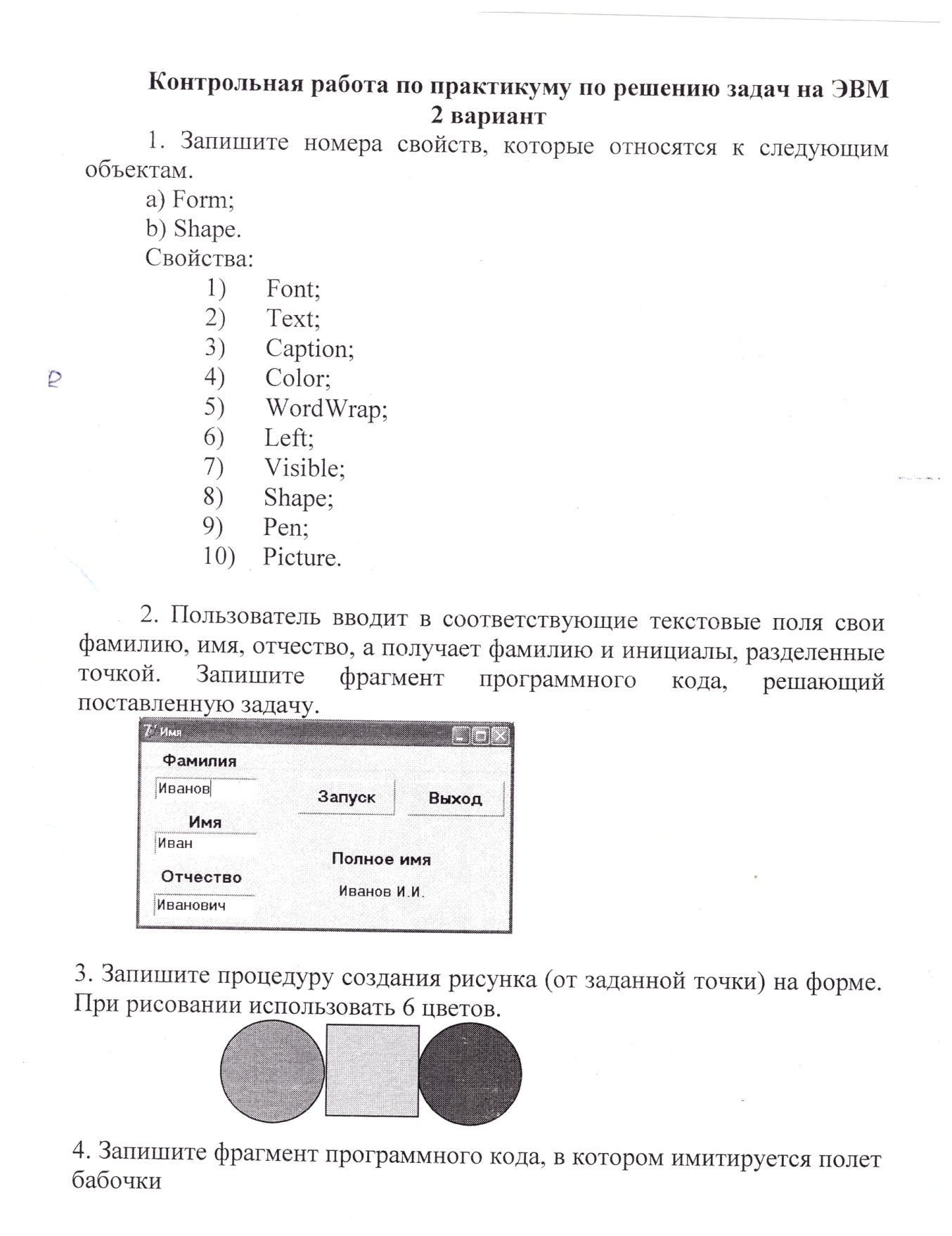
Замечание. Цвет записывается своим английским названием, но впереди ставится приставка «cl», например: clBlack — черный; clMaroon — каштановый; clGreen — зеленый; clOlive — оливковый; clNavy— темно-синий; clPurple — розовый; clTeal — зелено-голубой; clGray—серый; ciSilver — серебристый; clRed — красный; clLime — салатный; clBlue — синий; clFuchsia - ярко-розовый; clAqua — бирюзовый;clWhite - Название цвета можно узнать в свойстве Color, например объекта Form.

