

Приложение ППССЗ/ППКРС по специальности 09.02.07 Информационные системы и программирование 2023-2024 уч.г.: Комплект контрольно-оценочных средств междисциплинарного курса МДК.02.02 Инструментальные средства разработки программного обеспечения

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ БЕЛГОРОДСКОЙ ОБЛАСТИ
ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«АЛЕКСЕЕВСКИЙ КОЛЛЕДЖ»**

**Комплект
контрольно-оценочных средств
междисциплинарного курса
МДК.02.02 Инструментальные средства разработки
программного обеспечения
для специальности
09.02.07 Информационные системы и программирование**

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утвержденный приказом Министерства образования и науки Российской Федерации от 9 декабря 2016 года № 1547, с учетом профессионального стандарта «Специалист по информационным системам», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2014 года № 896н.

Составитель:

Дешина И.А., преподаватель ОГАПОУ «Алексеевский колледж»

1. Паспорт комплекта оценочных средств

1.1 Область применения комплекта оценочных средств

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу МДК.02.02 Инструментальные средства разработки программного обеспечения.

КОС включают контрольные материалы для проведения промежуточной аттестации в форме дифференцированного зачета.

КОС разработан на основании рабочей программы МДК.02.02 Инструментальные средства разработки программного обеспечения.

1.2 Цели и задачи МДК – требования к результатам освоения МДК

С целью овладения указанным видом деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения МДК должен:

иметь практический опыт:

О1 интегрировать модули в программное обеспечение;

О2 отлаживать программные модули;

О3 инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

уметь:

У1 использовать выбранную систему контроля версий.

У2 использовать методы для получения кода с заданной функциональностью и степенью качества.

У3 организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.

У4 использовать различные транспортные протоколы и стандарты форматирования сообщений.

У5 выполнять тестирование интеграции.

У6 организовывать постобработку данных.

У7 создавать классы-исключения на основе базовых классов.

У8 выполнять ручное и автоматизированное тестирование программного модуля.

У9 выявлять ошибки в системных компонентах на основе спецификаций.

У10 использовать приемы работы в системах контроля версий

У11 анализировать проектную и техническую документацию.

У12 использовать инструментальные средства отладки программных продуктов.

У13 определять источники и приемники данных.

У14 выполнять отладку, используя методы и инструменты условной компиляции.

У15 приемы работы в системах контроля версий.

знать:

31 модели процесса разработки программного обеспечения.

32 основные принципы процесса разработки программного обеспечения.

33 основные подходы к интегрированию программных модулей.

34 основы верификации программного обеспечения.

35 современные технологии и инструменты интеграции.

36 основные протоколы доступа к данным.

37 методы и способы идентификации сбоев и ошибок при интеграции приложений.

38 основные методы отладки.

39 методы и схемы обработки исключительных ситуаций.

310 основные методы и виды тестирования программных продуктов.

311 стандарты качества программной документации.

312 основы организации инспектирования и верификации.

313 приемы работы с инструментальными средствами тестирования и отладки.

314 методы организации работы в команде разработчиков.

315 основы верификации и аттестации программного обеспечения.

316 встроенные и основные специализированные инструменты анализа качества программных продуктов.

Перечень знаний и умений в соответствии с профессиональными стандартами «Специалист по информационным системам», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2014 г. № 896н которые актуализируются при изучении междисциплинарного курса:

- 1) Собирать исходную документацию
- 2) Предметная область автоматизации
- 3) Инструменты и методы выявления требований
- 4) Современные стандарты информационного взаимодействия систем

Перечень знаний, умений, навыков в соответствии со спецификацией стандарта компетенции чемпионатного движения по профессиональному мастерству «Профессионалы» и Чемпионата высоких технологий Программные решения для бизнеса, которые актуализируются при изучении профессионального модуля: 1) знать и понимать: общие типы проблем и требований, которые могут возникнуть при разработке программного обеспечения;

2) знать и понимать: как настроить, разработать и интегрировать в разработанное решение новейшие технологии и оборудование, которые будут способствовать лучшему бизнес-решению.

3) знать и понимать: важность соблюдения стандартов (например, соглашения по формату кода, руководства по стилю, дизайна пользовательского интерфейса, управления каталогами и файлами)

4) знать и понимать: важность точного и постоянного контроля версий

5) знать и понимать: важность использования существующего кода в качестве основы для анализа и модификации.

1.3. Планируемые личностные результаты освоения рабочей программы

ЛР 1. Осознающий себя гражданином и защитником великой страны.

ЛР 5. Демонстрирующий приверженность к родной культуре, исторической памяти на основе любви к Родине, родному народу, малой родине, принятию традиционных ценностей многонационального народа России.

ЛР 6. Проявляющий уважение к людям старшего поколения и готовность к участию в социальной поддержке и волонтерских движениях.

ЛР 12. Принимающий семейные ценности, готовый к созданию семьи и воспитанию детей; демонстрирующий неприятие насилия в семье, ухода от родительской ответственности, отказа от отношений со своими детьми и их финансового содержания.

