You completed this exam on *22/12/2022, 23:38*
Your score is 100.00%

CORRECT

Transforming Data using Choices

Which of the following choice types does not modify contract's data in Daml:

- *`nonconsuming` choice*
- `consuming` choice
- `transform` choice
- `update` choice

CORRECT

Choices and their Properties

Select all that are **true** for nonconsuming choices:

- *Nonconsuming choices can be denoted with the keyword nonconsuming*
- Nonconsuming choices change the contract's status from active to passive
- *Nonconsuming choices do not archive the contract automatically*
- Nonconsuming choices are a type of a fetch action

CORRECT

Templates

Select all that apply for templates in Daml:

can describe contracts' data restrictions

must have at least one party to be a signatory

are instances of a contract

must have an execution and expiration date set

CORRECT

Template Syntax

Which of the following templates is valid?

```
template A
 with
   p : Party
   i : Integer
   observer p
```

```
template B with
   p : Party
   b : Bool
 where
   signatory p
```

```
template C
 with
   p : Party
   d : Decimal
 where
   signatory t
```

```
template D with
   p : Party
 where
   signatory p
```

```
template E
 where
   p : Party
   t : Text
 with
   maintainer p
```

A

B

C

D

E

CORRECT

Parties and their Rights

Select the correct answer:

Only observers see when a contract has been created and archived

Only signatories see when a contract has been created and archived

Both signatories and observers see when a contract has been created and archived

---

CORRECT

Authorized Choices

Given the below Daml contracts, will the final submission in the script succeed?

```
module Main where

import Daml.Script

template Question
 with
   party1: Party
   party2: Party
   party3: Party
   content: Text
 where
   signatory party2, party3
   controller party1 can
     BlankOutContent: ContractId Question
       do create this with content = ""
   controller party1, party3 can
     ModifyContent: ContractId Question
       with newContent: Text
       do create this with content = newContent

template QuestionProposal
 with
   q: Question
 where
   signatory q.party2
   observer q.party1
   controller q.party3 can
     Accept: ContractId Question
       do create q

test : Script ()
test = do
  party1 <- allocateParty "Party1"
  party2 <- allocateParty "Party2"
  party3 <- allocateParty "Party3"
  p <- submit party2 do
    createCmd QuestionProposal
      with
        q = Question
          with
            party1, party2,party3, content=""
  q <- submit party3 do exerciseCmd p Accept
  submit party1 do
    exerciseCmd q ModifyContent
      with
        newContent = "Foo"
  pure()
```

Yes, every action in the script is authorized by all the required authorizers

No, because party1 does not have any rights on the final contract

No, because party3 has not authorized the final exercise in this context

No, party3 must also be defined as a maintainer in the template

CORRECT

Daml Application Components

Which of the following APIs is exposed by every ledger that runs Daml?

Ledger API

Deploy API

Upload API

LedgerDeploy API

LedgerUpload API

JSON API

React libraries

Ledger bindings (Java, Scala, NodeJS)

CORRECT

Recommended Daml Application Architecture

Arrange the components as they are in the recommended application architecture (from highest level/frontend components, to lowest level/backend components) of a full-stack Daml application.

React Application Code

Daml React libraries

Typescript Generated Code

JSON API Server

Participant Node

Daml Drivers

Synchronization Technology

CORRECT

Interacting with a Daml Ledger

Select all that apply when interacting with a Daml Ledger:

There is a time window in which the same command cannot be executed twice

Transaction's ledger time must match exactly the ledger's system time, otherwise transaction will be rejected

Each transaction is automatically assigned a ledger time by the participant server

In development environment requests sent to the ledger do not need to be authorized

CORRECT

Authentication and Authorization

When accessing a Daml Ledger in a production environment:

The Ledger API is used to authenticate users

The JSON API validates the authorization of the token

A third party service such as Auth0 can be used for access tokens if you want your Ledger API to require authorization

The Ledger API validates the authorization of the token

CORRECT

Ledger API Structure

Select all that apply for the Ledger API:

It is structured as a stream of commands to the ledger

It is structured as a stream of transactions and corresponding events from the ledger

Commands sent to the ledger are the only way an application can cause the state of the ledger to change

CORRECT

Ledger API Services

The Ledger API can be used to:

Bootstrap a Daml application with all the visible contracts that are active on a ledger

Reset the ledger state on a production ledger

Creating a new ledger instance

Submit commands to the ledger

CORRECT

JSON API Services

Select all that apply: The JSON API can be used to:

