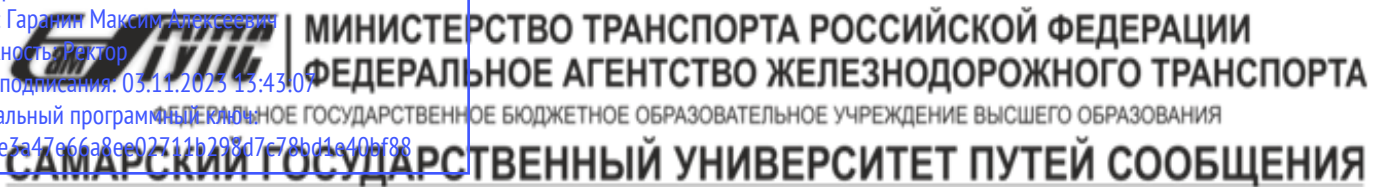


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Гарант Максим Алексеевич  
Должность: Ректор  
Дата подписания: 03.11.2023 13:43:07  
Уникальный программный ключ:  
7708e7a47e66a8ee02711b298d7e78bd1e40bf88



Приложение  
к рабочей программе дисциплины

## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

### Визуальное программирование

---

*(наименование дисциплины(модуля))*

Направление подготовки / специальность

**09.03.02 Информационные системы и технологии**

---

*(код и наименование)*

Направленность (профиль)/специализация

**Информационные системы и технологии на транспорте**

---

*(наименование)*

**Форма обучения**

Очная

Семестр 5 (зачет)

## Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

## 1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации: зачет в 5 семестре.

### Перечень компетенций, формируемых в процессе освоения дисциплины

| Код и наименование компетенции                                     | Код индикатора достижения компетенции   |
|--|---|
| ПК-1 Способен создавать инструментальные средства программирования | ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня |
|  | ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня       |

### Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

| Код и наименование индикатора достижения компетенции                            | Результаты обучения по дисциплине  | Оценочные материалы |
|---|--|---------------------|
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня | Обучающийся знает: Методы и средства интеграции модулей и компонент программного обеспечения; интерфейсы взаимодействия модулей системы между собой и с внешней средой; методы и средства разработки процедур развертывания программного обеспечения.  | Вопросы (1-8)       |
|   | Обучающийся умеет: Выполнять процедуры сборки программных модулей и компонент в программный продукт; проводить оценку работоспособности программного продукта; документировать произведенные действия, выявленные проблемы и способы их устранения.  | Вопросы (1-5)       |
|   | Обучающийся владеет: Навыками интеграции программных компонент собственной разработки и приобретенных.   | Вопросы (9-18)      |
| ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня       | Обучающийся знает: методы и средства верификации работоспособности выпусков программной продукции; языки, утилиты и среды программирования, средства пакетного выполнения процедур.  | Вопросы (15-28)     |
|   | Обучающийся умеет: производить настройки параметров программного продукта и осуществлять запуск процедур сборки; проводить оценку работоспособности программного продукта; создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных. | Вопросы (22-27)     |
|   | Обучающийся владеет: разработки и осуществления процедур верификации выпусков (сборок) программной продукции.  | Вопросы (23-28)     |

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий;
- 2) выполнение заданий в ЭИОС СамГУПС.