Результатом освоения МДК является овладение обучающимися видом деятельности - Осуществление интеграции программных модулей, в том числе профессиональными компетенциями (ПК):

Код	Наименование результата обучения
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе

	традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение.
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

1.3 Результаты освоения междисциплинарного курса, подлежащие проверке

Наименование тем	Коды компетенций (ОК, ПК), личностных результатов (ЛР), умений (У), знаний (З), формированию которых способствует элемент программы	Средства контроля и оценки результатов обучения в рамках текущей аттестации (номер задания)	Средства контроля и оценки результатов обучения в рамках промежуточной аттестации (номер задания/контрольного вопроса/ экзаменационного билета)
Тема 1.1. Современные технологии и инструменты интеграции	ПК 2.2 ЛР 5 ЛР 6	ПЗ № 6-9	КВ № 1-15
Тема 1.2. Инструментарий тестирования и анализа качества программных средств	ОК 03 ОК 09 ПК 2.3 ПК 2.5 ЛР 1 ЛР12	ПЗ № 23-26	КВ № 16-25

2. Комплект оценочных средств для текущей аттестации

2.1. Практические задания (ПЗ)

ПЗ № 6. Задание

1. Описать этапы проектирования модулей программы.
2. Составить в виде блок-схемы алгоритм решения задачи.
3. Описать набор тестов для отладки программы.

Задача. Необходимо распределить обязанности между участниками вашей команды и составить алгоритм решения задачи, приведенной ниже, с использованием структурных единиц: процедур и/или функций.

Варианты индивидуальных заданий (согласно номеру в журнале (работа в команде по 2 человека))

1,2. Даны два двумерных массива вещественных элементов. Размер исходных массивов не превосходит 10x10 элементов. Для каждого из массивов указать номера столбцов, содержащих только положительные элементы. Если таких столбцов в массиве нет, то вывести соответствующее сообщение. Проверку столбца на положительность элементов оформить в виде процедуры с передачей в нее всех элементов текущего столбца.

3,4. Даны два двумерных массива натуральных элементов. Размер исходных массивов не превосходит 10x10 элементов. Для каждого из массивов указать номера столбцов, содержащих только кратные 5 или 7 элементы. Если таких столбцов в массиве нет, то вывести соответствующее сообщение. Проверку столбца на наличие указанных элементов оформить в виде процедуры с передачей в нее всех элементов текущего столбца.

5,6. Даны пять одномерных массива вещественных элементов. Размер каждого массива не превосходит 100 элементов. Для каждого из массивов определить, составляют ли его элементы знакопередающуюся последовательность. Если да, то указать порядковый номер такого массива, в противном случае вывести отрицательный ответ. Проверку массива на выполнение условия оформить в виде процедуры с передачей в нее всех элементов рассматриваемого массива.

7,8. Даны два двумерных массива символьных (буквы русского алфавита) элементов. Размер исходных массивов не превосходит 10x10 элементов. Для каждого из массивов указать номера строк, содержащих элементы только строчных букв, если таких строк нет ни для какого массива, то вывести соответствующее сообщение. Проверку строки на наличие

указанных элементов оформить в виде процедуры с передачей в нее всех элементов текущей строки.

9,10. Даны два двумерных массива вещественных элементов. Размер исходных массивов не превосходит 10×10 элементов. Для каждого из массивов указать количество столбцов, содержащих только не положительные элементы. Если таких столбцов нет ни для одного из массивов, то вывести соответствующее сообщение. Проверку столбца на наличие указанных элементов оформить в виде процедуры с передачей в нее всех элементов текущего столбца.

11,12. Даны пять одномерных массива вещественных элементов. Размер каждого массива не превосходит 100 элементов. Для каждого из массивов определить, составляют ли его элементы одного знака. Если да, то указать порядковый номер такого массива, в противном случае вывести отрицательный ответ. Проверку массива на выполнение условия оформить в виде процедуры с передачей в нее всех элементов рассматриваемого массива.

13,14. Даны два двумерных массива целочисленных элементов. Размер исходных массивов не превосходит 10×10 элементов. Для каждого из массивов указать количество строк, содержащих элементы, четность которых чередуется, а вторым в четных строках является нечетный элемент. Если таких строк нет ни для одного из массивов, то вывести соответствующее сообщение. Проверку строки на наличие указанных элементов оформить в виде процедуры с передачей в нее всех элементов текущего столбца.

ПЗ № 7. Задание (с помощью любого ЯП)

1. Необходимо распределить обязанности между участниками вашей команды и разработать модули программы, спроектированные во время практического занятия №6.(текст программы на ЯП)
2. Отладить программу с использованием тестов, составленных во время практического занятия №6.(отчет о тестировании)

ПЗ № 8. Задание.

Согласно поставленной задаче выполните ручную отладку:

- Опишите математическую модель задачи с указанием имен и назначения переменных;
- Опишите спецификацию программы;
- Запишите алгоритм программы;

– Составьте тестовые наборы для проверки функционала системы.

Варианты заданий

Создать приложение, реализующие линейный и разветвляющийся алгоритмы, которые размещены на форме. При решении линейного алгоритма предусмотреть поля ввода значений переменных и поле вывода результата вычисления. При решении разветвляющегося алгоритма предусмотреть поля для ввода значений переменных, поле вывода результатов расчета по одной из трех формул в зависимости от результата выполнения условия. **В качестве $f(x)$ использовать по выбору: $\cos(x)$ или x^2 или e^x .** Пример рабочей формы представлен на рисунке 1.

