

Nya tjänster • Nya tjänster •



Lena Jämtgren, med placering i Solna och tidigare assistent till Göran Persson, är sedan april assistent till Mikael Leksell i hans roller som vd för Siemens i Sverige, Norge och Nordics samt Digital Industries-chef.

lena.jamtgren@siemens.com



Jessica Boqvist, tidigare vd-assistent, är sedan april assistent till Andreas Hillerström, Head of Finance Digital Industries i Sverige och Nordics, samt support för Digital Industries-organisationen med placering i Solna.

jessica.boqvist@siemens.com



Fredrik Hänninen anställdes i augusti i Jönköping som utbildare inom Sitrain utbildningscenter inom Digital Industries. Fredrik kommer närmast från Beijer Industri och har tidigare arbetat på Siemens.

fredrik.hanninen@siemens.com



Mårten Thelin anställdes i augusti i Linköping som Proposal Specialist Process Automation inom Digital Industries. Mårten kommer närmast från Saab och har tidigare arbetat på Siemens.

marten.thelin@siemens.com

siemens.se/job

Boka in • Boka in • Boka in • Boka in • Boka in •



SITRAIN utbildningscenter

Nya utbildningar inom SIMATIC WinCC Unified

Om du jobbar med att skapa och/eller ändra i HMI-projekt finns nu vår nya utbildning Simatic WinCC Unified & Unified Comfort Panels (TIA-UWCCM) om du vill lära dig grunder och konfiguration av Simatic WinCC Unified och Unified Comfort-paneler.

Om du även skapar/ändrar HMI-projekt för Simatic WinCC Unified PC Runtime är utbildningen Simatic WinCC Unified för pc, avancerad kurs (TIA-UWCCPC) ett naturligt fortsättningssteg för att lära dig mer om Simatic WinCC Unified i SCADA-miljön.

I den digitala utbildningsplattformen Sitrain access finns den webb-baserade utbildningen Advanced Simatic WinCC Unified Scripting (JavaScript) (TIA-UWCCJA) om du vill lära dig mer om att använda Javascript i Simatic WinCC Unified. ■

siemens.se/sitrain

siemens.com/sitrain-access

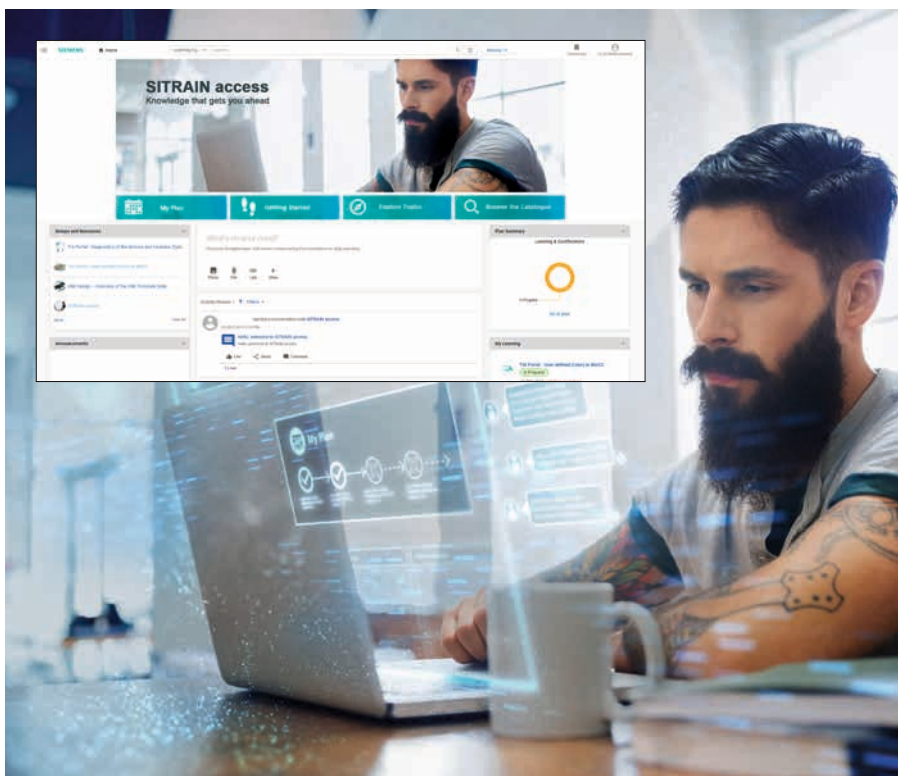
SITRAIN access – ny digital utbildningsplattform

