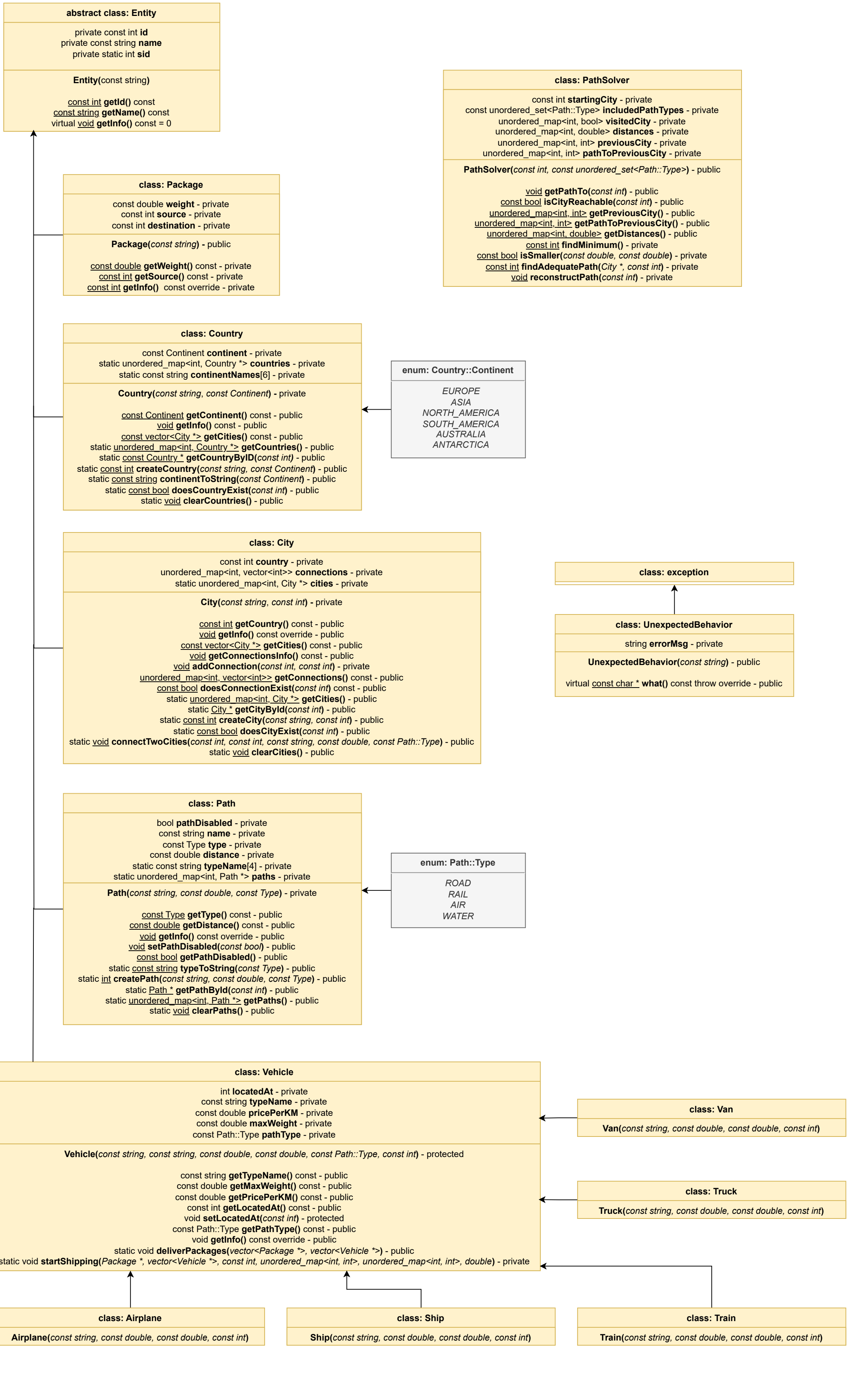
abstract class: Entity
private const int id
private const string name
private static int sid
Entity(const string)
const int getId() const
const string getName() const
virtual void getInfo() const = 0
class: PathSolver
const int startingCity - private
const unordered_set<Path::Type> includedPathTypes - private
unordered_map<int, bool> visitedCity - private
unordered_map<int, double> distances - private
unordered_map<int, int> previousCity - private
unordered_map<int, int> pathToPreviousCity - private
PathSolver(const int, const unordered_set<Path::Type>) - public
void getPathTo(const int) - public
const bool isCityReachable(const int) - public
unordered_map<int, int> getPreviousCity() - public
unordered_map<int, int> getPathToPreviousCity() - public
unordered_map<int, double> getDistances() - public
const int findMinimum() - private
const bool isSmaller(const double, const double) - private
const int findAdequatePath(City *, const int) - private
void reconstructPath(const int) - private
class: Package
const double weight - private
const int source - private
const int destination - private
Package(const string) - public
const double getWeight() const - private
const int getSource() const - private
const int getInfo() const override - private
class: Country
const Continent continent - private
static unordered_map<int, Country *> countries - private
static const string continentNames[6] - private
Country(const string, const Continent) - private
const Continent getContinent() const - public
void getInfo() const - public
const vector<City *> getCities() const - public
static unordered_map<int, Country *> getCountries() - public
static const Country * getCountryById(const int) - public
static const int createCountry(const string, const Continent) - public
static const string continentToString(const Continent) - public
static const bool doesCountryExist(const int) - public
static void clearCountries() - public
enum: Country::Continent
EUROPE
ASIA
NORTH_AMERICA
SOUTH_AMERICA
AUSTRALIA
ANTARCTICA
class: City
const int country - private
unordered_map<int, vector<int>> connections - private
static unordered_map<int, City *> cities - private
City(const string, const int) - private
const int getCountry() const - public
void getInfo() const override - public
const vector<City *> getCities() const - public
void getConnectionsInfo() const - public
void addConnection(const int, const int) - private
unordered_map<int, vector<int>> getConnections() const - public