The image shows two panels of a web application interface. The left panel displays the formula $u = \operatorname{tg}^5(\sqrt{x} - y^3) + e^{y/x} * \sin z^2$ and three input fields labeled 'Введите значение X:', 'Введите значение Y:', and 'Введите значение Z:'. Below these is a text area labeled 'В этом поле выводится результат'. The right panel displays a piecewise function for z : $z = \begin{cases} (f(x) + y)^2, & xy > 0 \\ f(x) * \cos y, & xy < 0 \\ f(x) + y, & xy = 0 \end{cases}$. It includes input fields for $X =$ and $Y =$, a dropdown menu for 'Выбор функции' with radio buttons for $\cos(x)$, $\operatorname{sqr}(x)$, and $\exp(x)$, and a text area labeled 'Здесь будет выводиться результат расчетов'.

Рисунок 1 – приложение

Линейный алгоритм (согласно журналу):

$$\boxed{1, 8} \quad t = \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y} \left(1 + \frac{z^2}{3 - \frac{z^2}{5}}\right).$$

$$\boxed{2, 9} \quad u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x - y|} * (\operatorname{tg}^2 z + 1)^x.$$

$$\boxed{3, 10} \quad v = \frac{1 + \sin^2(x + y)}{\left|x - \frac{2y}{1 + x^2 y^2}\right|} * x^{|y|} + \cos^2\left(\operatorname{arctg} \frac{1}{z}\right).$$

$$\boxed{4, 11} \quad w = |\cos x - \cos y|^{1 + 2 \sin^2 y} * \left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

$$\boxed{5, 12} \quad \alpha = \ln\left(y^{-\sqrt{|x|}}\right) * \left(x - \frac{y}{2}\right) + \sin^2 \operatorname{arctg}(z).$$

$$\boxed{6, 13} \quad \beta = \sqrt{10(\sqrt[3]{x} + x^{y+2})} * (\arcsin^2 z - |x - y|).$$

$$\boxed{7, 14} \quad \gamma = 5 \operatorname{arctg}(x) - \frac{1}{4} \arccos(x) * \frac{x+3|x-y|+x^2}{|x-y|z+x^2}.$$

Разветвляющийся алгоритм (согласно журналу):

$$\boxed{1, 8} \quad a = \begin{cases} \ln(y+2) + f(x), & x/y > 0 \\ \ln|y| - \operatorname{tg}(f(x)), & x/y < 0 \\ f(x) * y^3, & \text{иначе} \end{cases}$$

$$\boxed{2, 9} \quad b = \begin{cases} \ln|y + f(x)| + ,3 < xy < 8 \\ \cos(f(x)) - y, & xy > 12 \\ \operatorname{sh}(f(x) + \operatorname{cs}(y)), & \text{иначе} \end{cases}$$

$$\boxed{3, 10} \quad c = \begin{cases} f^3(x) + \operatorname{ctg}(y), & xy > 12 \\ \operatorname{sh}(f^3(x)) + y^2, & xy < 7 \\ \cos(x - f^3(x)), & \text{иначе} \end{cases}$$

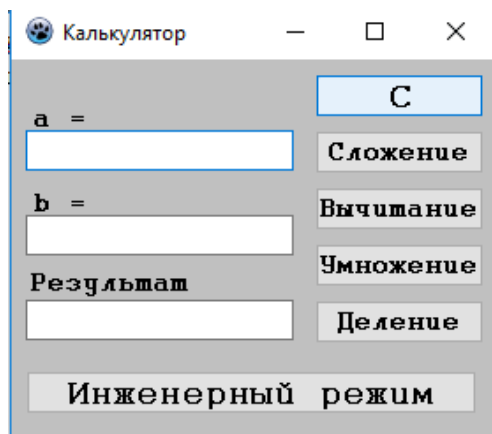
$$\boxed{4, 11} \quad d = \begin{cases} \operatorname{ctg}(y) + f(x), & x/y > 0 \\ \ln|y| + \operatorname{tg}(f(x)), & x/y < 0 \\ f(x) * y^3, & \text{иначе} \end{cases}$$

$$\boxed{5, 12} \quad k = \begin{cases} (f(x) + y)^2, & 4 > xy > 1 \\ f(x) * \operatorname{tg}(y), & 8 < xy < 10 \\ f(x) + y, & \text{иначе} \end{cases}$$

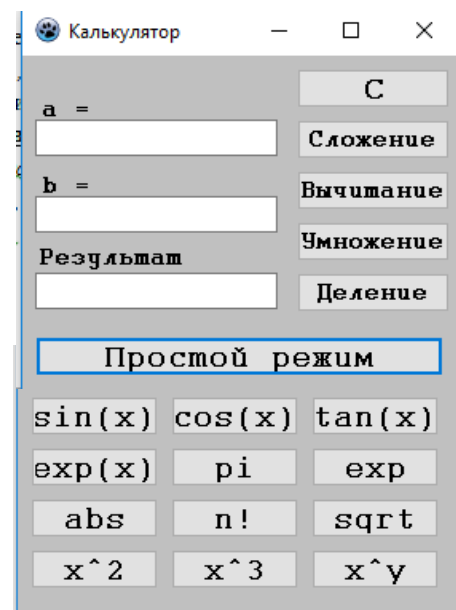
$$\boxed{6, 13} \quad l = \begin{cases} f^2(x) + \operatorname{arctg}(f(x)), & 1 \leq x < 5 \\ (y - f(x))^2 + \operatorname{arctg}(f(x)), & y > x \\ (y + f(x))^3 + 0.5, & \text{иначе} \end{cases}$$

$$\boxed{7, 14} \quad m = \begin{cases} f^3(x) + \sin(y), & xy < 5 \\ \operatorname{ch}(f^3(x)) + y^2, & xy > 7 \\ \cos(x + f^3(x)), & \text{иначе} \end{cases}$$