I våras lanserades vår nya digitala utbildningsplattform Sitrain access. Denna digitala utbildningsplattform ger dig möjlighet att lära dig nya saker och utveckla din kompetens när och var det passar dig via dator, surfplatta eller smarttelefon.

Sitrain access passar alla oavsett förkunskapsnivå och innehåller webbaserade utbildningar, expert talks, kortare learning nuggets, tester för att stämma av din kunskap, övningar, forum med möjlighet att bolla med andra deltagare med mera. Som chef kan du få en överblick över dina anställdas genomförda utbildningar och kan rekommendera utbildningar som verkar vara bra.

Sitrain access tecknas som ett personligt abonnemang per deltagare och är giltigt ett år om du inte väljer att förlänga abonnemang. ■

siemens.com/sitrain-access



Boka in • Boka in • Boka in • Boka in • Boka in • Boka in • Boka in • Boka in • Boka in •



SITRAIN utbildningscenter

Vill du lära dig OPC-UA?

Vill du lära dig grunder, terminologi och datamodeller kring OPC UA i automationsmiljö?

Nu finns utbildningen OPC UA – Basics and Configuration (IK-OPCUA1). I denna lär du dig att förstå interaktionen mellan de olika komponenterna och att konfigurera de viktigaste OPC UA-servrarna och -klienterna i Simatic-produktportföljen.

 siemens.se/sitRAIN

SITRAIN utbildningscenter

SIMIT i diskret automationsteknik för nybörjare

Simit möjliggör simulering av maskin- och anläggningsbeteende för virtuell idrifttagning.

Med hjälp av komponenterna integrerade i Simit kan du testa plc-kod och etablera nya produkter på marknaden mycket snabbare.

Utbildningen Simit i diskret automationsteknik för nybörjare (DI-SIMITFA) ger dig en översikt över simuleringsmjukvaran Simit's funktioner och bibliotek. Med praktiska exempel lär du dig steg för steg hur du utformar simuleringar/modeller för testning av automationsmjukvara. Målet med utbildningen är att du i Simit ska kunna skapa dina egna komponenter och mallar, skapa en simulering för plc-programmet och ställa in kopplingarna mellan simuleringen i Simit och Simatic S7-PLCSim Advanced. ■

 siemens.se/sitRAIN



SITRAIN utbildningscenter

Nya SINAMICS-utbildningar

Vill du lära dig mer om idrifttagning, service och underhåll av Sinamics S120 med inriktning mot servodrifter? Nu finns två olika möjligheter beroende på vad du använder för mjukvara.

Om du vill lära dig mer om idriftsättning av Sinamics S120 med mjukvaran Startdrive i TIA Portal finns nu utbildningen Sinamics S120 parametrering och idriftsättning/service i TIA Portal (DR-S12-PMT). Om du använder Starter rekommenderas vår utbildning Sinamics S120 startup & service (DR-SNS-SIS).

