EXAMENSARBETE INOM TEKNIK OCH LÄRANDE,
AVANCERAD NIVÅ, 30 HP
STOCKHOLM, SVERIGE 2022

Automatisk återkoppling på programmeringsuppgifter

Undersökning och utveckling av hur automatisk återkoppling kan användas för att främja lärande

Hera Dalianis

KTH
SKOLAN FÖR INDUSTRIELL TEKNIK OCH MANAGEMENT

Automatisk återkoppling på programmeringsuppgifter

Undersökning och utveckling av hur automatisk återkoppling kan användas för att främja lärande

Hera Dalianis

EXAMENSARBETE INOM TEKNIK OCH LÄRANDE PÅ PROGRAMMET CIVILINGENJÖR OCH LÄRARE

Titel på svenska: Automatisk återkoppling av programmeringsuppgifter.
Titel på engelska: Automatic feedback of programming assignments.
Teknikhandledare: Linda Kann, Kungliga Tekniska Högskolan.
Lärarhandledare: Ida Naimi-Akbar, Kungliga Tekniska Högskolan.
Examinator: Helena Lennholm, Kungliga Tekniska Högskolan.
Examinationsdatum: 220808
TRITA-nummer: TRITA-ITM-EX 2022:450

Sammanfattning

I denna studie undersöktes värdet av automatisk bedömning av programmeringsuppgifter inom högre utbildning ur studenternas perspektiv. Studien utgick från tidigare litteratur och studier om lärande av programmering för att föreslå en utvecklad pedagogik i användningen av automatbedömning. För att konkretisera automatiserad bedömning så undersökte studien hur det automatiserade bedömningsverktyget Kattis används inom ramen för en datalogikurs på Kungliga Tekniska Högskolan. För att förstå studenternas upplevelse av Kattis genomfördes kvalitativa intervjuer med studenter som läst eller läser datalogikursen. Därefter genomfördes en tematisk analys av intervjuerna för att identifiera de centrala delarna i hur studenter använder och upplever Kattis. Dessa delar analyserades sedan utifrån tidigare studier för identifiera vad som bör ändras för att förbättra verktyget ur ett pedagogiskt perspektiv. Däribland identifierades behovet av att informera studenterna och lärare om användning av Kattis och avläsning av Kattis återkoppling.
Utifrån denna analys samt diskussion med lärare som använder Kattis i sin undervisning utformades en arbetsguide med metoder och information för lärare och studenter.

Nyckelord: automatiserad återkoppling, automatiserad bedömning, programmering

Abstract

This study examines the value of automated assessment of programming assignments in higher education from a student perspective. The study used earlier literature and studies about learning programming and pedagogy to suggest a developed pedagogy in the use of automated assessment. To concretize automated assessment the study explored how the automated assessment tool Kattis have been used in a course in computer science at KTH, Royal Institute of Technology. To understand students' experience of Kattis, qualitative interviews was conducted with students who are taking or have taken the course in computer science. The interviews were analyzed using thematic analysis to identify central aspects of how students use and experience Kattis. These aspects were then analyzed based on earlier scientific studies to identify what should be changed in the use of the tool from a pedagogical perspective. Among this there is a need to inform students and teachers about the use of Kattis and how to read the feedback from Kattis. Based on this analysis and discussion with teachers who use Kattis in their courses, a work guide for teachers was designed with methods and information for teachers and students.

Keywords: automated assessment, automated feedback, programming.

Förord

Efter fem års studier i tre länder och på fyra universitet så avslutar jag nu mina studier med detta examensarbete. Det har varit kämpigt men framförallt otroligt roligt och jag vill tacka alla som gjort min examen möjlig - vänner, lärare och familj. Speciellt tack till mina handledare Ida Naimi-Akbar som med fantastisk vägledning, detaljrika kommenterar och idéer möjliggjorde detta examensarbete och Linda Kann som utöver att hoppa in som handledare i sista sekund dessutom gav mig idén till mitt examensarbete och ett otroligt stöd i det tekniska, Linda har dessutom varit en inspirerande mentor under hela min studietid på KTH. Tack till Helena Lennholm som varit min examinator men också en fantastisk lärare och mentor under min tid på KTH.

Tack för de tröstande orden från min pappa Hercules Dalianis som när jag brutit ihop totalt över min kuggade algebratenta berättade att han minsann tappade håret på sin. Det gav insikt i att KTH kan tära mycket på en, men så länge man inte tappar håret så är det bara att ställa sig upp och försöka igen, och om man tappar håret, borsta av det och försök igen. Tack till min mamma Kerstin Johansson som med sitt ideliga stöd genom min studier varit en ständig uppmuntran att kämpa på, även när det är tufft, och när det är som tuffast att ta en termin utomlands. Tack till min syster Freja Dalianis som gjorde mig sällskap på min utbytesstudier i Texas och gjorde den tiden till ett minne för livet. Tack till Sara Lingevall, min bästa vän och studiekamrat som tappert hjälpt mig genom labbar, mattetentor och detta examensarbete.

Innehåll

1 Introduktion 10

1.1 Syfte och Frågeställning 11

2 Bakgrund 11

2.1 Kunskap och bedömning 11

2.2 Formativ bedömning 12

2.3 Lärandet av programmering 12

2.4 Sociokulturellt perspektiv av programmeringsundervisning 13

2.5 Bedömning av programmeringskunskap 13

2.6 Automatisk återkoppling på programmeringsuppgifter 14

2.7 Kontextbeskrivning 15

3 Metod 18

3.1 Datainsamlingsmetod – Intervjuer 19

3.2 Metod för utveckling 21

4 Resultat 21

4.1 Empiriska resultat 21

4.2 Resultat från metodutveckling 24

5 Diskussion 24

5.1 Behovet av utvecklade arbetsmetoder/pedagogisk praktik 25

5.2 Kattis och behaviorismen 25

5.3 Metoddiskussion 26

5.4 Lärandet av programmering och automatisk återkoppling 27

5.5 Begräsningar 29

6 Slutsats 29

7 Vidare forskning 30

Referenser 31

8 Bilagor 33

8.1 Bilaga 1 33

8.2 Bilaga 2 35

8.3 Bilaga 3 38

1 Introduktion

Det fjärde globala målet – *God utbildning* syftar till att alla ska ha tillgång till utbildning, och att detta är essentiellt för att välstånd och jämställdhet (*Globala målen – För hållbar utveckling*, 2022). Edwards (2022) skriver i sin nyligen utgivna artikel *Automated Feedback, the Next Generation: Designing Learning Experiences* om den automatiska återkopplingens utveckling, potential och utmaningar. Enligt Edwards antas automatiserad återkoppling göra det möjligt att minimera arbetsbördan för lärare. Detta möjliggör för lärare att undervisa större klasser och för fler barn, unga och vuxna att få tillgång till utbildning. Automatiserad bedömning kan även användas fristående från en skola, och göra det möjligt att utbilda sig utan tillgång till en formell utbildningsmiljö.

Edwards betonar att den framtida utvecklingen av automatiserad återkoppling bör ta avstamp i lärandeperspektivet och studenternas egna upplevelser av verktygets funktionalitet. Bearman et al. (2020) identifierar att det finns ett gap mellan den pedagogiska och den digitala utvecklingen. Pedagogiken utvecklats inte i samma takt som digitaliseringen och utnyttjar inte den fulla potential som digitaliseringen ger.