ПЗ № 9. Задание: Расширить функционал программы «Калькулятор»(рабочая программа предоставляется во вложении к ПЗ) путем введения дополнительных операций ($\sin(x)$, $\cos(x)$, $\operatorname{tg}(x)$, $\operatorname{ctg}(x)$, $\exp(x)$, abs , x^2 , x^3 , x^y , sqrt , π , $n!$). Интерфейс программы должен соответствовать образцу:



Простой вид калькулятора



Инженерный вид калькулятора

Выполнение задания

Создание обработчиков событий новых кнопок:

1. Обработчик события нажатия кнопки изменения режимов калькулятора имеет следующий вид:

```
procedure TForm1.Button16Click(Sender: TObject);
begin
  if Button16.Caption = 'Инженерный режим' then begin
    Button16.Caption := 'Простой режим';
    Form1.Height := 510;
  end
  else if Button16.Caption = 'Простой режим' then begin
    Button16.Caption := 'Инженерный режим';
    Form1.Height := 220;
  end;
end;
```

В этом обработчике событий мы, впервые в этом проекте, используем условия. Оно необходимо нам для определения текущего режима работы калькулятора и переключения в другой режим. Флагом, сигнализирующим нам о том, в каком режиме работает калькулятор, является свойство `Caption` данной кнопки, а точнее его значение. Если `Caption` имеет значение «Инженерный режим», то это значит, что в данный момент калькулятор находится в простом режиме и при нажатии кнопки «Инженерный режим» мы переключимся в расширенный режим калькулятора. Если же `Caption` имеет значение «Простой режим», то это значит, что в данный момент калькулятор находится в инженерном режиме и при нажатии кнопки «Простой режим» мы переключимся из расширенного режима калькулятора в упрощенный.

Сама процедура переключения режима калькулятора из простого в инженерный, представляет собой ни что иное, как программное изменение размеров формы, путем установки свойства `Height` формы в значение «510» при переключении в расширенный режим и в значение «220», при переключении в упрощенный режим. При этом мы меняем и наш флаг – с «Инженерный режим» на «Простой режим», при переключении в расширенный режим и с «Простой режим» на «Инженерный режим», при переключении в упрощенный режим работы.

2. Далее создадим обработчики событий нажатия кнопок, выполняющих операции: $\sin(x)$, $\cos(x)$, $\tan(x)$, $\exp(x)$, abs , x^2 , x^3 , sqrt . **В качестве примера рассмотрим листинг обработчика события нажатия кнопки «sin»(подумайте в какую кнопку добавить новые операции?):**

```
procedure TForm1.Button5Click(Sender:
TObject);
var a, c: real;
begin
```

```

a := StrToFloat(Edit1.Text);
c := sin(a);
  Edit1.Text :=
FloatToStr(c); end;

```

Задание: самостоятельно напишите обработчики событий оставшихся кнопок, используя стандартные функции Lazarus: $\sin(x)$, $\cos(x)$, $\exp(x)$, $\text{abs}(x)$, $\text{sqr}(x)$ (для квадрата числа) и sqrt .

3. Теперь напишем обработчики событий оставшихся двух операций: x^y , $n!$. В качестве примера рассмотрим листинг обработчика события нажатия кнопки

```

«xy»:
procedure TForm1.Button16Click(Sender:
TObject); var a, b, c, i: longint;
begin
a := StrToInt(Edit1.Text); b
:= StrToInt(Edit2.Text); c
:= 1;
      if b = 0 then c:=1
      else for i:=1 to b do
c:=c * a; Edit3.Text
:= IntToStr(c);
end;

```

В данном обработчике событий мы также используем условие, но цель ее использования уже другая – известно, что любое число в степени 0 – это единица, поэтому мы и выполняем проверку на то, какое число введено в качестве степени, и если это 0, то результат в любом случае будет равен единице, в противном случае, т.е. если степень отлична от нуля, мы выполняем процедуру подсчета результата, используя для этого цикл с предусловием.

Тело цикла повторяется b раз, т.е. если нам необходимо посчитать 2 в степени 3 , то мы должны выполнить умножения числа 2 на себя же ровно 3 раза: $2*2*2$. Для выполнения этого действия мы используем следующую команду: $c := c * a$; Причем перед выполнением условия и цикла, мы присвоили переменной c значение единицы, так как в противном случае переменная c была бы равна нулю. Поясним это на нашем примере: 2^3 , т.е. $a = 2$, $b = 3$.

I. Рассмотрим случай, когда мы не сделали бы присвоения $c := 1$:

II. Шаг 1. $c = 0$, $c := c * a \rightarrow c := 0*2 = 0$;

Шаг 2. $c = 0$, $c := c * a \rightarrow c := 0*2 = 0$;

И, наконец, третий шаг – т.к. $b = 3$, следовательно в цикле три итерации(повторения):

Шаг 3. $c = 0$, $c := c * a \rightarrow c := 0*2 = 0$.

Результат $c = 0$, что конечно же является неверным, т.к. 2 в степени 3 равно 8 .