Om du använder Sinamics G120 går vi under en dag igenom vad som krävs för konfigurering och initial uppstart för en Sinamics G120 i utbildningen Sinamics G120 startup & service (DR-G120). ■

 siemens.se/sitRAIN



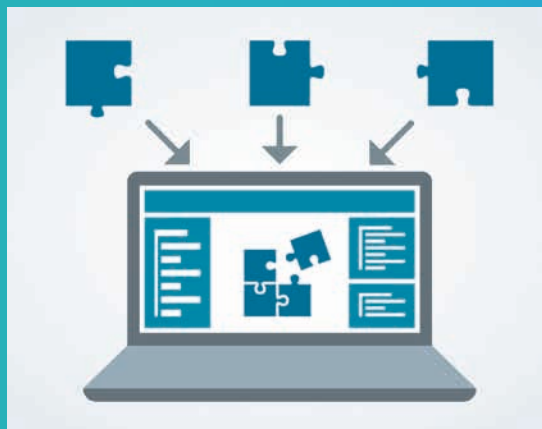
SITRAIN utbildningscenter

TIA Portal Openness Programming 1

TIA Portal Openness är API eller programmeringsgränssnitt som tillhandahålls med TIA Portal, med vilket du effektivt kan automatisera vissa uppgifter utan att använda TIA Portal själv. Detta inkluderar uppgifter som projekthantering, konfiguration och parametrering av hårdvara, automatisk generering av blockkod samt olika onlinefunktioner.

Utbildningen TIA Portal Openness Programming 1 (DI-OPEN1) ger en introduktion kring att programmera med C#. Du kommer att lära dig grunderna i C# och prestandan för Openness API i Visual Studio och hur du använder programmeringsspråket C#. Målet med utbildningen är att du ska lära dig att programmera med Visual Studio och programmeringsspråket C# och skapa mindre applikationer med TIA Openness.

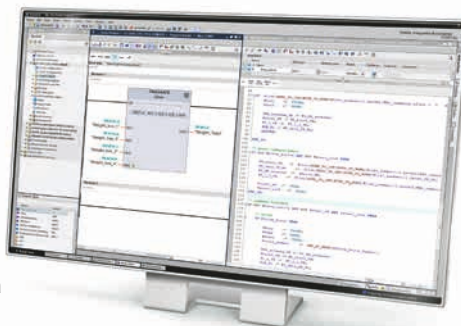
 siemens.se/sitrain



SITRAIN utbildningscenter

Lär dig SCL!

Genom att använda Structured Control Language (SCL) istället för Statement List (STL) eller LAD/FBD (ladder/funktionsblockdiagram) kan du minska tiden för att skapa och underhålla program. Vi har därför två nya separata utbildningar för SCL.