Vidare menar Paiva et al. (2022) att det finns ett behov av att utföra empiriska studier som undersöker värdet av automatiserad bedömning i programmeringsuppgifter för lärande och undervisning. Eftersom kravet på fler erfarna och duktiga programmerare ökar drastiskt för varje år, och eftersom digital undervisning tar allt större plats på universiteten finns det både potential och behov men även ett värde i att undersöka och utveckla automatiserad återkoppling av programmeringsuppgifter inom högre utbildning. Utifrån behovet att kunna ge automatiserad återkoppling har flera bedömningsverktyg utvecklats, bland annat det svenska verktyget Kattis som används på de stora tekniska universiteten i Sverige men även internationellt. (*Kattis,* 2022)

Kattis har under de senaste 15 åren använts i undervisningen på KTH som ett verktyg för automatiserad bedömning av programmeringsuppgifter. Under min studietid har jag funnit en varierande inställning - både glädje och frustration - hos de studenter som använt detta verktyg i sin utbildning. Det är därför intressant att undersöka vad som i utformningen eller användningen av Kattis som orsakar denna varierande bild av verktyget, men även hur det kan utvecklas för att främja lärandet hos studenterna.

Sammantaget finns ett behov av att göra en undersökande studie av studenters upplevelse av automatiserad återkoppling, samt utreda hur material för lärare och studenter att tillgå vid användning av automatisk återkoppling bör vara utformat.

1.1 Syfte och Frågeställning

Syftet med denna studie är att, utifrån studenternas upplevelse, undersöka hur automatisk återkoppling kan utvecklas för att stödja studenternas lärande och att utforma förslag till arbetsmetoder till lärare som använder verktyget Kattis för automatisk bedömning.

De frågeställningar som kommer behandlas i denna rapport är som följer.

- Hur upplever studenter att den automatiska återkopplingen begränsar eller påverkar deras lärande?
- Hur kan arbetsmaterial för lärare som använder automatisk återkoppling av programmeringsuppgifter i undervisningen utformas?

2 Bakgrund

I bakgrunden kommer en överblick på tidigare forskning i ämnet kunskap i den digitala eran och automatisk återkoppling presenteras. Detta är relevant för denna studie då den digitala eran ändrat definitionen av kunskap men även hur återkoppling och bedömning görs inom utbildningar. Även tidigare studier på lärandet av programmering kommer presenteras då sådant lärande skiljer sig mycket från inhämtning av traditionell teoretisk kunskap.

2.1 Kunskap och bedömning

Burkle och Cobo (2018) skriver om hur digitaliseringen har ändrat hur vi ser på, hur vi producerar och hur vi förstår kunskap, framförallt inom högre utbildning. Enligt dem är kunnande idag är inte längre begränsat till vilken kunskap vi har att tillgå, utan även hur vi kan använda den. När kunskap och utbildning blivit mer jämställda och tillgängliga för en större massa har även uppstått krav på arbetsmarknaden på högutbildad personal. Burkle och Cobo (2018) betonar att det finns ett behov inom högre utbildning att utveckla det digitala lärandet för att bli mer tillgängligt.
Detta kan uppnås genom att förmedla kunskap genom flera sorters medier, genom implementation av nya digitala lärandeformer och en sammankoppling mellan akademin och industri samt att göra lärande mer globaliserat.

Även Bearman et al. (2020) menar att människors arbetsliv och livsstil har ändrats drastiskt på grund av den digitala utvecklingen, och att bedömning och utbildning inte har utvecklats i samma takt. Enligt Bearman et al. så är bedömning med hjälp av digitala verktyg på universitetsnivå ofta begränsad till tentor, quiz eller essäer som är lätta att betygsätta. Denna användning av digitaliseringen är en begränsning, och bedömning med hjälp av digitala verktyg har betydligt mycket mer potential. Enligt dem bör reformen av bedömning på universitetsnivå ta den moderna digitala världen i beaktande. Världen är i ständig förändring. Bearman et al. betonar att en stor del av

lärande tidigare var förmågan att återskapa och återberätta fenomen. Denna förmåga är idag inte lika relevant, då kunskap blivit mer lättillgänglig på grund av internet. Det är nu andra förmågor som står i centrum, såsom innovation, problemlösning och förmåga att arbeta i grupp. Bearman et al. anser att det finns ett större värde i att ge studenterna möjlighet att själva reflektera över sitt arbete och göra en bedömning själva.

2.2 Formativ bedömning

Enligt Taras (2005) innebär formativ återkoppling att det finns ett gap mellan den nivå som arbetet befinner sig på och den nivå som förväntas på arbetet. Detta skiljer sig från summativ bedömning som är begränsad till en viss poäng eller betygssättning. Taras betonar att formativ bedömning kan användas för två olika metoder, dels för klassrumspraktik där syftet är att lära ut, dels för bedömningspraktik där lärandemålet är integrerat med prestationsmålet. I denna studie avses med begreppet formativ bedömning den senare metoden.

Enligt Gikandi et al. (2011) finns det ett värde i att kombinera klassrumspraktiken och bedömningspraktiken av formativ bedömning vid digital undervisning. Det kan i praktiken innebära att erbjuda asynkron undervisning och återkoppling med ett klassrumskomplement där återkoppling i person ges. De menar att denna typ av arbete kan göra studenter mer engagerade men även hjälpa lärare att notera var det finns möjlighet och utrymme för vidareutveckling. Gikandi et al. identifierar sju principer som är nödvändiga för effektiv digital formativ återkoppling (författarens översättning).

1. *”Hjälper till att tydliggöra vad en bra prestation innebär (mål, kriterier och förväntad standard);*
2. *Underlättar utvecklingen av självbedömning (reflektion) i lärande;*
3. *Levererar högkvalitativ information till studenter om deras lärande;*
4. *Uppmuntrar lärare och studenter till dialog kring lärande.*
5. *Uppmuntrar till och ökar självkänsla;*
6. *Ger möjligheter att minska klyftan mellan nuvarande och önskad prestation;*
7. *Ger information till lärare som kan användas för att forma undervisningen.”* Gikandi et al. (2011)

2.3 Lärandet av programmering

Enligt von Hausswolff (2021) finns det en hög barriär för de flesta studenter när de ska lära sig programmera för första gången och det är ojämnt mellan studenternas prestation i programmering, det vill säga att det ofta finns en grupp av högpresterande studenter och en grupp av lågpresterande studenter i programmeringskurser. Von Hausswolff finner även att det ofta förekommer en defensiv kommunikation i klassrummet som främjar tävling snarare än samarbete

och att den lågpresterande studentgruppen ofta är osäkra på sin egen förmåga och i stort behov av lärarstöd.
Ett sätt att tackla detta hinder kan enligt von Hausswolff vara att låta studenterna med stöd av lärare arbeta i parprogrammering med programmeringsuppgifter.

