- AUTomotive Open System ARchitecture -

Đorđe Milićević

Šta je AUTOSAR?

- Zajednica koja se bavi razvojem standarda koji se koriste u automobilskoj industriji
- Osnivači: **BMW**, **Bosch**, **Continental**, **DaimlerChrysler**, **Siemens** i **Volkswagen**
- Osnovana u julu 2003. godine
- Primarni cilj obuhvata standardizaciju softverske arhitekture za razvoj elektronskih jedinica koje upravljaju radom odgovarajućih sistema unutar vozila:
  - modul za kontrolu motora (Engine Control Module)
  - modul za kontrolu kočionog sistema (Brake Control Module)
  - modul za kontrolu menjača (Transmission Control Module)
  - modul za kontrolu stabilnosti (Suspension Control Module)
  - modul za kontrolu šasije (Body Control Module)
- Moderni automobili poseduju do 80 upravljačkih modula

AUTOSAR C++14 standard

- Propisan veliki broj pravila kodiranja za korišćenje programskog jezika C++ u oblasti sigurnosnih i vremenski kritičnih sistema
- Primarni sektor je automobilska indrustrija, ali se pravila mogu koristiti i u drugim sektorima za razvoj ugrađenih sistema
- Određena pravila nasleđena su iz MISRA (Motor Industry Software Reliability Association) C++:2008 standarda
- Moguća provera velikog broja pravila korišćenjem statičke analize
- Neka pravila je nemoguće automatizovati i zahtevaju ručnu proveru
- Razlog nastanka?

AUTOSAR C++14 standard

- Standard propisuje 402 pravila:
  - 148 pravila usvojeno je iz MISRA C++:2008 standarda (64%)
  - 254 nova pravila (194 izvedeno iz postojećeg C++ standarda i 54 pravila zasnovana na istraživačkim radovima)
- Određena pravila se mogu preklapati sa ostalim pravilima
- U okviru Clang-a trenutno je podržano 26 pravila
- Trenutna ograničenja:
  - skup pravila za paralelno izračunavanje nije podržan
  - skup pravila za standardne C++ biblioteke delimično je podržan
  - skup pravila za opštu bezbednost nije podržan

Klasifikacije pravila

- Različite klasifikacije pravila
- Klasifikacija prema nivou važnosti:
  - **obavezna** (357 pravila)
  - **savetodavna** (33 pravila)
- Klasifikacija prema primenljivosti statičke analize:
  - **automatizovana** (324 pravila)
  - **delimično automatizovana** (20 pravila)
  - **neautomatizovana** (46 pravila)
- Ciljna klasifikacija:
  - **implementaciona** – programski kod, softverski dizajn i arhitektura
  - **verifikaciona** – testiranje, analiza i pregled koda
  - **infrastrukturna** – kompajler, linker, biblioteke
  - **pravila koja se odnose na alate** – operativni sistem i hardver

Primeri automatizovanih pravila

- Projekat ne sme da sadrži mrtav kod
- Projekat ne sme da sadrži neiskorišćene promenljive i deklaracije tipova
- Svaka definisana funkcija mora biti pozvana bar jedanput
- Tipovi podataka **long double** i **wchar_t** se ne smeju koristiti
- Projekat ne sme da sadrži kompajlerska upozorenja
- Identifikator deklarisan u unutrašnjem dosegu ne sme da prikriva identifikator deklarisan u spoljašnjem dosegu (konflikt identifikatora)
- Ključna reč **volatile** se ne sme koristiti
- Heksadekadne konstante i sufiksi literala moraju biti napisani velikim slovima
- Trivijalne funkcije za pristup i promenu vrednosti objekata moraju biti inline-ovane
- **NULL** se ne sme koristiti kao celobrojna vrednost
- Jedino se **nullptr** sme koristiti kao *null* pokazivačka konstanta
- Vrednost izraza mora biti ista bez obzira na redosled evaluacije operanada
- **dynamic_cast** se ne sme koristiti

Primeri automatizovanih pravila

- Tip **char** može se koristiti samo za čuvanje karakterskih konstanti
- Uslovni izraz **if** naredbe mora biti tipa **bool**
- Način kojim se adresiraju nizovi mora da bude jedini oblik pokazivačke aritmetike
- Bitovski operatori se mogu primenjivati jedino na neoznačene operande
- Lambda izraz ne bi trebalo da bude definisan u okviru drugog lambra izraza
- Tradicionalne C-ovske konverzije se ne smeju koristiti
- Operatori inkrementiranja i dekrementiranja se ne smeju pojavljivati zajedno sa drugim operatorima u okviru istog izraza
- Operandi operatora **!**, **&&** i **||** moraju biti tipa **bool**
- Ternarni operator se ne sme koristiti u podizrazima
- Unarni operator **–** se ne sme primenjivati na izraze neoznačenog tipa
- Operator zarez se ne sme koristiti
- Operatori dodele se ne smeju koristiti u podizrazima
- Svaka višestruka **if/else** naredba mora se završavati **else**-om

Primeri automatizovanih pravila

- Uslovni izraz **switch** naredbe ne sme biti tipa **bool**
- Petlja **do/while** se ne sme koristiti
- Naredba **goto** se ne sme koristiti
- Završna klauza u okviru **switch** naredbe mora biti **default**
- Ključna reč **register** se ne sme koristiti
- Direktiva **using** se ne sme koristiti
- Deklaracija **asm** se ne sme koristiti
- Funkcije deklarisane sa **[[noreturn]]** atributom ne smeju vraćati vrednost
- Funkcija ne sme da poziva samu sebe (direktno ili indirektno)
- Unije se ne smeju koristiti
- Deklaracije **friend** se ne smeju koristiti
- Direktiva **#pragma** se ne sme koristiti
- Pseudoslučajni brojevi se ne smeju generisati **std::rand()** funkcijom
- **std::auto_ptr** se ne sme koristiti

Primeri delimično automatizovanih i neautomatizovanih pravila

- Izrazi u pokretnom zarezu se ne smeju implicitno (ili eksplicitno) jednakosno porediti
- Operatori **new** i **delete** ne smeju biti eksplicitno korišćeni
- Objekti koji ne mogu da nadžive funkcije u kojima su uvedeni moraju imati automatski životni vek
- Korišćenje aritmetike u pokretnom zarezu mora biti dokumentovano
- Nivo upozorenja procesa prevođenja mora biti podešen u skladu sa politikom projekta
- Metrike koda moraju biti definisane i poštovane u toku razvoja projekta
- Svaka upotreba asemblera mora biti dokumentovana
- Javno nasleđivanje se mora koristiti za implementaciju **is-a** odnosa
- Funkcija se ne sme završiti izuzetkom ako je sposobna da u celosti obavi svoj zadatak
- Objektu se ne sme pristupati izvan njegovog životnog veka

Literatura

- **MISRA C++:2008 Guidelines for the use of the C++ language in critical systems**
- **AUTOSAR Guidelines for the use of the C++14 language in critical and safety-related systems**
- **AUTOSAR - Wikipedia**