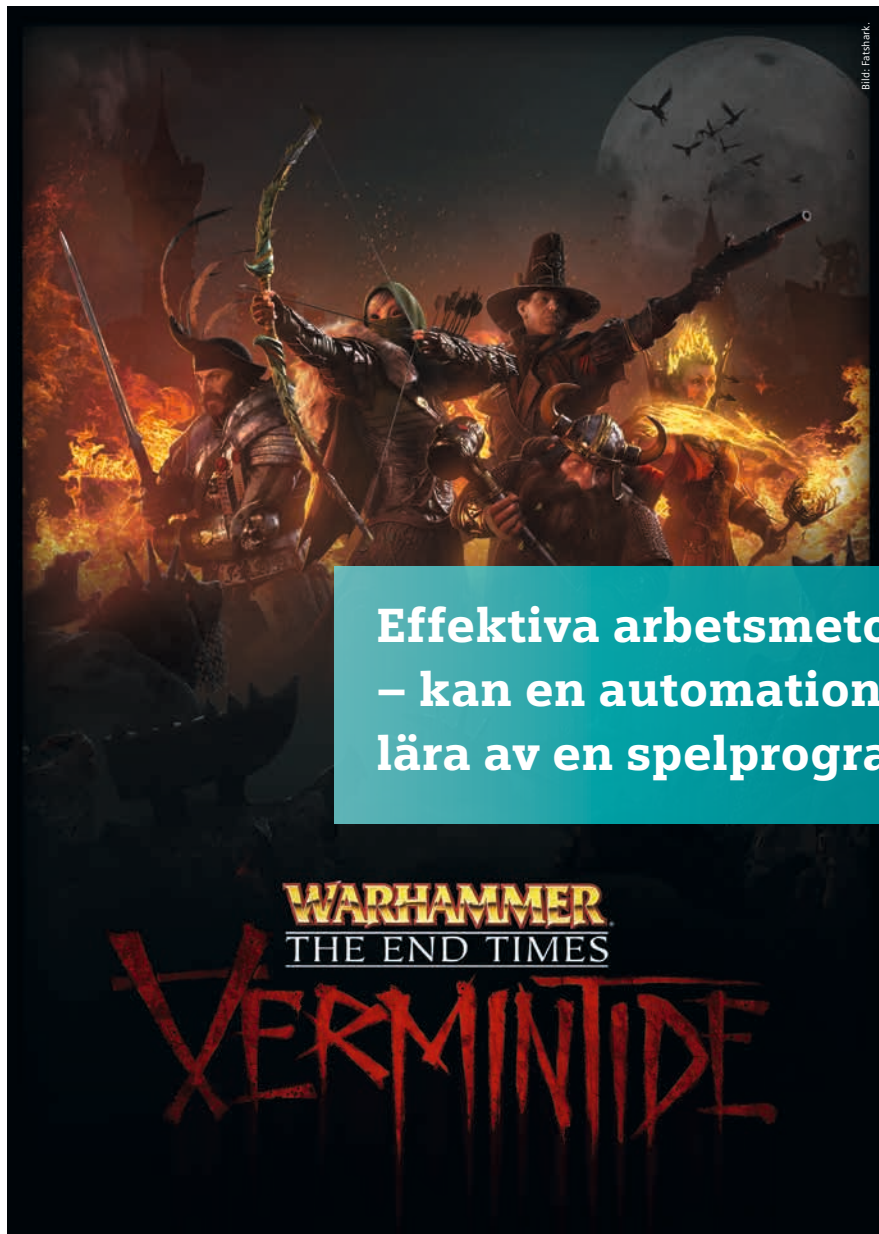
I utbildningen Simatic programmering 1 med S7-SCL i TIA Portal (TIA-SCL1) får du lära dig grunderna och fördelarna med att använda Structured Control Language (SCL). Du får bland annat lära dig att skapa grundläggande SCL-program, testa dessa och att genomföra diagnostik i SCL-block. I utbildningen Simatic programmering 2 med S7-SCL i TIA Portal (TIA-SCL2) kommer du att utveckla dina kunskaper och skapa mer komplexa SCL-program. ■

 siemens.se/sitrain

Webbinarier

Passa på att ta del av information som kan effektivisera ditt arbete, fördjupa dig inom olika teman och hör våra experter berätta om produktnyheter och smarta tips. Välj själv: lyssna direkt eller se dem i efterhand när det passar dig bäst!

 siemens.se/webbinarier



Effektiva arbetsmetoder – kan en automationsprogrammerare lära av en spelprogrammerare?

Simon Lundmark, vd och programmerare på Pixeldiet Entertainment.



En automationsingenjör jobbar vanligtvis själv med en lösning, är ofta den enda som ser koden, jobbar mestadels lokalt i en kopia, sällan med versionshantering och testar först när det är idriftsättning. Detta gäller inte alla automationsingenjörer men denna typ av arbetsmetod är inte ovanlig och förekommer kanske särskilt på mindre bolag inom diskret tillverkningsindustri. Kraven på att det ska bli rätt och gå snabbt att få fram en lösning ökar men frågan är om arbetsmetoderna har hunnit med för att stödja de allt hårdare kraven från industrin.

För Simon Lundmark, vd och programmerare på spelutvecklingsbolaget Pixeldiet Entertainment, låter detta vansinnigt. Spelbranschen är ganska många snäpp bättre på att nyttja system för samarbete och versionshantering och att använda effektiva arbetsmetoder för att hantera risker och testa kod.