Enligt von Hausswolff finns ofta en attityd kring programmering från både lärare och studenter att *”antingen har man det eller inte”* och Eckerdal et al. (2005) identifierar förmågan att förstå koncept, metoder och studietekniker för att lära sig programmering som att ha *”programmeringstänk”*, något som behöver utvecklas hos studenter för att lära sig programmera. Enligt von Hausswolff bör studenter arbeta praktiskt med programmering och datalogi för att lära sig. Genom att arbeta in vanor och repetitiva beteenden i en programmeringsmiljö som ger respons kan studenten att få bättre förståelse. Hon betonar att det finns ett värde i att låta studenter testa teser och idéer som ännu inte är färdigformulerade för att få dem att reflektera och agera, vilket även ett praktiskt arbetssätt ger utrymme för. Hon menar även att det bör - utöver standardiserad återkoppling - finnas personlig och individuell återkoppling från lärare direkt till studenternas kod. Eckerdal et al. kommer fram till att kanoniska procedurer är nödvändiga för att studenterna ska få *”programmeringstänk”*. En kanonisk procedur innebär att de problem som studenter möter på i programmeringsuppgift triggar igång en av två lösningsprocedurer hos studenten (Eckerdal et al. 2005). Antingen är lösningen för problemet självklar eller så krävs det en flerstegsprocess för att lösa problemet som studenten kan applicera från tidigare erfarenhet. Utöver att använda sig av kanoniska procedurer behöver studenter vid lärandet av objektorienterad programmering analysera innehållet och egenskaperna hos koncept för att verkligen förstå dem.

2.4 Sociokulturellt perspektiv av programmeringsundervisning

Enligt Säljö (2017) som redogör Vygotskij teorier om sociokulturellt lärande, bör lärandet ske genom sociala och kulturella interaktioner. Vygotskij lägger stor vikt vid att lärandet inte avgörs av biologin utan även människan förmåga att använda f*ysiska verktyg* för att *mediera* handlingar. Vygotskij menar att vi blir tänkande varelser genom att interagera och delta i kommunikation, detta kallas för *appropriering*. En viktig aspekt av appropriering är att det sker genom att människan delar med sig av kunskap till varandra. Detta kan reflekteras i undervisning av programmering inom högre studier, där undervisningen ofta sker via parprogrammering. Dessutom låter användningen av en dator för programmering ge utrymme för studenterna att mediera sina handlingar genom ett fysiskt verktyg.

2.5 Bedömning av programmeringskunskap

Enligt Bearman et al. (2020) så låg fokus i den tidiga utvecklingen av automatisk återkoppling främst på automatisering, på att bearbeta studenternas arbete och

minimera tiden som lärare behöver avsätta på bedömning. Verktygen utvecklades ofta i syfte att bedöma studenters förmåga att återupprepa fakta. I dag har fokus skiftat till lärandet och hur återkoppling kan utvecklas för att påverka studentens lärande enligt Bearman et al. Edwards (2022) menar att det finns utrymme och potential för att utveckla verktyg för automatiserad återkoppling, där verktyget uppmuntrar till utveckling, registrerar framsteg och engagemang. Edwards betonar att genom studentperspektivet kan den automatiserade återkopplingen utvecklas till ett verktyg som främjar lärandet. Riese och Bälter (2022) finner att programmeringslärare tenderar att vilja använda sig av formativ bedömning för att främja lärandet, men ofta känner sig tvungna att även implementera summativ bedömning. Uppgifterna bedöms därmed både summativa och formativa samtidigt. Riese och Bälter finner att det finns ett värde för lärare att reflektera över och integrera formativ och summativ bedömning i programmeringsuppgifterna. De betonar även vikten av att kommunicera syftet med bedömningen till både studenterna och övningsassistenterna.

2.6 Automatisk återkoppling på programmeringsuppgifter

Pears et al. (2007) finner frågan om lärandet av programmering så pass komplex att det är svårt att hitta en specifik metod för att lära ut programmering, och att olika sammanhang kräver olika metoder. Bland de metoder som Pears et al. (2007) behandlar i sin studie är automatisk bedömning av programmeringsuppgifter. Automatisk bedömning av programmeringsuppgifter har blivit nödvändig på grund av den kraftiga ökningen av studenter som läser programmering inom högre utbildning.

Enligt Wasik et al. (2019) använder automatisk bedömning av programmeringsuppgifter sig ofta av en så kallad onlinedomare. Sammantaget är en onlinedomare ett program som med en labbuppgift med givna exempel på förväntad in - och utdata, kan rätta ett lösningsprogram på uppgiften. Detta sker genom att onlinedomaren kompilerar lösningsprogrammet och skickar in indata och jämför utdatan med ett redan existerande facit, och ger därefter återkoppling. Nedan följer den formella definitionen av en sådan onlinedomare hämtad från Wasik et al., 2019 (författarens översättning).

”Den formella definition av en onlinedomare är ett online system som utför vilket som helst av följande steg i en evalueringsprocess.

(1) Samlar och kompilerar program om det behövs och verifierar körbarheten.
(2) Bedömer en lösning $b(d_i, p_i')$ baserat en mängd av testdata, $\boldsymbol{T}$, som finns definierat för ett problem $\boldsymbol{\Pi}$ i en pålitlig, homogen utvärderingsmiljö.
(3) Beräknar aggregerad status $\boldsymbol{s}$ och en utvärderingspoäng $\boldsymbol{v}$ baserat på status och poäng för särskilda fall (dvs $\boldsymbol{s_i}$ och $\boldsymbol{v_i}$ där $1 \leq i \leq |\boldsymbol{T}|$).”

För att förenkla denna definition, så syftar (1) till att online domaren tar in ett program, kompilerar det och avgör om programmet kan köras. (2) syftar att $b(d_i, p_i´)$ -> $o_i´$ är en lösning, där d_i är input $o_i´$ är den output som beräknas givet parametrarna $p_i´$ givna av online domaren. (3) Innebär att online domaren ger en bedömning för varje testfall, om det är godkänt eller varför testfallet inte godkändes.

Enström et al. (2011) studie på införandet av ett verktyg för automatisk återkoppling på programmeringsuppgifter på ett tekniskt universitet under tidigt 2000-tal visar på att studenter initialt blev underkända vid användning av verktyget i större utsträckning, men att det stabiliserade sig efter ett par år. Enström et al. noterade att studenternas attityd inför automatisk återkoppling ändrades över de närmsta åren där cirka hälften initialt var nöjda med verktyget, och fem år senare ökade den siffran till närmare 80 %.
Enström et al. finner att användningen av automatisk återkoppling har skiftat rollen för lärare från att vara domare till att vara ett stöd för studenterna. Enströms studie visar på att studenterna i regel har en mer accepterande attityd till att det automatiska verktyget är strängt än sina lärare och att verktyget minskar arbetsbördan för lärare som kan lägga mer arbetstid på muntlig återkoppling, samt gör studenterna mer självständiga i sitt lärande. Enström et al. (2011) betonar däremot att utformningen av problem i verktyget är tidskrävande för lärare, som bland annat kräver en mängd testdata.
Enström et al. (2010) skriver om att verktyg för automatisk bedömning inom ramen av testdriven utbildning möjliggör för en mer objektiv och korrekt bedömning än manuell bedömning. Enström et al. (2010) menar även att det finns mindre press hos studenterna när de redovisar sitt program, när det redan blivit godkänt i det automatiska bedömningsverktyget.