## 2. Типовые<sup>1</sup> контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

### 2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

| Код и наименование индикатора достижения компетенции   | Образовательный результат   |
|--|---|
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня  | Обучающийся знает: Методы и средства интеграции модулей и компонент программного обеспечения; интерфейсы взаимодействия модулей системы между собой и с внешней средой; методы и средства разработки процедур развертывания программного обеспечения. |
| <i>Примеры вопросов/заданий</i><br>1. В чем принципиальное отличие визуального программирования от любого другого программирования?<br>2. Возможно, ли средствами визуального программирования создать законченное программное приложение?<br>3. Какие достоинства и недостатки присущи визуальному программированию?<br>4. Какие языки программирования поддерживают визуальный стиль программирования?<br>5. В каких средах программирования реализован визуальный стиль программирования?<br>6. Какие фирмы-разработчики пакетов программирования имеют наиболее удачные решения для визуального программирования?<br>7. Можно ли сказать, что визуальное программирование - это построение графического интерфейса пользователя?<br>8. Какие элементы процесса визуализации вы можете перечислить?<br>9. Что такое "визуализируемая модель"?<br>10. Приведите примеры визуального программирования в известных вам программных средах.   |   |
| ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня  | Обучающийся знает: методы и средства верификации работоспособности выпусков программной продукции; языки, утилиты и среды программирования, средства пакетного выполнения процедур.   |
| <i>Примеры вопросов/заданий</i><br>1. Базовые понятия. Программирование. Язык. ЭВМ (компьютер). ЦВМ и АВМ.<br>2. Разновидности языков. Синтаксис, семантика и прагматика языка. Примеры.<br>3. История развития языков программирования. Первое, второе и третье поколения языков программирования.<br>4. Направления исследований в области языков программирования четвертого поколения.<br>5. Классификация средств визуального программирования по основному описываемому аспекту программы.<br>6. Классификация средств визуального программирования по области применения и этапам жизненного цикла программных средств. Понятие CASE.<br>7. Системы визуального конструирования интерфейса пользователя. Основные возможности, ограничения. Концепции WIMP и WYSIWYG. Примеры.<br>8. Визуальный язык VUFC. Возможности, область применения.<br>9. Язык Scratch. Область применения, основные возможности.<br>10. Графический интерпретатор блок-схем. Области применения, возможности.<br>11. Языки Sivil и VIPR. Области применения, возможности.<br>12. Графические языки, ориентированные на состояния. Язык SDL. Графический алфавит, область применения. Инструментальные средства программирования. |   |

<sup>1</sup> Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и социальной сферы. Ответственность за нераспространение содержания оценочных средств среди обучающихся университета несут заведующий кафедрой и преподаватель – разработчик оценочных средств.

13. Графические языки, ориентированные на состояния. Диаграмма конечного автомата языка UML. Основные графические примитивы. Инструментальные средства.

## 2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

| Код и наименование индикатора достижения компетенции   | Образовательный результат   |
|--|---|
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня  | Обучающийся умеет: Выполнять процедуры сборки программных модулей и компонент в программный продукт; проводить оценку работоспособности программного продукта; документировать произведенные действия, выявленные проблемы и способы их устранения. |
| <p><i>Примеры заданий</i></p> <p>46. Какой агрегат называется правильно построенным?</p> <p>47. В чем суть проблемы приведения графа управления модели параллельного алгоритма к стандартному виду.</p> <p>48. Что такое десуперпозиция Р-графа?</p> <p>49. Опишите алгоритм F-нумерации Р-графа.</p> <p>50. Какие данные называются критическими?</p> <p>51. Дайте определение понятию «Состояние модели параллельных вычислений».</p> <p>52. Опишите простейший алгоритм поиска критических данных.</p> <p>53. Какие операции используются в алгебре способов использования данных.</p> <p>54. Чем отличаются друг от друга операции следования и ветвления?</p> <p>55. Опишите правило построения формул над способами использования данных.</p> <p>56. Свойства операций следования, ветвления и параллельного исполнения.</p> <p>57. Применение формул над способами использования данных.</p> <p>58. Что такое семафорный предикат?</p> <p>59. Дайте определение обобщенного семафорного предиката.</p> <p>60. Когда критическое данное не приводит к конфликту?</p> <p>61. В каких случаях в параллельных вычислениях возникают тупиковые ситуации?</p> <p>62. Что такое покрывающее дерево?</p>  |   |
| ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня  | Обучающийся владеет: Навыками интеграции программных компонент собственной разработки и приобретенных.  |
| <p><i>Примеры заданий</i></p> <p>1. На рисунке представлена авторская граф-модель параллельного алгоритма. Используя алгоритм F-нумерации, произведите разметку вершин графа. Постройте модель В.</p> <p>2. На основе построенной В-модели графа произведите его десуперпозицию на совокупность вложенных правильно построенных агрегатов (постройте С-модель графа).</p> <p>3. Произведите преобразование графа управления модели параллельного алгоритма (С-модели) к стандартному виду D-модели.</p> <p>4. Используя формулу вычисления способа использования данных, построить один из вариантов графа вычислительного процесса. <math>H(G,d)=A_0 \square (A_1 \# A_2 \# (A_3 \square (A_4 \# A_5) \square A_6)) \square A_7</math>.</p> <p>5. Используя формулу вычисления способа использования данных, построить один из вариантов графа вычислительного процесса. <math>H(G,d)=A_0 \square (A_1 \# A_2 \# (A_3 \square (A_4 \square A_5) \square A_6)) \square A_7</math>.</p> <p>6. Используя формулу вычисления способа использования данных, построить один из вариантов графа вычислительного процесса. <math>H(G,d)=A_0 \square (A_1 \square A_2 \square (A_3 \square (A_4 \# A_5) \square A_6)) \square A_7</math></p> <p style="text-align: center;"><b>Задание</b></p> <p>Используя определение операций следования, ветвления и параллельного исполнения, а также признак использования параметра d, заданного в таблице 1 для каждого модуля, вычислить значение</p> |   |