Automationsingenjörer är inte ovana vid tidspress och att axla ansvar. De kommer ofta sist i utvecklingen av en ny produktion och har stort ansvar på individnivå för att få saker att fungera. Ofta sitter automationsingenjören själv med sin kod utan någon annan som granskar den.

Samtidigt ställs mer och mer krav på att produktionen ska kunna ställas om oftare, och snabbare. Detta ställer stora krav på automationsingenjören och dennes arbetsmetoder. Det kanske är dags att börja snegla på hur man har åstadkommit den snabba mjukvaruutveckling man ser i andra

programmerande yrkesgrupper? Idag finns det mängder av projekt som programmeras av hundratals programmerare runtom i världen. De har dessutom ofta krav på omedelbara resultat; snabbhet i att lansera en ny lösning kan vara skillnaden i att vara marknadsledare eller bli utkonkurrerad helt. När kraven är så stora på lösningen är kraven på arbetsmetoderna lika stora.

Lösningar för att samarbete. Nyckeln till framgång är ofta att nyttja system för samarbete, versionshantering och testning. De lösningar som finns för samarbete handlar delvis om att hantera uppgifter och distribuera dessa men även om att kunna dela kod mellan olika personer, även på helt olika geografiska platser.

I vissa projekt har vem som helst möjlighet att ladda ned kod från en kod-delningssajt, som GitHub, utveckla nya funktioner och be ägaren att implementera dessa i programmet. Just ett sådant projekt är kärnan i alla så kallade Linuxbaserade operativsystem, Linux Kerneln, där vem som helst kan bidra och be om att få denna funktion eller ändring in i kerneln. Om Linus Torvalds tillåter förstås. Denna koddelning möjliggör alltså att flera olika programmerare kan jobba samtidigt och alltid få tag på den senast testade versionen.

Forts. nästa sida

GitHub ses främst som ett versionshanteringsprogram, precis som Subversion eller Microsoft Team Foundation Server. Varje gång en ändring görs skapas ett bibliotek av tidigare versioner, som gör att man kan "rulla tillbaka" en uppdatering om programmet skulle få en bugg. Ändringar dokumenteras flitigt så det är lätt att hitta ändringar och även att kunna se tillbaka på gamla lösningar för att återanvända i senare projekt. Man ser vem som har skrivit varje rad kod och man har samtlig historik över alla förändringar som har skett i kodbasen.

Dessa system (GitHub, Subversion eller Microsoft Team Foundation Server) kan även användas för TIA Portal-projekt, så versionshantering och samarbete finns det även här möjligheter till. Andra lösningar som Multiuser Engineering kan också göra att man uppnår versionshantering. Mellan ingenjördiscipliner kan man även använda lösningar som Teamcenter, där man kan koppla ett visst TIA Portal-projekt till dess elschema och CAD-ritning och hela tiden få tag på den senaste uppdateringen när ändringar sker. Detta möjliggör ju ännu mer samarbete.

Riskhantering och kodtestning. Den andra delen i att effektivisera arbetsmetoder handlar om hur man hanterar risker och testar sin kod. Det finns sedan en tid tillbaka faktiskt arbetsmetoder som går ut på att man skapar tester innan man skapar lösningen. Det gör att man måste ha en bra analys och fundering på vad systemet måste klara, alla möjliga fel, alla möjliga kombinationer av fel och även besluta hur dessa bör testas. Inte sällan är ingenjören som skriver testerna inte samma ingenjör som gör lösningen, detta för att kunna säkerställa att inga genvägar tas eller kanske ännu mer, att den mänskliga faktorn spelar in, och något missas.

Hur man testar ett program är oerhört beroende av hur detta program ska nyttjas men generellt ska ett program både kunna uppfylla ett mål med applikationen men även kunna hantera när systemen runtomkring inte beter sig som väntat och kunna ge bra felmeddelanden när detta händer. Systemet ska alltså genomgå både positiv och negativ testning.