2.7 Kontextbeskrivning

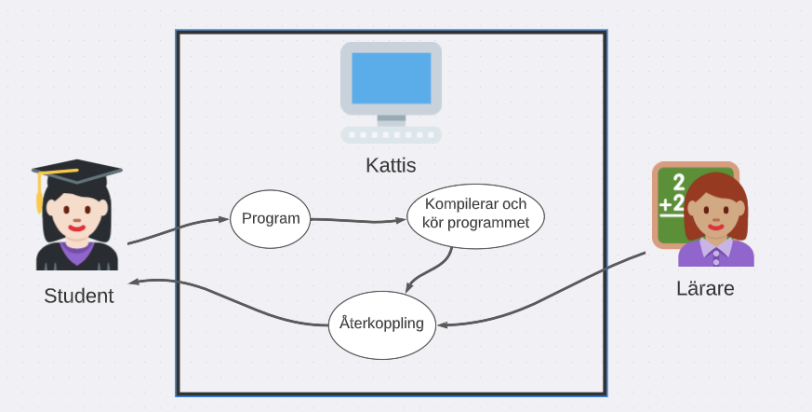
2.7.1 Undervisningsteknologi och undervisningsmaskiner

I sin bok *Undervisningsteknologi* (1968) diskuterar Burrhus Frederic Skinner begreppet *undervisningsmaskiner* ur ett behavioristiskt perspektiv. Begreppet *undervisningsmaskiner* syftar på mekaniska eller elektriska maskiner som ställer frågor till eleven. Om frågan besvaras rätt får eleven gå vidare till nästa fråga. Att frågan besvarats korrekt kan visas med betingad förstärkning, till exempel genom en klocka. Om eleven besvarar frågan fel så kommer inte nästa fråga upp, och ska alltså ge negativ förstärkning. Till denna undervisningsmaskin kan ett räkneverk tillsättas för att kommunicera hur många rätt respektive fel eleven fått. Ytterligare en viktig egenskap hos undervisningsmaskinen är att den ska kunna användas självständigt av eleven. På grund av detta har undervisningsmaskinen har även ett värde ur en evalueringssynpunkt. Lärare kan med hjälp av undervisningsmaskinen minimera den arbetstid som behövs läggas på rättning. Skinner (1968) föreslår även en

variation på en undervisningsmaskin som där en stegsekvens, möjliggör för studenten att ta mindre steget för att slutligen nå ett visst mål. Idag har undervisningsmaskinen utvecklats till digital automatisk återkoppling, och även om den automatiska återkoppling är komplex så finns det fortfarande aspekter hos automatisk återkoppling som tar avstamp i Skinners undervisningsmaskin.

2.7.2 Kattis - en onlinedomare

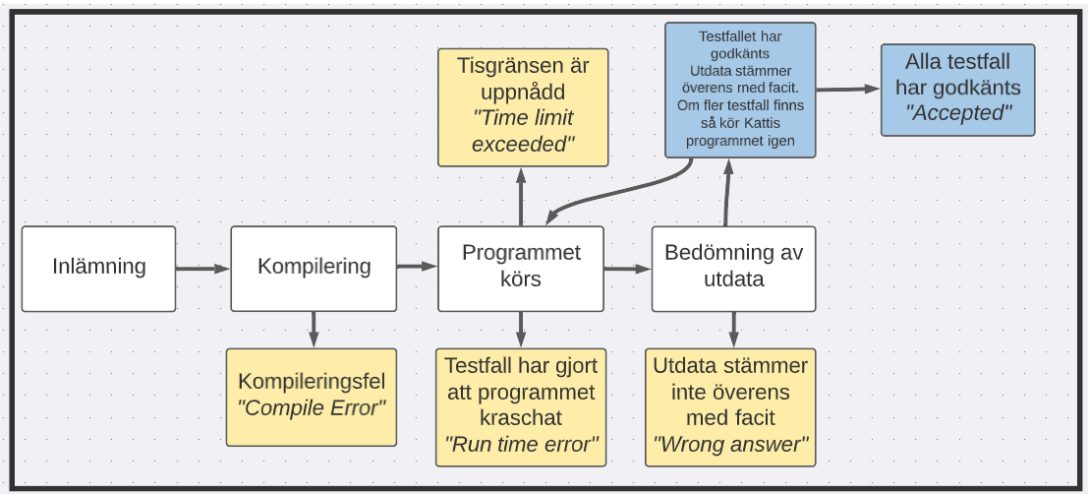
Kattis är ett webbaserat rättningsprogram för programmeringsuppgifter. På Kattis kan lärare lägga upp eller använda redan existerande problem i sin undervisning. Problemen är formulerade så att de beskriver förväntad indata respektive utdata för problemet. När studenter gjort uppgiften så skickar de in programmet i Kattis som kompilerar och kör programmet och därefter ger återkoppling. Denna återkoppling kan senare kompletteras av en lärare. Kattis har även en plagiatkontroll som gör det möjligt för lärare att göra plagiatkontroller mot programmeringskoderna som studenterna skriver. (Se figur 1)



FIGUR 1

Kattis använder sig av så kallad *”black box testing”*. Enligt Zhou et al. (2018) innebär detta att när en student lämnat in sin lösning för en programmeringsuppgift så kompilerar Kattis programmet och skickar in ett testfall som indata, därefter jämför Kattis utdata som programmet ger för detta testfall med ett redan existerande svar, ett facit. Om testfallet är godkänt så testar Kattis nästa testfall.
Varje testfall som körs representeras av en ruta för användaren, om programmet passerat testet så dyker en grön bock upp i rutan. Problemen är formulerade så att de beskriver vad programmet förväntas kunna göra, samt en tidsgräns. De första testfallen samt förväntat svar finns givna i problemet, medans de sista testfallen är dolda.
Ett program som ger rätt utdata för samtliga testfall betraktas som godkänt eller *”Accepted”,* då kommer alltså samtliga av rutorna fyllas av gröna bockar. Om något av testfallen skulle ge ut utdata som inte stämmer överens med Kattis facit så returneras återkopplingen fel svar eller *”Wrong answer”* och en röd bock dyker upp

i rutan för det testfall som inte godkänts. Då slutar Kattis testa nya testfall. Likaså om den tidsgräns som satts för uppgiften av läraren uppnåtts så avslutas testning och felmeddelandet *”Time limit exceeded”* returneras. Om ett testfall orsakar att programmet kraschar så ger Kattis ut felmeddelandet *”Run time error”*. Det kan också vara så att Kattis inte klarar av att kompilera programmet, då ges felmeddelandet *”Compile error”* som återkoppling. Kattis har även fler felmeddelanden men beskrivningen ovan har begränsats till de vanligaste, där den absolut vanligaste är *”Wrong answer”* (Se figur 2). (Enström et al., 2011)