**3. Радуга**

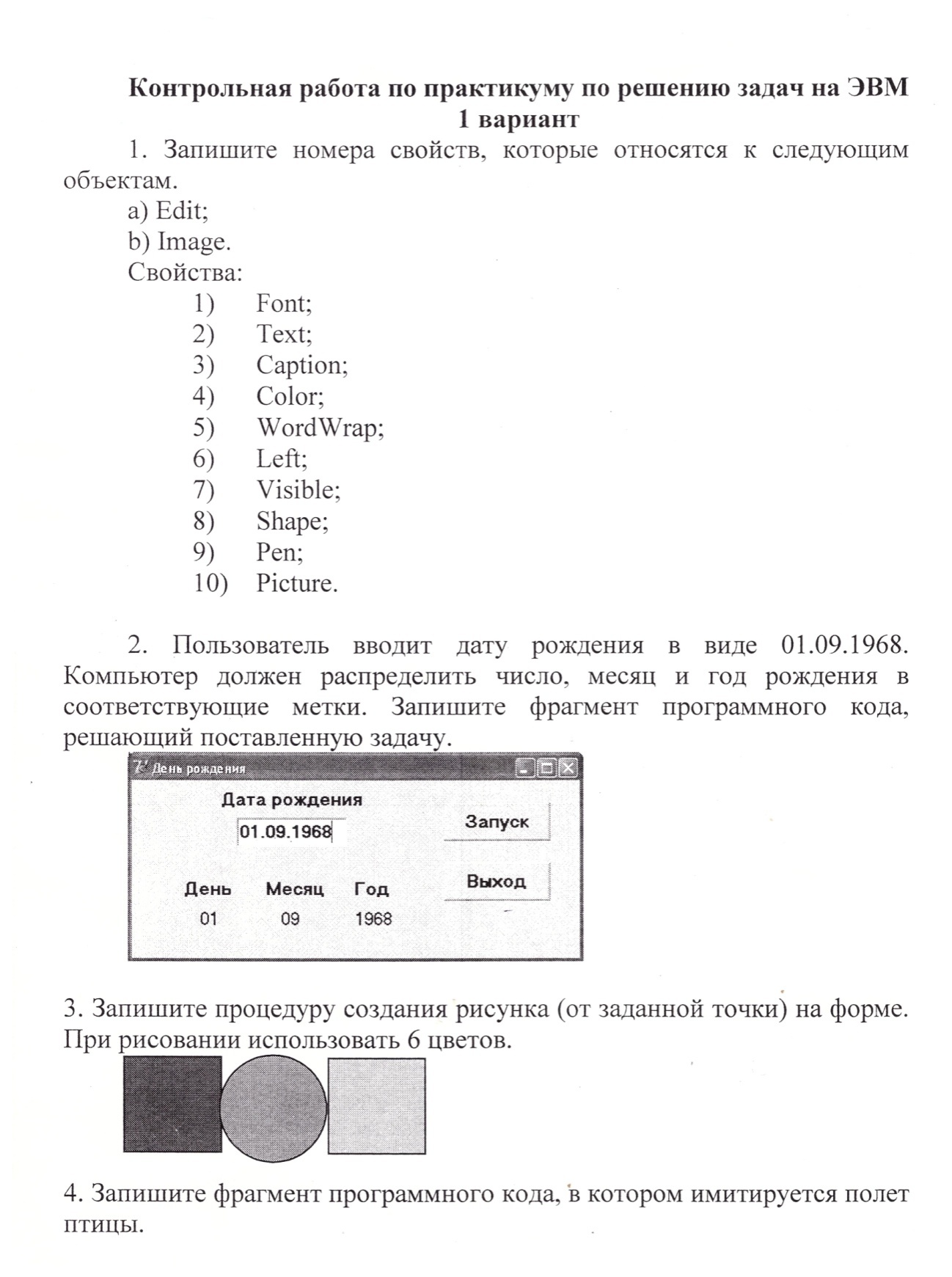
Разместите на форме 7 командных кнопок, надписи которых соответствуют названиям цветов радуги. При нажатии кнопки на текстовом поле, расположенном справа, выводить соответствующее слово считалочки: «Каждый охотник желает знать, где сидит фазан» выбранным цветом



**Контрольная работа «Линейные алгоритмы»**



Контрольная работа «Линейные алгоритмы»



1. [↑](#footnote-ref-1)