const bool doesConnectionExist(const int) const - public
static unordered_map<int, City *> getCities() - public
static City * getCityById(const int) - public
static const int createCity(const string, const int) - public
static const bool doesCityExist(const int) - public
static void connectTwoCities(const int, const int, const string, const double, const Path::Type) - public
static void clearCities() - public
class: exception
class: UnexpectedBehavior
string errorMsg - private
UnexpectedBehavior(const string) - public
virtual const char * what() const throw override - public
class: Path
bool pathDisabled - private
const string name - private
const Type type - private
const double distance - private
static const string typeName[4] - private
static unordered_map<int, Path *> paths - private
Path(const string, const double, const Type) - private
const Type getType() const - public
const double getDistance() const - public
void getInfo() const override - public
void setPathDisabled(const bool) - public
const bool getPathDisabled() - public
static const string typeToString(const Type) - public
static int createPath(const string, const double, const Type) - public
static Path * getPathById(const int) - public
static unordered_map<int, Path *> getPaths() - public
static void clearPaths() - public
enum: Path::Type
ROAD
RAIL
AIR
WATER
class: Vehicle
int locatedAt - private
const string typeName - private
const double pricePerKM - private
const double maxWeight - private
const Path::Type pathType - private
Vehicle(const string, const string, const double, const double, const Path::Type, const int) - protected
const string getTypeName() const - public
const double getMaxWeight() const - public
const double getPricePerKM() const - public
const int getLocatedAt() const - public
void setLocatedAt(const int) - protected
const Path::Type getPathType() const - public
void getInfo() const override - public
static void deliverPackages(vector<Package *>, vector<Vehicle *>) - public
static void startShipping(Package *, vector<Vehicle *>, const int, unordered_map<int, int>, unordered_map<int, int>, double) - private
class: Van
Van(const string, const double, const double, const int)
class: Truck
Truck(const string, const double, const double, const int)
class: Airplane
Airplane(const string, const double, const double, const int)
class: Ship
Ship(const string, const double, const double, const int)
class: Train
Train(const string, const double, const double, const int)