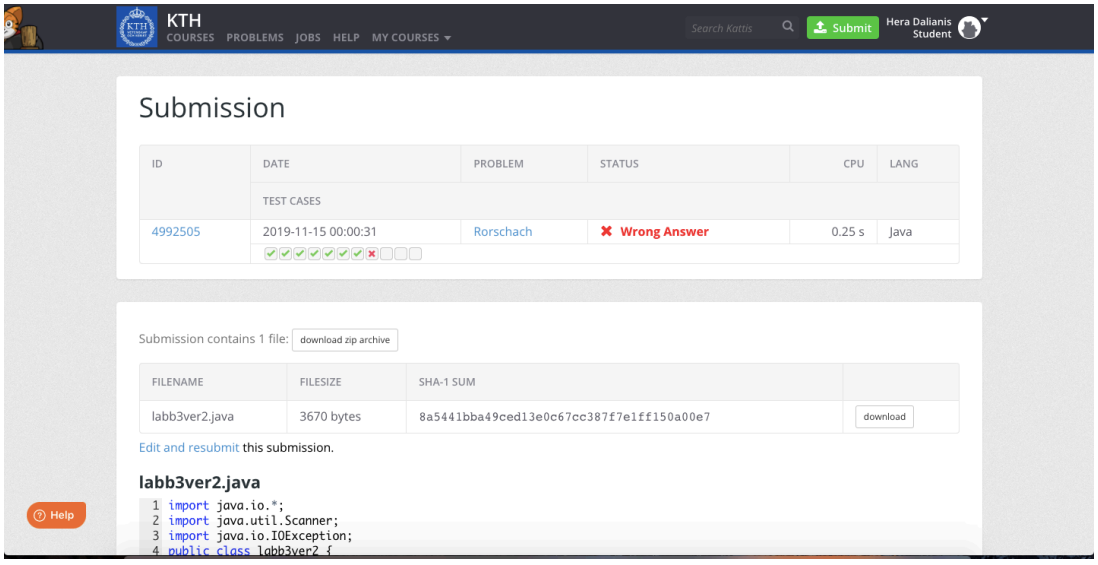
FIGUR 2 – FELMEDDELANDE I KATTIS

2.7.3 Tekniken bakom Kattis

Kattis backend är skrivet i Python. Personuppgifter, inlämningar, bedömningar och data om uppgifterna är lagrade i en PostreSQL databas. Eftersom Kattis kompilerar och kör de program så skickas in så kan Kattis vara väldigt sårbart för exempelvis virus, säkerhetssystemen som skyddar Kattis och de personuppgifter som Kattis lagrar mot attacker är skrivna främst i C och Java.
När en inlämning görs lagras denna i databasen och backend informeras om att en bedömning ska göras. Backend hämtar sedan inlämningen från databasen och lagrar koden i en tillfällig katalog. Efter detta kompilerar backend koden med hjälp av kompileringsflaggor som är specificerade för uppgiften.

2.7.4 Kattis som undervisningsmaskin

Även om Skinner inte kunde visualisera just en digital undervisningsmaskin, överensstämmer Kattis och undervisningsmaskinen i många aspekter. Kattis är ett digitalt verktyg för lärande och fungerar med positiv förstärkning respektive negativ förstärkning. Kattis ger studenter en grön bock i respektive ruta för det testfall som godkänts eller ett rött kryss för det testfall som inte blivit godkänd. Här blir studenternas associationer till respektive symbol och färg viktigt för den förstärkning som sker hos studenten. (Se figur 3)



FIGUR 3 – INLÄMNING I KATTIS

Detta innebär i praktiken att trots att ett program inte blivit godkänd på sista testfallet så ges studenterna positiv förstärkning för de testfall som programmet faktiskt har klarat. I den benämningen bör detta och den negativa förstärkning som sker när studenterna läser av den röda bocken för ett underkänt testfall ge ett incitament för studenterna att arbeta vidare på sin kod och skicka in den igen. Precis som den reviderade versionen som Skinner föreslog så har Kattis flera mindre steg, där varje steg leder till en färdig produkt. I dessa mindre steg innefattas att när studenterna blivit underkända på ett okänt testfall så ska de skriva egna testfall för att se varför deras kod inte blivit godkänd. När studenten har klarat alla steg och gjort ett fungerande program är tanken att studenten har lärt sig det området som labbuppgiften ska täcka in.

3 Metod

I denna studie undersöktes hur högskolestudenter upplever automatisk återkoppling som stöd i att lära sig programmering. Specifikt undersöks hur studenter upplever och använder verktyget Kattis inom ramen för kurser i datalogi på ett tekniskt lärosäte i Sverige. Det empiriska materialet är genererat genom kvalitativa intervjuer och en induktiv tematisk analys tillämpades.

Resultatet av den empiriska studien användes för att vidare utforma material för att utveckla bedömnings och undervisningspraktiken på lärosätet. Utvecklingen av lärarguide och studentguide skedde genom inläsning på automatisk återkoppling och lärandet samt i diskussion med lärare inom datalogi på lärosätet.

3.1 Datainsamlingsmetod – Intervjuer

3.1.1 Urval av respondenter

Till respondenterna valdes åtta studenter som läst eller läser samma kurs i datalogi och algoritmer. Samtliga studenter läser ett civilingenjörs-program på KTH. För flertalet av studenterna är det den första kursen de läser som använder Kattis i sin bedömning i några av labbarna. Studenterna läser första respektive andra året på sin utbildning. Respondenterna är både män och kvinnor i åldern mellan 19–29 år.
För att komma i kontakt med respondenter så lades ett anslag ut på den digitala plattformen i kursrummet för kursen. Dessutom kontaktades studenter som redan läst kursen, cirka tre månader innan via en Facebook-grupp för ett civilingenjörsprogram på KTH. Dessutom tillfrågades studenter att delta i intervjuer i samband med en digital föreläsning i pågående kursomgång. Se tabell 1 för en summering av respondenterna.

TABELL 1

Respondent	Ålder	Kön	Program
A	21	Man	Elektroteknik
B	22	Kvinna	Civilingenjör och lärare
C	25	Man	Teknisk matematik
D	22	Man	Civilingenjör och lärare
E	20	Kvinna	Civilingenjör och lärare
F	29	Man	Civilingenjör och lärare
G	23	Man	Elektroteknik
H	21	Kvinna	Industriell ekonomi

3.1.2 Genomförande av intervjuer

En kombination av semistrukturerade kvalitativa individuella och gruppintervjuer genomfördes. Enligt Kvale och Brinkmann (2009) ger en kvalitativ intervjustudie en helhetsbild av respondenternas upplevelse i motsats till en kvantitativ intervjustudie vilka kan begränsa data samlas in till skalor eller kortare svar. Syftet till

gruppintervjuerna är att väcka diskussion och samtal mellan respondenterna. Några av intervjuerna genomfördes individuellt på grund av schemakrockar.

För varje intervju avsattes 30 minuter. Samtliga intervjuer startades med en orientering där intervjuaren tydliggjorde syftet med intervjun, låtit respondenterna läsa igenom en samtyckesblankett för intervju och datahantering och avsatt tid för eventuella frågor. Intervjuerna genomfördes i ett privat rum. Intervjun avslutades med tid för reflektion där intervjuobjekten får möjlighet att reflektera med några avslutande tankar och intervjuaren summerar det den noterat under intervjun så att intervjuobjekten kan bekräfta att det stämmer eller inte. Hela intervjun genomfördes med hjälp av en intervjuguide, se bilaga 1. Se tabell 2 för längd för respektive intervju samt vilka respondenter som närvarade vid respektive intervjutillfälle.

TABELL 2

Intervju	Närvarande respondenter	Längd
1	A B	17 minuter
2	C	8 minuter
3	D E F	26 minuter
4	G	13 minuter
5	H	10 minuter

3.1.3 Analysmetod

Den analysmetod som använts i denna studie är tematisk analys. Enligt Braun och Clarke (2006) är tematisk analys en metod för att organisera, analysera och beskriva de mönster som finns inom en given datamängd på ett så sammanfattat sätt som är möjligt. Metoden syftar till att forskaren i fråga synar sin data för att identifiera mönster och teman och sedan avgöra vilka mönster och teman som är relevanta. Tematisk analys är särskilt användbart som analysmetod för att synliggöra vad datapunkter reflekterar i verkligheten. En central del av tematisk analys är att identifiera teman och mönster i datamängden. Värt att notera är att teman inte behöver vara närvarande i en specifik mängd av data för att vara relevant. Det finns även en vikt att identifiera mindre teman som tillsammans kan fånga ett intressant element i datamängden. (Braun & Clarke, 2006)

3.1.4 Tematisk analys genom sex steg

Den tematiska analysen som genomförts in linje med Braun och Clarke (2006) metoder för tematisk analys.