III. Теперь рассмотрим случай, когда мы сделали присвоение $c := 1$:

Шаг 1. $c = 1$, $c := c * a \rightarrow c := 1 * 2 = 2$;

Шаг 2. $c = 2$, $c := c * a \rightarrow c := 2 * 2 = 4$;

И, наконец, третий шаг – т.к. $b = 3$, следовательно в цикле три итерации(повторения):

Шаг 3. $c = 4$, $c := c * a \rightarrow c := 4 * 2 = 8$.

Результат $c = 8$, что является верным результатом.

IV. Теперь рассмотрим случай, когда степень равна 0, т.е. $b = 0$ и мы не используем условие: *if $b = 0$ then $c := 1$* . В этом случае сразу же можно заметить, что цикл *for $i := 1$ to b do $c := c * a$* ; не будет выполнен, так как переменная i начинается со значения 1 и должна увеличиваться, а $b = 0$ меньше 1, т.е. увеличения переменной i не будет.

Задание: самостоятельно напишите обработчик события нажатия кнопки подсчета факториала « $n!$ ». Напомню, что факториалом числа n является значение: $n! = 1 * 2 * 3 * \dots * n$, например $5! = 1 * 2 * 3 * 4 * 5 = 120$, т.е. $5! = 120$. Также напомню, что факториалом числа 0, является 1, т.е. $0! = 1$ (для описания данного случая в обработчике событий Вам потребуется использовать условие наподобие того, которое использовалось в обработчике события нажатия кнопки « xY »).

ПЗ № 23. Задание. Разработать приложение, состоящее из трех модулей:

- 1) главный модуль, считывающий из текстового файла координаты точек на плоскости;
- 2) модуль, содержащий функции расчета расстояния между двумя точками;
- 3) модуль, содержащий функцию, определяющую треугольник с максимальной площадью.

ПЗ № 24. Задание . Описать этапы нисходящего проектирования разработанного приложения.

Задание . Описать этапы восходящего проектирования разработанного приложений.

ПЗ № 25. Задание. Разработать тест - план на основе примера по своему варианту задания

1. Microsoft Excel
2. Acrobat Reader DC
3. Dia
4. Adobe Photoshop CS6
5. 7-Zip File Manager
6. FastStone Image Viewer
7. GIMP

8. Inskape
9. Microsoft Outlook
10. Visio
11. Paint
12. Math Input Panel
13. Microsoft OneNote

Пример Тест-плана

1. ID

Тестирование Блокнота версии 6.1

2. Введение

Этот документ является тест планом по тестированию десктоп приложения Блокнот версии 6.1. Он описывает стратегию и подходы к тестированию продукта. План используется для валидации качества программного обеспечения.

3. Объекты тестирования

Ниже приводится список объектов функционального тестирования:

- работа с файлами,
- печать,
- изменение параметров работы,
- правка,
- форматирование,
- изменение вида,
- вызов справки

4. Что будет тестироваться?

Функции Блокнота, с точки зрения пользователя, что будут тестироваться:

- открытие файла с помощью Блокнота;
- создание файла;
- закрытие приложения;
- печать;
- изменение параметров работы;
- правка;
- форматирование;
- изменение вида;
- вызов справки.

5. Что не будет тестироваться?

Функции Блокнота, с точки зрения пользователя, что не будут тестироваться:

- функции «Печати» - диапазон страниц: выделение, выбор страницы, разобрать по копиям. Причина - во-первых: для тестирования не будет задействован физический принтер; во-вторых: данная функциональность не активна на виртуальном принтере, а так же и для печати в файл.
- функция «Параметры страницы» - способ подачи бумаги. Причина - данная функциональность отсутствует на виртуальном принтере

6. Подход

Вовремя тестирования приложения будет проводится нефункциональное тестирование, а именно:

- тестирование интерфейса
- тестирование удобства использования/ юзабилити

Для функционального тестирования будут использоваться следующие техники тестирования:

- 1) Разбиение на классы эквивалентности (Шрифты)
- 2) Анализ граничных значений (Шрифты)
- 3) Комбинаторное тестирование.

Необходимо написать тест план, с указанием всех ключевых требований, подходов, а так же обязанностей и компетенций соответственно.

Написание тест кейсов в соответствии с распределёнными обязанностями, обязательное их согласование и занесение в тест менеджмент систему.

При создании последнего тест кейса составление матрицы трассируемости требований и просчет покрытия требований тестами.

7. Критерии успешности тестирования

Все тест кейсы с высоким приоритетом закрыты с результатом «пройден/разз».

Тестовое покрытие проверено и является достаточным, где критерий достаточности составляет не менее 99% покрытия требований тестами.

Тест репорт составлен и утвержден тест лидом и заказчиком.

8. Критерии прерывания и продолжения тестирования

Критерием прерывания тестирования является появления и занесения в баг-трекинг систему блокирующих багов.

Критерием продолжения тестирования закрытие блокирующего бага в баг-трекинг системе.

9. Результаты проведения тестирования

Результатом проведения тестирования является получение следующих документов: тест план, тест кейсы, матрица трассируемости требований.

