Class	Type
IEncoderSensor.h	Interface
IGyroSensor.h	Interface
EncoderSensor	Hardware component object
GyroSensor	Hardware component object
MathUtils	Math helper
Vector2D	Vector object
Odometry	Logic
Pose	Data object
SwerveDriveKinematics	Logic
SwerveModule	Hardware component object
ChassisState.h	Data object
WheelModuleState.h	Data object
Robot	Main robot class

Description
Abstract interface for position sensors
Abstract interface for angular orientation sensors
Implementation of IEncoderSensor for NEO motors
Implementation of IGyroSensor for NavX gyroscope
Helper functions for math operations
Represents 2D vectors for positions, velocities and offsets
Tracks robots position using encoders and a gyro
Stores robot pose (position + heading) and history
Converts chassis velocity --> individual wheel states(and also other way around)
Controls one wheel (speed and steering)
Represents robot chassis motion
Represents single wheel state
WPILib robot, intergrates subsystems and handles TeleOp/Autonomous

Responsibilities
Define a standard way to read distance and velocity from any encoder Define a standard way to read heading and angular velocity from any gyroscope
Read and convert motor encoder counts to units (meters, m/s), provides a reset functionality Read robot heading, converts to rad, allows resetiing the heading
Provides reusable math operations Represents positions and velocities, supports vector math operations
Update robot pose based on encoder data and gyro heading, provides current pose for other subsystems Keep track of robot position and heading, store history for debugging
Translate desired chassis motion in individual wheel speeds/angles, supports field-relative driving Controls a single wheels speed and angle converts physical units to motor commands Store robot translational and rotational velocity Store individual wheel speed and angle for swerve control
Coordinate subsystems, read joystick inputs, updates odometry, calculates wheel states and send commands to swe

erve modules