De första steget var att bekanta sig med den data som samlats in genom intervjuer, detta gjordes genom att transkribera, läsa igenom och föra anteckningar. Nästa steg i den tematiska analysen var kodningen. Det innebär att det som studenterna sade summerades under olika koder. Varje student tilldelades i syfte för anonymisering en bokstav, denna bokstav indikerade även i kodningen vilken student som uttryckte vad. Siffrorna som följde varje bokstav indikerade vilken ordning det sades. Kodningen gjorde det därefter möjligt att organisera teman och underteman, där studenterna hade gemensamma eller liknande upplevelser av Kattis mellan studenterna.
Därefter behandlades och analyserades temata utifrån följande punkter.

1. Kan vissa av dessa teman slås ihop?
2. Kan vissa av teman delas upp?
3. Har temat tillräckligt mycket data för att anses vara ett tema?
4. Kontrollera mot forskningsfrågan.

3.2 Metod för utveckling

Metoden för utveckling av det arbetsmaterial som utformats i samband med denna studie följer designmetoder som presenteras av Wikberg-Nilsson et al. (2021).
Strategin för utveckling av arbetsmaterialet var att förvalta Kattis som lärandeverktyg, det vill säga att då utgå från den färdiga produkten Kattis och utvärdera vilka mindre ändringar kan göras i hur lärare och studenter använder Kattis. De steg som ingick i designprocessen var planera projektet, utforska kontexten, skapa idéer och prototypa koncept. Det första steget i designprocessen var att planera projektet, och identifiera vad som behövs göras i vilket steg av projektet. Där lades planeringen fram som följer, att först utforska kontexten, genom inläsning om ämnet Kattis och lärande av programmering. Denna kontext gav sedan en grund för att göra en gav en kvalitativ informationsinsamling och utforma de intervjuer som utfördes med studenter som använt verktyget Kattis i den specifika datalogikursen.
För att identifiera de behov som fanns hos studenterna användes tematisk analys av datan från intervjuerna. Därefter behövdes idéer kring arbetsmaterialet skapas, detta gjordes genom resultatet från dataanalysen. När prototypen för lärarguiden och studentguiden utvecklats så presenterades denna för en grupp av lärare som använder Kattis i sin undervisning. Denna diskussion möjliggjorde en kompletterad utformning av guiderna. För guiderna se bilaga 2, 3.

4 Resultat

Här presenteras resultat från de kvalitativa intervjuerna med studenterna och diskussionen med lärare om det lärarmaterial för Kattis som utformats.

4.1 Empiriska resultat

Här presenteras de teman som kunde identifieras från den tematiska analysen. Temata beskriver olika aspekter på användningen av Kattis som påverkar studenterna och deras upplevelser av sitt lärande. Det första temat är *Metod för att klara ett underkänt program i Kattis*, detta tema summerar de metoder som studenterna använde för att ändra i ett program som blivit underkänt i Kattis för att bli godkänt. Det andra temat är *Hur studenter upplever röda kryss respektive gröna bockar*, alltså hur studenterna upplevde att få negativ återkoppling av Kattis och positiv återkoppling av Kattis. Temat därefter är *Negativa aspekter med Kattis – vad studenter inte uppskattar med Kattis*, som summerar de samtliga negativa upplevelser som studenterna haft med verktyget. Det sista temat är *Positiva aspekter med Kattis – vad studenter uppskattar med Kattis,* som summerar det som studenterna gillar med verktyget.

1. Metod för att klara ett underkänt program i Kattis

Studenterna hade flera olika metoder för att ändra i sitt program så att det skulle bli godkänt i Kattis. Om det var ett känt fall som blivit underkänt så testade studenterna testfallet i sin egen kompilator. Om det var okänt eller något annat fel då kunde de studenter som hade kunskap om felmeddelande använda Kattis för felsökning, och testa programmet med egna testfall i sin egen kompilator. De studenter som saknade förståelse för felmeddelande, försökte också testa programmet, men upplevde att det var betydligt svårare att lösa problem. Vissa studenter gav upp och gick till redovisningen utan att programmet blivit godkänt i Kattis. De studenter som förstod att de kunde kontakta lärare för hjälp gjorde de om de inte lyckades lösa problemet själva.

” Det första jag gör är att kolla om jag fått något felmeddelande. Men det får man ju inte oftast. Då går jag tillbaka i koden och undersöker och testar lite egna inputs. För att se om det kan vara något jag missat eller vad den kan reagera på.
Men ofta så ställer jag mig i allmänhandledningen och frågar, för jag har kommit på att de ser vad som felet är. ”

2. Hur studenter upplever röda kryss respektive gröna bockar.

Studenters upplevelse av de röda kryssen, som dyker upp då ett programmet blivit underkänt var i regel väldigt negativ och ilska samt tappade fokus och motivation av den negativa återkopplingen. Men det fanns även studenter som upplevde en nyfikenhet och engagemang av de röda kryssen. Det fanns alltså en bredd i hur studenter upplevde röda kryss. Studenterna upplevde glädje över att få gröna bockar av Kattis. Däremot upplevde en del studenter att de gröna bockarna inte nödvändigtvis indikerade att programmet de skrivit höll en hög nivå. Vissa studenter upplevde att gröna bockarna gav mer positiva känslor än vad de röda kryssen gav negativa känslor.

”Nyfiken. Det har att göra med mig som person, när någonting går fel då är det intressant. När någonting bara funkar då tänker man inte ens på det och går

vidare. Men om något går fel då är det intressant och man ska leta, det bli skattjakt. Debugga och hitta vad som är fel."

"Jag vill sätta eld på datorn – jag vet inte, irritation antar jag. Frustration för att det är dåligt kommunicerat."

" Vi brukar skoja om det och säga att det är bästa känslan, när man sitter där och har kodat några timmar och den går igenom, det är jubelrop. Det är faktiskt väldigt kul att få den feedbacken, den hör gröna upplysningen. Det är bra belöningssystem."

"Gröna är bara jävligt skönt. Koden är inte bra men jag lämnar in ändå. Skönt att bli klar med saker, helt enkelt."

3. Negativa aspekter med Kattis – vad studenter inte uppskattar med Kattis

När studenterna tillfrågades vad de inte gillade med Kattis var de ofta avsaknaden av återkoppling från Kattis. De betonade bland annat att en vanlig kompilator ger betydligt mer återkoppling om vad och var som kan vara fel i programmet och att Kattis hade väldigt få felmeddelande i hänsyn till hur många saker som faktiskt kan vara fel med programmet. De upplevde att Kattis var onödigt sträng, som gav fel för till exempel stavfel. De tyckte även att Kattis beskrivning av felmeddelande var otydliga.

"En sak som jag tycker kan vara problematisk med Kattis är att när du får ett rött kryss, får du så pass lite information att om du inte tar kontakt med en assistent eller är väldigt kreativ i din felsökning så är du ganska strandad i den felkoden. För du får ingen felkod, du får bara vad Kattis kallar för en felkod, men den säger ingenting om vad som har gått snett i koden."