Idag finns det enkla system för testning. Simatic S7-PLCSim, PLCSim Advanced och Simit gör att man enkelt kan kolla att kodblocken fungerar och



testa systemet på signalnivå. För att även kunna gå vidare i testningen kan man nyttja virtuell idriftsättning, där man även kopplar in en visualisering. Detta möjliggör att man även ser själva produkten som åker igenom produktionen, verifierar att saker händer i rätt ordning med mera.

En spelprogrammerares vardag. Simon Lundmark är medgrundare och vd för Pixeldiet Entertainment som sysslar med programmering inom spelutveckling för dator- och konsolspel. Pixeldiet har funnits i nio år och har medverkat till att utveckla spel som Magicka: Wizard Worlds (Paradox North) och Warhammer: Vermintide (Fatshark).

Förutom att vara vd är Simon Lundmark själv senior programmerare. Programmeringen sker främst i C och C++ beroende på prestandakrav, där en spelmotor, som kan delas, hanterar det som är maskinnära. Gamekoden kan vara i annat språk (Lua, C#).

– Ett större spel tar tre till fem år att utveckla och kan ha en budget på 200 miljoner upp till 2,5 miljarder kronor. Det största projekt jag har jobbat i involverade 800 personer varav 200 programmerare.

De involverade i ett projekt behöver inte sitta på samma kontor eller ens i samma land. Ubisoft Entertainment,

ett av världens största spelbolag, har kontor över hela världen. Vissa utvecklingsbolag har inte ens kontor utan arbetar bara online.

De senaste fyra åren har han arbetat i ett projekt för Arrowhead Game Studios, som gör spel för Playstation och dess underkonsoler. 65 personer jobbar i det projektet, varav 12 programmerare.

– Tusentals system interagerar med varandra vilket gör att helheten blir komplex och små förändringar får stora följder. Jag har programmerat i över 20 år och är det något jag har lärt mig är det att det är viktigt att göra koden så enkel som möjlig att ändra och att testa så ofta som möjligt. Det ger resultat.

Simon Lundmark arbetar mest i Visual Studio men också i Slack för kommunikation, Jira för buggar och uppgifter och HG och GitHub för versionshantering.

Utmaningen är att det inte går att göra ett spel enligt vattenfallsmodell, den systemutvecklingsprocess som följer ett logiskt flöde genom olika faser. I spelvärlden måste koden kunna anpassas jättesnabbt efter kravbeställarnas behov.

– Här pratar vi verkligen om en agil utvecklingsmetod. Vår projektmetod kan ändras snabbt och iterationstiden blir väldigt viktig. Spelutvecklarna och programmerarna måste kunna

En scrummodell nyttjas ofta då samarbete är essentiellt.

- 1 Spelprogrammeraren får en uppgift via Jira
– dokumenterar vad man ska göra.
Innan man börjar ges feedback på uppgiften
– kommer överens om uppgiften.
- 2 Source versioning control tools GitHub, Subversion.
Synkar motorn från GitHub.
Synkar gamekod från HG Mercurial.
- 3 Bygger motorn.
Kompilerar content.
Startar spelet.
- 4 Börjar koda och lösa uppgiften.
Sparar och kompilerar om den delen.
Laddar om och ser den delen (iterativt).
- 5 Checkar in saker (motor, i gamekoden).
- 6 Synkar med kvalitetssäkringsgrupp
(testar spelet dagligen).
- 7 TeamCity kompilerar till alla plattformar.

göra ändringar snabbt i spelen och ägarna måste snabbt kunna se ändringarna realiserade. Vi talar om timmar eller maximalt dagar. Om det skulle ta tre månader att testa om fienden exploderar på rätt sätt skulle det vara alldeles för lång tid.

