Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт экзамена

по дисциплине «Инфраструктура разработки программного обеспечения для мобильных роботов», 2 семестр

1. Методика оценки

Экзамен проводится в форме письменного тестирования с использованием электронной информационно-образовательной среда НГТУ (http:// www.nstu.ru/sveden/eos).

Тестовые задания охватывают все содержание «Инфраструктура разработки программного обеспечения для мобильных роботов».

Тест состоит из 30 вопросов различного вида и позволяет проверить результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций.

2. Критерии оценки результатов обучения, соотнесенных с уровнями освоения индикаторов достижения компетенций

Выполнение теста засчитывается на **продвинутом** уровне, если студент правильно ответил на все вопросы теста, знает определения всех понятий, продемонстрировал способность безошибочно устанавливать последовательность и соответствие между процессами и явлениями, выявлять проблемы, предлагать механизмы их решения, представляет количественные и качественные характеристики определенных процессов и не допускает ошибок при решении задач. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций не содержит пробелов. Установленные в программе компетенции сформированы в полном объеме. Оценка составляет *от 35 до 40 баллов.*

Выполнение теста засчитывается на **базовом** уровне, если студент правильно ответил на ⅔ вопросов теста, знает определения основных понятий, продемонстрировал способность устанавливать последовательность и соответствие между процессами и явлениями, правильно характеризует процессы, явления, не допускает существенных ошибок при решении задач. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит несущественные пробелы. Установленные в программе компетенции сформированы на базовом уровне. Оценка составляет *от 34 до 27 баллов.*

Выполнение теста засчитывается на **пороговом** уровне, если студент правильно ответил от ½ до ⅔ вопросов теста, знает определения основных понятий, продемонстрировал способность устанавливать последовательность и соответствие между процессами и явлениями, при решении задач допускает непринципиальные ошибки, например, вычислительные. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит пробелы. Установленные в программе компетенции сформированы на пороговом уровне. Оценка составляет *от 26 до 20 баллов.*

Выполнение теста считается **неудовлетворительным**, если студент правильно

ответил менее чем на половину вопросов теста, не знает определений понятий, не продемонстрировал способность устанавливать последовательность и соответствие между процессами и явлениями, при решении задач допускает принципиальные ошибки. Совокупность результатов обучения по дисциплине и соотнесенных с ними индикаторов достижения компетенций содержит существенные пробелы. Установленные в программе компетенции не сформированы. Оценка составляет *менее 20 баллов.*

3. Шкала оценки

Экзамен считается сданным, если сумма баллов по всем заданиям билета составляет от 20 до 40 баллов включительно. Сумма менее 20 баллов признается неудовлетворительным результатом промежуточной аттестации по дисциплине.

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, установленными в НГТУ.

4. Примерный тест для экзамена

Утверждаю:
Зав. кафедрой ____ _____ ФИО
«____» _____ 20_ г.

Тест

по дисциплине «Инфраструктура разработки программного обеспечения для мобильных роботов»[1]

1. Какая основная задача операционной системы робота (*ROS*)? а) Управление электропитанием робота б) Прямое управление пинами микроконтроллера в) **Обеспечение взаимодействия между программными компонентами (модульность и коммуникация)** г) Создание 3D-моделей для симуляции

2. Что такое "Нода" (*Node*) в контексте *ROS*? а) Физический вычислительный узел в кластере б) **Отдельный исполняемый файл, который выполняет вычисления** в) Точка в графе, визуализируемом в *rviz* г) Сетевой адрес компьютера

3. Какой процесс в *ROS 1* является центральным и представляет собой единую точку отказа? а) *roscore* б) *rviz* в) *gazebo* г) **Мастер-узел (*Master Node*)**

4. Как называется основной механизм асинхронной передачи данных в *ROS*? а) Сервис (*Service*) б) Действие (*Action*) в) **Топик (*Topic*)** г) Параметр (*Parameter*)

5. Какой тип взаимодействия предполагает немедленный ответ на запрос (запрос-ответ)? а) Топик (*Topic*) б) **Сервис (*Service*)** в) Трансформация (*tf*) г) Действие (*Action*)

6. Для каких задач лучше всего подходит механизм "Действия" (*Actions*)? а) Для передачи текущего значения одометрии б) Для запроса имени ноды в) **Для длительных**

[1] Правильные ответы выделены жирным шрифтом

задач с возможностью отмены и получения обратной связи (например, навигация) г) Для передачи одиночного сообщения о столкновении

7. Что такое "tf" в *ROS*? а) Тип файловой системы для хранения карт б) Библиотека для машинного обучения в) **Система для отслеживания взаимного расположения координатных фреймов робота** г) Протокол передачи данных

8. Какой инструмент используется для визуализации данных с сенсоров и состояния робота? а) *catkin_make* б) *rosbag* в) **rviz** г) Gazebo

9. Какая утилита позволяет записывать и воспроизводить данные, проходящие по топикам? а) *rviz* б) **rosbag** в) *rqt_graph* г) Gazebo