10. Задачи для проведения тестирования

<i>Задача</i>	<i>Расположение</i>
Написание тест плана	Создание тест плана, обязанности
Написание тест кейсов	Объекты тестирования, обязанности
Разработка критериев успешности тестирования	Критерии успешности тестирования
Проведение тестирования и оценка результатов	Подход к тестированию, обязанности
Создание отчетов о результатах тестирования	Результаты проведения тестирования

11. Технические требования

Тестирование приложение будет происходить на следующих операционных системах: Windows XP, Windows 7

12. Обязанности

№ п/п	Роль	Обязанности	Ответственный
1	Лид	Написание тест плана; написания тесткейсов для тестирования следующих функций: открытие, создание, закрытие; осуществление функционального тестирования	Пасечник А.
2	Тестирующий	Написания тест-кейсов для тестирования следующей функции: сохранение; осуществление функционального тестирования	Цимбалюк А.
3	Тестирующий	Написания тест-кейсов для тестирования следующей функции: Параметры страницы:	Бутенко А.
4	Тестирующий	Написания тест-кейсов для тестирования следующей функции: печать; осуществление	Костева В.
5	Тестирующий	Написания тест-кейсов для тестирования следующей функции: Правка; осуществление	Каверин А.
6	Тестирующий	Написания тест-кейсов для тестирования следующей функции: Формат (кроме шрифтов). Вид	Кононский А.
7	Тестирующий	Написания тест-кейсов для тестирования контекстного меню; осуществление функционального	Мирошник А.
8	Тестирующий	Написания тест-кейсов для тестирования горячих клавиш; осуществление функционального	Полищук П.
9	Тестирующий	Написания тест-кейсов для тестирования шрифтов; осуществление функционального	Мирошниченко С.

13. Необходимые компетенции и тренинги

Для выполнения поставленных задач необходимо обладать следующими компетенциями:

- знание и умение использования правил написания тест планов, в том числе основанных на стандарте IEEE-829;
- знание и умение применить техники тест дизайна
- знание различных типов тестирования в том числе функционального и нефункционального, такого как тестирование интерфейса и юзабилити
- умение использование тест менеджмент системы, выбранной для текущего проекта И т.д.

Необходимые тренинги для проведения тестирования проекта:

- тренинг по тестированию шрифтов.=)
- тренинг по использованию специфического программного обеспечения

для более качественного и полного тестирования юзабилити

14. Расписание/срок сдачи

Срок утверждение и внесения всех тест кейсов в тест менеджмент систему - 12/05/2021 23:59:59

Срок составления отчетов 12/05/2021 23:59:59

Срок сдачи проекта - 17/05/2021 19:00:00

15. Риски и их устранение

Возможные риски во время тестирования:

- Недостаточное количество кадровых ресурсов для тестирования приложения в установленные сроки
- Отсутствие необходимого оборудования, программного обеспечения, данных или инструментов.
- Изменения в оригинальных требований или инструкций.
- Количество допустимых дефектов будет увеличено.
- Тест команда будет работать сверхурочно. Это негативно может повлиять на боевой дух команды.
- Объемы плана могут быть изменены.
- тестирование приложения может быть просто остановлено (крайний случай)

16. Утверждение

Утверждение тест кейсов - Ответственный тест лид – Пасечник

Прием готового проекта - Ответственный – Петрова Карина, 652 группа

ПЗ № 26. Задание. Разработать отчет о дефектах на основе примера по своему варианту задания (выбираете любую часть и тестируете ее, оформляя отчет о дефектах)

1. Microsoft Excel
2. Acrobat Reader DC
3. Dia
4. Adobe Photoshop CS6
5. 7-Zip File Manager
6. FastStone Image Viewer
7. GIMP
8. Inksape
9. Microsoft Outlook
10. Visio
11. Paint
12. Math Input Panel
13. Microsoft OneNote

Пример отчета о дефектах

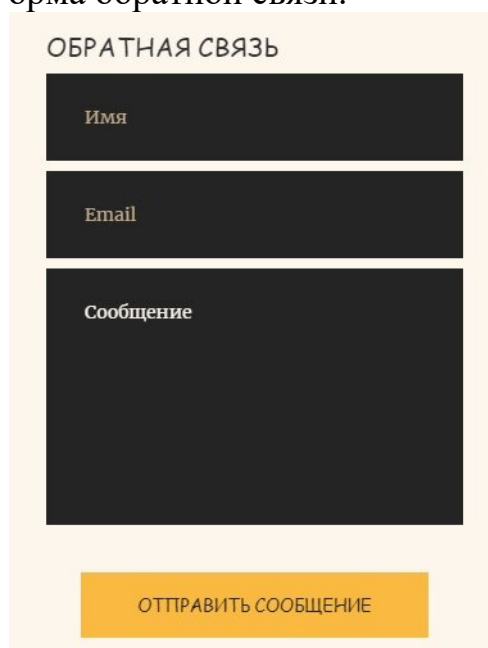
Отчеты о дефектах (баг репорт). Шаблон отчета об ошибке

Шаблон отчета о дефекте, который отвечает запросам тестировщика и программиста, выглядит следующим образом:

1. Заголовок ошибки
2. Описание ошибки
3. Начальные условия
4. Шаги воспроизведения
5. Ожидаемый результат
6. Фактический результат
7. Вложения

В зависимости от ситуации или компании, в которой вы работаете, шаблон может изменяться и отклоняться в разные стороны. Например, в некоторых случаях «Начальные условия» не пишут. Если дефект связан с графикой, то рекомендуется добавить скриншот. Также вводятся дополнительные атрибуты для указания Платформы или Браузера и т.д.

Теперь рассмотрим структуру шаблона подробно на конкретном примере. Допустим, мы тестируем сайт. На сайт есть раздел Контакты. В этом разделе находится Форма обратной связи.