Create ledger parties

Querying the current active contract set on a ledger

Retrieving all known parties

Creating ledger instances

CORRECT

Daml TypeScript types

The @daml/types library contains TypeScript data types that correspond to (select all that apply)

Time data type

ContractId data type

Decimal data type

Bool data type

CORRECT

Interacting with a Daml Ledger via @daml/react library

With @daml/react library you can

query Daml contracts

create Daml contracts

exercise choices on Daml contracts

create ledger parties

create ledgers

communicate with the JSON API via @daml/ledger library

CORRECT

Interacting with a Daml Ledger

The command `ledger.fetchByKey(User.User, credentials.party);` uses the [✓ ***@daml/ledger***] library to make a call to the [✓ ***JSON API***]

CORRECT

JSON API Error messages

Select all that apply: the JSON API can return status codes indicating that:

the Ledger API cannot be initialized (500)

the endpoint was not found (404)

the exercise choice for a specific contract ID was successfully executed (200)

all known parties have been successfully fetched (200)

CORRECT

Daml Contract Data Types

Select all that apply: **On the frontend side**, Daml contract data types

are created when the Daml model is compiled to a DAR file

are deployed via the JSON API

are generated via the TypeScript code generator

are generated from data types declared in the deployed DAR

CORRECT

Daml Tooling

Select all that are **true** about Daml tools and their respective functionalities:

Daml Sandbox enables rapid application prototyping by simulating a ledger

Daml Navigator is a front-end application that allows viewing templates and active and archived contracts, as well as exercising choices on contracts

Daml REPL allows you test and manipulate a ledger interactively

Daml scripts are used for creating a ledger instance

CORRECT

Daml Assistant

Select all that apply: The following command `daml start --json-api-port=7899`

starts the JSON API on localhost

starts the Navigator on localhost

starts the Sandbox on port 6865

starts the JSON API on port 7899

starts the IDE

starts a node and connects is to the global testnet

CORRECT

Daml Sandbox

Select all that are true for Daml Sandbox:

uses MySQL DB for persistent storage by default

can be started with `daml start` command

can be started with `daml deploy sandbox` command

runs without authentication by default

CORRECT

Daml Script

Daml Script can be used to (select all that apply):

List known parties

Create a contract of a given template

Create a new ledger

Frontend and UI testing

CORRECT

The Navigator

Select all that are true for the Navigator

The Navigator needs to be installed with `daml install navigator` command

The Navigator can be started with `daml start` command

The Navigator can be used to view active contracts

The Navigator can be used to exercise choices on contracts

The Navigator can be used to view transaction details

CORRECT

Daml REPL

Daml REPL can be used to (select all that apply):

List known parties to a given participant

Ledger initialization

Allocate a party with a given display name and id hint

Create a contract with a specific id

Upload new DAML Packages to a Ledger

Delete a ledger

CORRECT

Deploying to a Ledger

Which of the following service(s) can be used to deploy a DAR file **to a running ledger:**

Ledger API

Deploy API

LedgerDeploy API

Upload API

LedgerUpload API

JSON API

Sandbox

Navigator

CORRECT

Daml SDK Tools to Interact with a Deployed Daml Ledger

What Daml SDK tools can you use to inspect and modify a deployed ledger:

Navigator

Daml Cube

Sandbox

Daml Script

CORRECT

Deploying to a Ledger via Daml Assistant - I

Which of the following commands can be used to deploy a Daml model:

daml upload

daml deploy

daml ledger upload-dar

daml distribute

daml post

CORRECT

Deploying to a Ledger via Daml Assistant - II

The following command `daml deploy --host localhost --port 7575` will

Start the Sandbox on localhost:7575 as a deployment ledger

Compile the current project to a DAR file

Deploy templates contained in the compiled DAR file

Deploy the UI via JSON API running on localhost:6865

Deploy the DAR to a ledger running on localhost:7575

Allocate the parties specified in the project configuration file

---

CORRECT

Interacting with a deployed Daml Ledger

The following command

`daml ledger upload-dar Bank.dar --host localhost --port 9000 --access-token-file /path/to/jwt` will:

- *Authenticate against a ledger with an access token*
- Allocate parties on a ledger running on localhost:6865
- Will use your client key (.pem) and certificate chain (.crt) files in mutual authentication process
- *Deploys the Bank.dar on a ledger running on localhost:9000*

CORRECT

Daml Supported Ledgers

Select all the ledgers where Daml can be deployed:

- *Hyperledger Besu*
- *PostgreSQL*
- Hyperledger Indy
- *Daml Hub*