|  |   |
|--|---|
| <p>формулы использования данных некоторого графа агрегата G: <math>H(G,d)=A0 \square(A1 \# A2 \# (A3 \square(A4 \# A5) \square A6)) \square A7</math>. Является ли данное d критическим?</p>   |   |
| <p>ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня</p>   | <p>Обучающийся умеет: производить настройки параметров программного продукта и осуществлять запуск процедур сборки; проводить оценку работоспособности программного продукта; создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных.</p> |
| <p><i>Примеры заданий</i></p> <ol style="list-style-type: none"> <li>14. Графические языки, ориентированные на состояния. Языки Graphcet и SFC. Основные графические примитивы. Инструментальные средства программирования.</li> <li>15. Графический язык Argos. Области применения, особенности.</li> <li>16. Языки описания потока управления (control flow). Стандартный язык блок-схем алгоритмов и программ. Возможности описания, инструментальные средства.</li> <li>17. Языки описания потока управления (control flow). Технология ГРАФКОНТ/ГЕОЗ.</li> <li>18. Языки описания потока управления (control flow). Технология графосимволического программирования ГРАФ.</li> <li>19. Языки описания потока управления (control flow). Диаграмма деятельности (активностей) языка UML – возможности описания, основные графические примитивы.</li> <li>20. Языки описания потока управления (control flow). Технология ГРАФИТ/ФЛОКС.</li> <li>21. Языки описания потока управления (control flow). Язык P-схем. Особенности, пример диаграммы.</li> <li>22. Языки графического описания структур данных. Возможности, инструментальные средства.</li> <li>23. Диаграмма классов языка UML. Возможности, инструментальные средства генерации программы.</li> <li>24. Языки графического описания потоков данных. Пример DFD-диаграммы. Языки LD и FBD стандарта МЭК 61131</li> <li>25. Языки графического описания потоков данных. LabVIEW и Simulink. Проект «Птолемей».</li> <li>26. Событийный подход в графическом программировании. Описание возможностей и принципов функционирования среды Visual Age.</li> <li>27. Передача методов и событий в среде графического программирования HiAsm.</li> </ol> |   |
| <p>ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня</p>   | <p>Обучающийся владеет: разработки и осуществления процедур верификации выпусков (сборок) программной продукции.</p>  |
| <p><i>Примеры заданий</i></p> <ol style="list-style-type: none"> <li>1. Какие средства автоматизации проектирования программного обеспечения Вы знаете?</li> <li>2. Перечислите основные парадигмы программирования, способствующие развитию методов автоматизации разработки программного обеспечения.</li> <li>3. Что такое визуальное программирование?</li> <li>4. Классификация средств визуального программирования.</li> <li>5. Какие визуальные средства разработки программного обеспечения Вы знаете?</li> <li>6. Перечислите графические языки визуального программирования.</li> <li>7. Какие основные направления разработки программ, описывающих параллельные вычислительные процессы, Вы знаете?</li> <li>8. Опишите основные классы графических моделей параллельных вычислительных процессов.</li> <li>9. Какие графические нотации используются в средстве визуального описания моделей параллельных процессов в сетях Петри?</li> <li>10. Содержательное наполнение вершин и дуг в диаграммах описания потоков данных.</li> <li>11. Содержательное наполнение вершин и дуг в диаграммах описания потоков управления.</li> <li>12. В чем состоят основные положения технологии графосимволического программирования?</li> <li>13. Какую алгоритмическую модель использует технология ГСП?</li> <li>14. Какие объекты используются в технологии ГСП?</li> <li>15. Опишите содержательно акторы, предикаты, агрегаты.</li> <li>16. Что такое тип данного в ГСП?</li> </ol>  |   |

17. Дайте определение типа базового модуля.
18. Какую роль в ГСП играют базовые модули?
19. Чем отличаются агрегаты от inline модулей?
20. Каким образом в технологии ГСП организован межмодульный информационный интерфейс?
21. Как в технологии ГСП происходит управление вычислительным процессом? Для агрегатов? Для акторов?
22. Роль граф-машины в управлении агрегатов технологии ГСП?
23. Опишите способ представления графа управления, принятый при компиляции агрегата.
24. 24. Перечислите способы синтеза объектов технологии ГСП.
25. Как порождаются акторы и предикаты в технологии ГСП? Что такое «паспортизация» модулей?
26. Как строятся агрегаты?
27. Объясните алгоритм работы граф-машины, роль граф-машины в технологии ГСП.
28. Основные компоненты автоматически порождаемой граф-программы.
29. В чем состоит проблема классификации данных агрегатов?