The image shows a contact form with the following structure:

- Title: ОБРАТНАЯ СВЯЗЬ
- Field 1: Имя (Name)
- Field 2: Email
- Field 3: Сообщение (Message)
- Button: ОТПРАВИТЬ СООБЩЕНИЕ (SEND MESSAGE)

Баг на сайте

После заполнения всех полей мы нажимаем на кнопку «Отправить сообщение» и ничего не происходит.

Данный баг мы и будем описывать по шаблону.

Заголовок ошибки

По сути это краткое описание найденной ошибки. Его задача — в понятной и простой форме передать смысл найденной ошибки.

Наиболее эффективным описанием считается описание, которое отвечает на три вопроса:

- *Что* произошло?
- *Где* появилась ошибка?
- *Когда или при каких условиях* найден дефект?

Также важно, чтобы заголовок был именно **кратким**, т.е. он должен содержать максимально полную и, в то же время, краткую информацию об ошибке.

Заголовок ошибки — это первое, что видит разработчик, получая отчет. В некоторых случаях, при правильном оформлении, этого бывает достаточно, чтобы понять в чем заключается дефект и как его исправить.

Пример плохого заголовка: Ошибка, когда нажимаю «Отправить сообщение».

Пример хорошего заголовка: При нажатии на кнопку «Отправить сообщение» в форме обратной связи сообщение не отправляется.

Описание ошибки

Попробуйте в паре предложений описать суть дефекта, а также когда он появляется и в чем выражен.

Правильное и качественное описание также позволяет сразу понять проблему и приступить к ее исправлению.

Пример плохого описания: Жму «Отправить сообщение», а в ответ тишина.

Пример хорошего описания: При нажатии на кнопку «Отправить сообщение» в заполненной форме обратной связи ничего не происходит. Аналогичное поведение, если форма не заполнена.

Начальные условия

В случае, если есть специфичные действия или шаги воспроизведения достаточно объемные, то указываются начальные условия. Например:

1. Быть авторизованным в системе.
2. Находиться на главной странице.

Пример плохих начальных условий: Находиться на сайте.

Пример хороших начальных условий:

1. Страница «Контакты»,
2. Платформа и устройство не имеют значения.

Шаги воспроизведения

Шаги, при которых повторяется найденная ошибка. Например:

1. Нажать на кнопку «Войти»
2. Ввести «Имя пользователя» и «Пароль»
3. Нажать на кнопку «Ок»

Убедитесь, что у вас нет лишних или ненужных шагов воспроизведения, которые будут отвлекать и тратить время команды.

Пример плохих шагов: 1. Зайти на сайт, 2. Зайти на страницу обратной связи, 3. Поставить курсор в поле Имя, 4. Ввести имя, 5. Поставить курсор в поле e-mail, 6. Ввести действующий e-mail, 7. Навести курсор на Отправить сообщение, 8. Щелкнуть по Отправить сообщение.

Пример хороших шагов:

1. Заполнить поля формы обратной связи,
2. Нажать на кнопку «Отправить сообщение»

Ожидаемый результат

Результат, который должен быть при выполнении шагов. В идеале, его можно найти в ТЗ (техническом задании). На практике, ТЗ бывает не всегда и ожидаемый результат определяется либо здравым смыслом, либо по аналогии.

Пример плохого ожидаемого результата: Что-то должно произойти.

Пример хорошего ожидаемого результата: Сообщение отправляется либо система сообщает о невозможности его отправки.

Фактический результат

Указывается результат, который получил тестировщик при выполнении описанных шагов. Также может отвечать на три вопроса “Что? Где? Когда?”.

Пример плохого фактического результата: Ничего нет.

Пример хорошего фактического результата: Сообщение не отправляется, не появляется ошибка об отправке. После нажатия на кнопку ничего не происходит.

Вложения

При необходимости тестировщики прикладывают скриншоты или видео воспроизведения ошибки. Также могут прикладывать логи выполнения программы. При соблюдении правил оформления баг-репортов, ваша работа станет более эффективной.

3. Комплект оценочных средств для промежуточной аттестации

3.1. Контрольные вопросы (КВ)

КВ №1. Дайте определение понятия проект. Охарактеризуйте состав и структуру коллектива разработчиков, их функции.

КВ №2. Охарактеризуйте структурный подход к проектированию ИС. CASE - средства разработки ПО.

КВ №3. Опишите как осуществляется моделирование потоков данных (процессов). Внешние сущности. Системы и подсистемы. Процессы. Накопители данных. Потоки данных. Построение иерархии диаграмм потоков данных.

КВ №4. Охарактеризуйте метод моделирования IDEF3.

КВ №5. Охарактеризуйте, что представляет собой методология DFD как инструмент моделирования потоков данных.

КВ №6. Опишите инструменты функционального моделирования бизнес-процессов и использованием стандарта IDEF0.

КВ №7. Сформулируйте понятие и принципы работы с инструментальными средствами разработки ПО

- KB №8. Опишите методы организации коллективной разработки ПО
- KB №9. Охарактеризуйте процесс разработки сетевой модели
- KB №10. Опишите элементы Microsoft Office Project 2007
- KB №11. Опишите элементы графической нотации DFD
- KB №12. Опишите элементы методологии IDEF0
- KB №13. Охарактеризуйте процесс имитационного моделирования
- KB №14. Опишите Case-метод Баркера
- KB №15. Объясните как осуществляется генерация кода клиентской части с помощью ERwin
- KB №16. Опишите нотацию ARIS eEPC
- KB №17. Охарактеризуйте модель AS-IS
- KB №18. Охарактеризуйте модель TO-BE
- KB №19. Дайте определение понятию отладки программного средства
- KB №20. Дайте определение понятию программного модуля.
- KB №21. Опишите методические аспекты проектирования ПО. Общие принципы проектирования систем.
- KB №22. Расскажите про основы объектно-ориентированного подхода к анализу и проектированию ПО. Унифицированный язык моделирования UML.
- KB №23. Объясните функциональное проектирование ИСО, IDEF0, синтаксис, особенности проектирования.
- KB №24. Объясните функциональное проектирование ИСО, IDEF3, синтаксис, особенности проектирования.
- KB №25. Опишите методологию DFD для проектирования ИСО.