Smarta verktyg som underlättar din arbetsmetodik när du jobbar i TIA Portal

Nyttja Multiuser Engineering i TIA Portal för att enkelt samarbeta i ett projekt och även hålla koll på versioner: support.industry.siemens.com/cs/se/en/view/109740141

Nyttja TIA Portal Openness för att spara dina projekt på GitHub, Subversion eller Microsoft Team Foundation Server: support.industry.siemens.com/cs/se/en/view/109770552

Standardisera med hjälp av bibliotek: support.industry.siemens.com/cs/document/109747503/guideline-on-library-handling-in-tia-portal?dti=0&lc=en-SE

Utför testning och Virtual Commissioning med hjälp av Simatic Machine Simulator och samarbete med andra ingenjördiscipliner: siemens.com/virtual-commissioning

Utbildningar

Få utbildning inom hur du kan jobba med standardisering (DI-STAND), hur du kan generera istället för programmera/konfigurera (DI-AUTOEN & DI-OPEN2) samt hur du kan arbeta med testning (DI-SIMITFA & DI-VIRTCOM): sitrain-learning.siemens.com/SE/sv/rw59772/Digital-Enterprise

Vad är GitHub?

youtube.com/watch?v=w3jLUJ7DT5E

Vikten av versionshantering. Källkodsversionshantering är självklar inom spelvärlden. På programmerarnas arbetsdatorer laddas de senaste förändringarna ned.

– När jag laddar ned förändringarna är det bara förändringarna som kompileras. Versionshantering kör jag alltid med från dag ett. Skulle jag inte ha en server skulle jag ta en Raspberry Pi så att jag inte har all kod på ett ställe. Om man inte har källkodsversionshanteringssystem är det helt bedrövligt. Får du krångliga buggar i koden måste du kunna se var det buggade. Jag kan hitta exakt vilken filändring det var som orsakade en krasch. I systemet skriver jag också ned vad jag har gjort och varför. När jag sedan tittar på kod som jag inte förstår kan jag läsa vad jag höll på med när jag gjorde den ändringen och förstå hur jag tänkte. Källkodsversionshantering innebär att alltid ha koll på historiken och kommentarerna i versionshanteringen hjälper till att förstå koden.

Lärande och stödjande kultur. Simon Lundmark lyfter fram vikten av att ha en kultur där man lär av varandra och driver varandra att bli bättre.

– Jag kollar varje morgon i vår versionshantering och kan se fel som de andra har gjort och berätta hur de ska göra istället. Det blir en kvalitetssäkring och ett sätt för kollegor att granska och hjälpa varandra. Det blir också ett sätt för seniora programmerare att hjälpa unga programmerare att utvecklas. Det blir en positiv kultur för alla: när man ser ett problem kontaktar man den person som jobbade med det och löser uppgiften tillsammans.

Vad säger han då om att programmera som en automationsingenjör?

– Att sitta själv med sin dator utan testmiljö? Jisses. Hade jag inte haft en skulle jag ha börjat med att sätta upp någon form av testmiljö. Programmering handlar om att ta input, göra något magiskt och få en output. Om jag inte kan testa outputen kan jag inte göra något magiskt.

Automationsprogrammering behöver förvisso inte ändras lika ofta och koden heller inte testas lika ofta. Då kan man göra unit-tester, som i automationsvärlden innebär att kompilera koden. Men att inte få visuell feedback direkt, hela tiden, ter sig märkligt för Simon Lundmark.

– Det skulle vara konstigt att sitta och jobba på något som jag tror och hoppas är en optimal version utan att veta.

Att jobba ensam ser han väldigt få fördelar med.

– Du har ingen annan som granskar koden!

Och att jobba utan versionshantering:

– Bedrövligt! Det blir mycket svårare att hitta problem.