10. Что является "middleware" (промежуточным ПО) в архитектуре *ROS 2*? а) Мастер-узел б) Каталог сообщений в) **Протокол DDS (Data Distribution Service)** г) Система сборки *catkin*

11. Какая ключевая проблема архитектуры *ROS 1* решается в *ROS 2* с помощью DDS? а) Сложность установки б) **Отсутствие единой точки отказа и поддержка систем реального времени** в) Отсутствие поддержки Python г) Невозможность использования в симуляции

12. Что из перечисленного является реализацией стандарта DDS? а) Docker б) Git в) **OpenSplice** г) CMake

13. В чем заключается главное отличие ноды-писателя (*Publisher*) от сервиса-сервера (*Service Server*)? а) Писатель работает быстрее б) Сервис-сервер хранит данные в базе в) Писатель знает о своих читателях г) **Писатель отправляет данные всем подписчикам топика, не зная их, а сервис-сервер отвечает на запрос конкретного клиента**

14. Какой файл описывает структуру пользовательского сообщения в *ROS*? а) `CMakeLists.txt` б) `package.xml` в) `launch.launch` г) **`.msg`**

15. Для чего используется файл `package.xml` в пакете *ROS*? а) Для описания зависимостей и метаданных пакета б) Для компиляции исходного кода в) Для запуска нод г) Для определения топиков

16. Какая команда позволяет увидеть список всех активных топиков в системе? а) `rosnode list` б) `rospack list` в) `**rostopic list**` г) `rosparam list`

17. Что делает команда `roslaunch`? а) Запускает одну конкретную ноду б) Компилирует пакет в) **Запускает несколько нод и других процессов, описанных в файле .launch, одной командой** г) Записывает данные в bag-файл

18. В чем основная функция симулятора Gazebo? а) Визуализация данных с сенсоров б) Управление версиями кода в) **Создание физически реалистичного виртуального мира и симуляция поведения робота в нем** г) Мониторинг сети

19. Какой инструмент визуализирует граф связей между запущенными нодами? а) rviz б) rosbag в) **rqt_graph** г) Gazebo

20. Что такое "QoS" (Quality of Service) в контексте DDS/ROS 2? а) Качество написанного кода б) Скорость сборки пакета в) Версия операционной системы г) **Набор политик, определяющих надежность, долговечность и другие характеристики передачи данных (например, надежность или история)**

21. Какой профиль QoS гарантирует, что новый подписчик получит все пропущенные сообщения? а) BEST_EFFORT_RELIABILITY_QOS_POLICY б) VOLATILE_DURABILITY_QOS_POLICY в) **TRANSIENT_LOCAL_DURABILITY_QOS_POLICY** г) KEEP_LAST_HISTORY_QOS_POLICY

22. Какая система сборки используется по умолчанию в *ROS 1*? а) colcon б) make в) **catkin** г) CMake

23. Какая система сборки используется по умолчанию в *ROS 2*? а) catkin б) make в) CMake г) **colcon**

24. Что такое "Пространство фреймов" (Frame Tree) в tf? а) Список всех роботов на карте б) Иерархическая структура координатных систем робота (например, `map -> odom -> base_link`) в) Дерево файлов проекта г) Структура топиков

25. Какую задачу решает алгоритм SLAM? а) Локализация робота на готовой карте б) Построение кратчайшего пути к цели в) **Одновременная навигация и построение карты неизвестной среды** г) Управление манипулятором

26. Какую задачу решает алгоритм AMCL? а) Построение карты с нуля б) Управление скоростью моторов в) **Локализация робота на уже известной карте с помощью данных с лидара и одометрии** г) Распознавание объектов

27. В чем основное преимущество использования контейнеров (например, Docker) при разработке ПО для роботов? а) Ускоряет выполнение кода на CPU б) Упрощает написание алгоритмов на C++ в) **Изолирует окружение и зависимости, обеспечивая воспроизводимость и переносимость ПО между машинами** г) Позволяет запускать код без операционной системы

28. Что такое "Параметр" (*Parameter*) в ROS? а) Аргумент командной строки при запуске ноды (`rosrun`) б) Константа, зашитая в исходном коде ноды в) Файл конфигурации формата JSON г) **Динамически изменяемое значение на сервере параметров ROS, которое могут читать ноды для своей настройки**

29. Какой инструмент используется для анализа логов (сообщений `ROS_INFO`, `ROS_WARN`)? а) rostopic echo б) rosservice call в) rqt_console / rqt_logger_level г) rviz

30. В чем ключевое архитектурное отличие между ROS 1 и ROS 2? а) В ROS 2 нет поддержки Python. б) ROS 2 работает только с реальными роботами, а не в симуляции. в) В ROS 2 используется централизованный мастер-узел. г) **В ROS 2 децентрализованная архитектура на основе стандарта DDS, где нет единой точки отказа.**