### **2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации**

1. Базовые понятия. Программирование. Язык. ЭВМ (компьютер). ЦВМ и АВМ.
2. Разновидности языков. Синтаксис, семантика и прагматика языка. Примеры.
3. История развития языков программирования. Первое, второе и третье поколения языков программирования.
4. Направления исследований в области языков программирования четвертого поколения.
5. Классификация средств визуального программирования по основному описываемому аспекту программы.
6. Классификация средств визуального программирования по области применения и этапам жизненного цикла программных средств. Понятие CASE.
7. Системы визуального конструирования интерфейса пользователя. Основные возможности, ограничения. Концепции WIMP и WYSIWYG. Примеры.
8. Визуальный язык VUFC. Возможности, область применения.
9. Язык Scratch. Область применения, основные возможности.
10. Графический интерпретатор блок-схем. Области применения, возможности.
11. Языки Sivil и VIPR. Области применения, возможности.
12. Графические языки, ориентированные на состояния. Язык SDL. Графический алфавит, область применения. Инструментальные средства программирования.
14. Графические языки, ориентированные на состояния. Диаграмма конечного автомата языка UML. Основные графические примитивы. Инструментальные средства.
15. Графические языки, ориентированные на состояния. Языки Graphcet и SFC. Основные графические примитивы. Инструментальные средства программирования.
16. Графический язык Argos. Области применения, особенности.
17. Языки описания потока управления (control flow). Стандартный язык блок-схем алгоритмов и программ. Возможности описания, инструментальные средства.
18. Языки описания потока управления (control flow). Технология ГРАФКОНТ/ГЕОЗ.
19. Языки описания потока управления (control flow). Технология графосимволического программирования ГРАФ.
20. Языки описания потока управления (control flow). Диаграмма деятельности (активностей) языка UML – возможности описания, основные графические примитивы.
21. Языки описания потока управления (control flow). Технология ГРАФИТ/ФЛОКС.
22. Языки описания потока управления (control flow). Язык P-схем. Особенности, пример диаграммы.
23. Языки графического описания структур данных. Возможности, инструментальные средства.

24. Диаграмма классов языка UML. Возможности, инструментальные средства генерации программы.
25. Языки графического описания потоков данных. Пример DFD-диаграммы. Языки LD и FBD стандарта МЭК 61131
26. Языки графического описания потоков данных. LabVIEW и Simulink. Проект «Птолемей».
27. Событийный подход в графическом программировании. Описание возможностей и принципов функционирования среды Visual Age.
28. Передача методов и событий в среде графического программирования HiAsm.

### **3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации**

#### **Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий**

- оценка **«отлично»** выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90 % от общего объёма заданных вопросов;
- оценка **«хорошо»** выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76 % от общего объёма заданных вопросов;
- оценка **«удовлетворительно»** выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка **«неудовлетворительно»** выставляется обучающемуся, если количество правильных ответов – менее 60 % от общего объёма заданных вопросов.

#### **Критерии формирования оценок по результатам выполнения заданий**

**«Отлично/зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов.

**«Хорошо/зачтено»** – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

**«Удовлетворительно/зачтено»** – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

**«Неудовлетворительно/не зачтено»** – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

*Виды ошибок:*

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

#### **Критерии формирования оценок по результатам выполнения практических работ**

**«Зачтено»** – ставится за работу, выполненную полностью без ошибок и недочетов в соответствии с заданием. Обучающийся полностью владеет информацией по теме работы, решил все поставленные в задании задачи.

**«Не зачтено»** - ставится за работу, если обучающийся правильно выполнил менее 2/3 всей работы, использовал при выполнении работы неправильные алгоритмы, допустил грубые ошибки при расчетах, сформулировал неверные выводы по результатам работы.



### **Критерии формирования оценок по зачету**

**«Зачтено»** - обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности.

**«Не зачтено»** - выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.