Критерии оценивания

«5» «отлично» или «зачтено» – студент показывает глубокое и полное овладение содержанием программного материала по МДК, в совершенстве владеет понятийным аппаратом и демонстрирует умение применять теорию на практике, решать различные практические и профессиональные задачи, высказывать и обосновывать свои суждения в форме грамотного, логического ответа (устного или письменного), а также высокий уровень овладения общими и профессиональными компетенциями и демонстрирует готовность к профессиональной деятельности;

«4» «хорошо» или «зачтено» – студент в полном объеме освоил программный материал по МДК, владеет понятийным аппаратом, хорошо ориентируется в изучаемом материале, осознанно применяет знания для решения практических и профессиональных задач, грамотно излагает ответ, но содержание, форма ответа (устного или письменного) имеют отдельные неточности, демонстрирует средний уровень овладения общими и профессиональными компетенциями и готовность к профессиональной

деятельности;

«3» «удовлетворительно» или «зачтено» – студент обнаруживает знание и понимание основных положений программного материала по МДК но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения практических и профессиональных задач, не умеет доказательно обосновать свои суждения, но при этом демонстрирует низкий уровень овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности;

«2» «неудовлетворительно» или «не зачтено» – студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, беспорядочно и неуверенно излагает программный материал по МДК, не умеет применять знания для решения практических и профессиональных задач, не демонстрирует овладение общими и профессиональными компетенциями и готовность к профессиональной деятельности.

3. Информационное обеспечение

перечень учебных изданий, электронных изданий, электронных и Интернет-ресурсов, образовательных платформ, электронно-библиотечных систем, веб-систем для организации дистанционного обучения и управления им, используемые в образовательном процессе как основные и дополнительные источники.

Основные источники:

1. Гагарина, Л.Г. Технология разработки программного обеспечения: учебное пособие/ -М.: Форум,2020 - 400 с.
2. Федорова, Г.Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие/ - М.:Курс:ИНФРА-М,2019-336 с.
3. Федорова,Г.Н. Осуществление интеграции программных модулей: учебник/ 5-е изд., стер.- ОИЦ Академия, 2023 - 272 с.

Дополнительные источники:

1. Богомазова Г.Н. Модернизация программного обеспечения персональных компьютеров, серверов, периферийных устройств и оборудования: учебник.- М.: ИЦ Академия, 2016.-192 с.
2. Гвоздева В. А. Информатика, автоматизированные информационные технологии и системы. Учебник.- М.: ИД ФОРУМ, 2017.- 544 с.
3. Федорова Г.Н. Разработка программных модулей программного

обеспечения для компьютерных систем. Федорова Г.Н. – М.: Академия, 2017. – 336 с.

4. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; Под ред. Л. Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2017.-400 с.

5. Калайда В.Т., Романенко В.В. Технология разработки программного обеспечения: учебное пособие.-Томск: Томский межвузовский центр дистанционного образования, 2007.-257 с.

6. Рудаков А.В. Технология разработки программных продуктов (11-е изд., стер.) учебник / Рудаков А.В. - М : ИЦ Академия, 2017-208 с.

7. Черпаков И.В. Основы программирования: Учебник и практикум для СПО.- М.: Юрайт,2017.-219 с

Электронные издания (электронные ресурсы)

1. НОУ ИНТУИТ: <http://www.intuit.ru/department/se/devis/>

2. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. http://real.tepkom.ru/Real_OM-СМ_A.asp

Цифровая образовательная среда СПО PROФобразование:

- Сеницын, С. В. Основы разработки программного обеспечения на примере языка С : учебное пособие для СПО / С. В. Сеницын, О. И. Хлытчиев. — Саратов : Профобразование, 2019. — 212 с. — ISBN 978-5-4488-0362-8. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/86201> (дата обращения: 04.09.2020). — Режим доступа: для авторизир. пользователей

- Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие для СПО / Т. М. Зубкова. — Саратов : Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/86208> (дата обращения: 05.09.2020). — Режим доступа: для авторизир. Пользователей

- Вичугова, А. А. Инструментальные средства разработки компьютерных систем и комплексов : учебное пособие для СПО / А. А. Вичугова. — Саратов : Профобразование, 2017. — 135 с. — ISBN 978-5-4488-0015-3. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/66387> (дата обращения: 04.09.2020). — Режим доступа: для авторизир. пользователей

Электронно-библиотечная система:

IPR BOOKS - <http://www.iprbookshop.ru/86201.html>

<http://www.iprbookshop.ru/86208.html>

<http://www.iprbookshop.ru/66387.html>

Веб-система для организации дистанционного обучения и управления им:

Система дистанционного обучения ОГАПОУ «Алексеевский колледж»
<http://moodle.alcollege.ru/>