"Jag googlar Kattis och felmeddelande och så kommer man till någon felsökningssida där felen står beskrivna. Dock är de inte alltid jättebra beskrivna."

4. Positiva aspekter med Kattis – vad studenter uppskattar med Kattis

Det som studenter upplevde var bra med Kattis var bland annat den trygghet som medfördes av att blivit godkänd innan en redovisning, även att lärare kunde lägga mer tid på vägledning av teorin än att testa programmet i praktiken. De tyckte även om den direkta återkopplingen som Kattis gav, då hjälp från lärare kan ta några dagar att få. De uppskattade även möjligheten att kunna lämna in problem när som helst och att man kunde gå tillbaka och lösa gamla problem igen. De upplevde att Kattis är väldigt användarvänligt. De tyckte dessutom att det var lugnande att ha blivit godkända på uppgiften innan redovisningen.

”Jag tycker det är bra att responsen är direkt, man behöver inte sitta och vänta i timmar eller en dag på att få svar. Du har resultat direkt och kan ta ställning och gå tillbaka liksom.”

”För ibland sitter man klockan tre på natten, för det är nu man tycker att man har tid att programmera.”

4.2 Resultat från metodutveckling

Diskussionen med lärare utmynnade i behovet som lärare har av information om hur Kattis fungerar, hur de använder Kattis samt hur bedömning och återkoppling sker. Även hur Kattis skiljer sig mellan lärare och studenter, det vill säga vilken information som lärare kan ta del av och inte studenterna. Utöver behovet för just lärare att ta del av lärarguiden fann även lärarna ett värde i att guiden bör utformas på ett sånt sätt att den kan även delas ut till lärarassistenterna som ofta behöver vända sig till kursansvarig med sina frågor. De ansåg även att det kunde vara bra med information om hur man som lärare kan använda sig av redan existerande problem och hur man skapar ett nytt problem i Kattis, där denna information främst är riktad till läraren på kursen. Lärarna ansåg även värdet av en studentguide, som ger studenterna information om hur man Kattis fungerar, hur man kommer åt kursens labblydelser, hur man registrerar sig för kursen och hur man lämnar in en eller flera filer. Lärare och kursansvariga såg gärna att båda guiderna ska fungerar fristående från exempelvis en föreläsning och att studenter och lärarassistenter ska använda guiderna för självständig inlärning. Från metodutvecklingen kunde slutsatsen dras att både lärare och studenterna saknade kunskap om hur Kattis testar ett program och vad respektive felmeddelande beror på. Därför innefattar båda guiderna information om hur testning sker och ett flödesschema för att förklara när respektive felmeddelande inträffar. Dessutom innefattar studentguiden lösningsförslag för respektive felmeddelande. För lärarguide, se bilaga 2 och för studentguide se bilaga 3.

5 Diskussion

Från intervjustudien, inläsning av material och diskussion med lärare synliggjordes behovet av en guide vid användning av Kattis. I denna diskussion kommer frågeställningen *Hur upplever studenter att den automatiska återkopplingen begränsar eller påverkar deras lärande?* diskuteras. Frågeställningen *Hur kan arbetsmaterial för lärare som använder automatisk återkoppling av programmeringsuppgifter i undervisningen utformas*? kommer diskuteras och lyfta det som motiverar innehållet och utformningen av dessa guider. Det som kommer inkluderas i guiderna går att läsa av från de grå rutorna i kapitel 5.4.

5.1 Behovet av utvecklade arbetsmetoder/pedagogisk praktik

Resultatet från den tematiska analysen visade på att studenter saknade kunskap om Kattis, vilket försvårade användningen av verktyget för studenterna. I relation till den första och tredje principen av sju principer som Gikandi et al. (2011) presenterar som värdefulla för formativ återkoppling online, kan slutsatsen att studenterna behöver mer information om hur Kattis fungerar och vad som förväntas av studenterna dras. Studenterna var i regel mer självständiga i sin användning av verktyget vilket innebär att Kattis ger studenterna möjlighet till självreflektion, detta innebär att Kattis uppfyller den andra principen. Den fjärde principen som Gikandi et al. presenterar, lyfter vikten av kommunikation mellan lärare och student, enligt intervjustudien var lärare en central del i användningen av Kattis och genom muntlig presentation på teori upplevde studenterna att de fick hjälp att förstå teorin djupare. Däremot saknade en del studenter vetskapen om möjligheten att kontakta lärare, och att lärare kunde bistå med hjälp givet ett inlämningsid. Deras femte princip handlar om att uppmuntra till lärande och självförtroende, intervjuerna visade på att studenterna kände en glädje och stolthet vid avklarat problem i Kattis, vilket indikerar att denna princip uppnås. Deras sjätte princip handlar om lärandet i sig och Enström et al. (2011) studie på övergången från muntliga redovisningar till Kattis tyder på att studenterna lyckas prestera trots återkopplingen online. Den sista punkten Gikandi et al. (2011) presenterar betonar vikten av att det ska finnas tillgänglig information om hur lärare ska använda sig av automatisk bedömning i sin undervisning.

5.2 Kattis och behaviorismen

Baserat på de kriterierna som Skinner (1968) listar för undervisningsmaskiner – att en undervisningsmaskin ska ge någon form av positiv förstärkning när en uppgift är avklarad och negativ förstärkning när en uppgift inte är avklarad - är Kattis en form av undervisningsmaskin. Negativ förstärkning innebär att träna bort ett beteende, i detta fall träna bort att bli underkänd i Kattis. Studenterna upplevde negativa känslor vid röda kryss vilket tyder på negativ förstärkning. Det fanns så pass mycket negativ förstärkning att en del studenter upplevde att det bar emot att skicka in kod i Kattis, på grund av risken att få ett rött kryss. Det fanns alltså här ett problem med den negativa förstärkningen som skedde hos vissa studenter och att den negativa associationen med Kattis gör det stressande och svårt att använda verktyget. Å andra sidan upplevde vissa studenter att de röda kryssen var motiverande och agerade som en drivkraft att arbeta vidare med sin uppgift. Det fanns även en positiv förstärkning som skedde vid avläsning av gröna bockar hos studenterna. Studenterna uttryckte att de upplevde positiva känslor när ett program blivit godkänt av Kattis. Jakten på denna positiva upplevelse kan ligga som grund att arbeta hårdare för att slutföra sin uppgift. När man väger in behaviorismen i hur Kattis är utformat så tyder mycket på att användningen av Kattis behöver utvecklas, t.ex. hur man använder sig av negativ

förstärkning för att lära sig, den negativa förstärkningen i form av ett rött kryss räcker inte för att studenterna ska utvecklas och lära sig från Kattis.

5.3 Metoddiskussion

5.3.1 Metodutveckling

Designstrategin att förvalta Kattis som lärandeverktyg valdes då det redan fanns befintliga användare och ett verktyg för automatisk bedömning av programmeringsuppgifter. Anledningen till att utvecklingsmetoden valdes och implementerades var att det ansågs som en lämplig metod och process för att förvalta Kattis som lärandeverktyg, det vill säga att utforma arbetsmaterial till lärare och studenter som använder Kattis.