Visst är förhållandena i spelvärlden kanske anorlunda men eftersom det blir alltmer komplext även i automationsvärlden ökar behovet av snabb och kontinuerlig feedback, korta idriftsättningar och kort time-to-market. Vi kan ha en del att lära. ■

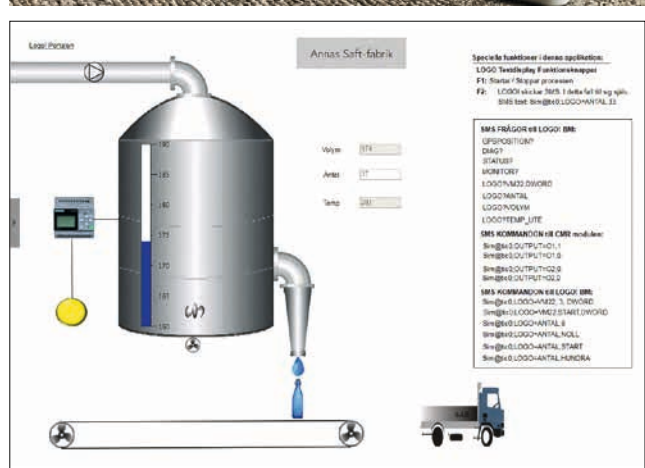
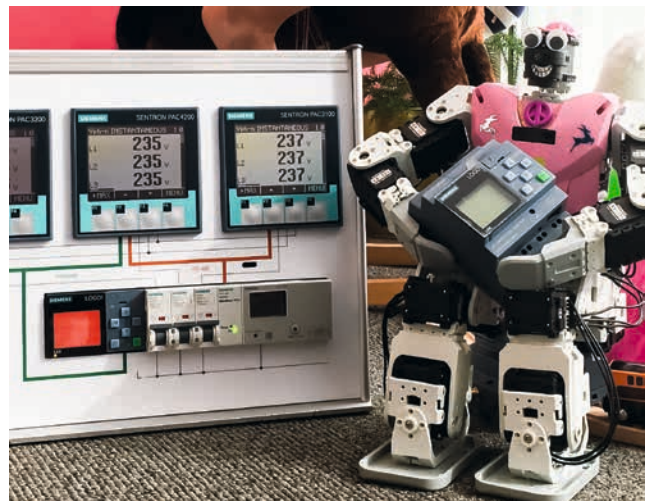
 siemens.se/industri
 elin.nordmark@siemens.com

Pixeldiet Entertainment AB är ett spelutvecklingsbolag med kontor i Stockholm. Företaget startades 2011 av Simon Lundmark och Fredrik Engkvist.

 pixeldiet.se



Framtidens programmerare: Inez Wirefors, Anna Jonsson och Polly Lindmark.



Lekande lätt med LOGO!

Lätt som en plätt att programmera är logikmodulen Logo.

Med logikmodulen Logo går programmeringen lekande lätt. Anna Jonsson, 16 år, lärde sig Logo på 3-4 dagar, som ett valfritt uppdrag hon valde första året på linjen Teknik, design och produktutveckling på Thorildsplans gymnasium i Stockholm.

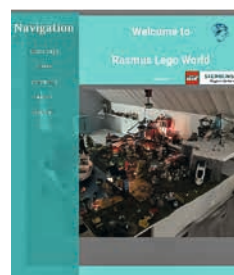
Första dagen lärde sig Anna Jonsson grunderna, andra dagen programmerade hon simulering av tank och flaskfyllning samt synkroniserad styrning av transportbana. Tredje dagen designade hon webbgränssnittet för att visualisera och kontrollera processen

och fjärde dagen konfigurerade hon GSM/SMS-modulen för övervakning, larm och kontroll av Logo via sms.

Rasmus Åberg, 12 år, fick till en Logoapp för att styra sin Legovärld med hjälp av appen. Lamporna i Legohuset styrs under vissa tidsintervaller. Nästa projekt för Rasmus Åberg blir att bygga en webbsida med hjälp av Logo Web Editor och att automatisera skidliften och järnvägsövergången.

Vill du också prova lekande lätt programmering? Läs mer på siemens.com/logo.

lars.jonsson@siemens.com



Rasmus Åberg styr sin Legovärld med en Logoapp.