5.3.2 Intervju

Initialt var tanken att använda sig av kvantitativ informationsinsamling för att utforska kontexten av hur studenter använder Kattis. Dock drogs slutsatsen att denna metod för informationsinsamling skulle ge för generella svar och inte skulle ge en tillräckligt nyanserad bild av vilka behov som fanns hos studenterna när de använder Kattis. Dessutom gav kvalitativa intervjuer möjlighet för att presentera förslag till studentguiden för studenter och jag kunde då få direkt feedback på de idéerna för studentguiden. Kvalitativa intervjuer ger dessutom mer flexibilitet till följdfrågor.

5.3.3 Tematisk analys

Tematisk analys valdes för att analysera data då det är en vedertagen metod för att identifiera, analyser och beskriva data från kvalitativa intervjuer. Det gav en bra överblick av den större mängd data som genererades från intervjuerna. Den tematiska analysen gav dessutom tydliga beskrivningar av vad som kan förbättras i studenternas användningen av Kattis.

5.3.4 Validitet och reliabilitet

För att dokumentera spelades intervjuerna in. Dessutom fördes anteckningar förts av intervjuaren under intervjumomentet. Efteråt transkriberades samtliga intervjuer. Genom att simultant banda intervjun och föra anteckningar så försäkras reliabiliteten av intervjun och möjliggör att förstärka det som antecknats med hjälp av inspelningen av intervjun. Validitet uppnås genom val av intervjuobjekt. Då intervjuobjekten själva använt Kattis och syftet är att ta reda på hur studenter upplever Kattis så uppnås hög validitet från intervjuerna. (Kvale & Brinkmann, 2009).

Det är värt att ta i beaktande att urvalet av studenter kan vara begränsad till högpresterande studenter som aktivt är inne på kursrummet samt närvarar vid föreläsningar. Från intervjuerna noterades även de betyg som studenterna siktar på i kursen, men värt att notera är att även om en student siktar på ett betyg på E nivå

så är denna nivå tämligen hög. Inspelning via endast ljud och anteckningar kan även begränsa vad det är studenterna vill förmedla med sina svar, inspelning med bild hade dessutom kunnat klargöra ytterligare vad studenterna vill säga.

5.3.5 Etiska övervägande

Enligt Vetenskapsrådet (2017) bör respondenterna informeras om syftet med studien samt de risker och fördelar som finns förenade med att delta i studien. Dessutom ska respondenterna informeras vem som kan få tillgång till studien. Detta gjordes, genom en samtyckesblankett som respondenterna fick skriva under innan intervjun startar. Respondenterna informerades även om möjligheten att dra sig ur intervju när de ville samt att de fick kontaktuppgifter till huvudansvarig för uppgifterna och intervjuerna om de hade ytterligare frågor. Respondenterna informerades också om syftet med intervju och studien. Respondenterna informerades dessutom om att intervjuerna bandades. För att försäkra respondenternas anonymitet så anonymiserades kursnamnet i denna rapport. Dessutom begränsades personlig information om respondenterna för att inte kunna identifiera vad respektive respondent sagt. Utöver detta informerades respondenterna om hur data kommer hanteras och hur det kommer presenteras i rapporten.

Data från intervjuerna hanterades enligt de riktlinjer som gäller för personuppgifter som regleras i Datahanteringslagen (GDPR). Detta gjordes genom att inga namn anges på filerna för transkriberingarna. Dessutom sparades transkriberingarna på en annan enhet än inspelningarna från intervjuerna. Samtyckesblanketterna med namn sparades fysiskt.

5.4 Lärandet av programmering och automatisk återkoppling

Von Hausswolff (2021) identifierar den höga barriär som finns för förstagångs programmerare. För att komma över denna barriär finns det ett värde i att testa icke färdig formulerade idéer för att få förståelse. Automatisk återkoppling ger utrymme för detta, när studenter initialt skapar enklare lösningar, som sedan utvecklas till mer komplexa lösningar. Dock kan det svårt för en förstagångsprogrammerare att förstå hur Kattis fungerar och därför är det svårt att använda verktyget. Även lärarna noterade detta hos sina studenter. Min analys visade att studenterna inte alltid förstår hur Kattis fungerar eller hur man registrerar sig i Kattis.

Studentguide	Bör innehålla information om hur Kattis fungerar. Däribland hur man registrerar sig på kursen i Kattis.

Användningen av automatisk återkoppling ger även studenter möjlighet att få återkoppling utanför undervisningstid vilket ger lågpresterande studenter möjlighet att lägga ner den tiden som behövs för att komma ikapp gruppen av högpresterande studenter, och alltså stänga det gap mellan hög och lågpresterande studenter som von Hausswolff (2021) identifierar. Däremot är återkopplingen mycket begränsad vilket

gör det svårt att åtgärda de fel som Kattis hittar i programmet, utan handledning från lärare. Dessutom kan möjligheten att arbeta med programmet när som helst på dygnet orsaka stress och psykisk ohälsa hos studenterna. Från intervjustudien upptäcktes att Kattis röda kryss orsakade en så pass så stor stress att vissa studenter inte klarade av att lämna in en uppgift såvida studenten ifråga inte visste att uppgiften skulle bli godkänd sen innan.

Studentguide	Bör innehålla information om hur man tolkar felmeddelande och löser problemen.

Enligt Enström et al. (2010) studie på testdriven utbildning, ger automatisk återkoppling en mer korrekt och objektiv bedömning på kunskap, vilket även studenterna upplevde att Kattis gjorde. Både enligt Enström et al. (2010) och den intervjustudie som genomfördes i denna rapport visar på att studenter upplever mindre press vid redovisningen om deras program redan blivit godkänt av Kattis.
Enligt Enström et al. (2011) har studenter en mer accepterande attityd till användningen av automatiserade verktyg i sin utbildning och har lättare att förstå att verktyget är strängt än att lärarna är stränga. Intervjustudien och diskussion med lärare stärker denna slutsats.
Om man även väger in Pears et al. (2007) så har den ökade mängden av programmeringsstudenter gjort det nästintill omöjligt för lärare att rätta manuellt, genom användning av Kattis underlättas detta arbete och mer fokus kan läggas på handledning för studenterna. Från diskussionen berättade lärarna om användningen av lärarassistenter för att hantera den stora mängd studenter som vill lära sig programmering, och att dessa lärarassistenter ofta uttryckte okunskap om verktyget. Enligt Riese et al. (2022) är det av stor vikt att kommunicera till både lärarassistenter och studenter syftet med bedömningen.

Lärarguide	Bör vara utformad så att den även kan ges till lärarassistenterna, bland annat med information om hur Kattis fungerar. Bör innehålla information om hur man handleder studenterna till en komplett lösning. Muntlig återkoppling är ett viktigt komplement till Kattis återkoppling. Bör motivera syftet med bedömningen.
Studentguide	Bör betona att en inlämning måste vara godkänd i Kattis innan redovisning. Bör motivera syftet med bedömningen.

Enligt Enström et al. (2011) och diskussion med lärare tar det mycket tid att skapa nya problem i Kattis. Därför bör lärarguiden innehålla information om hur man skapar egna problem i Kattis samt hur lärare gör för att använda sig av redan existerande problem i databasen. Enligt Bearman et al. (2021) så har den tidiga användningen av digitala verktyg där fokus låg hos studentens förmåga att

återberätta fakta inte vara tillräcklig för att reflektera den kompetens som behövs på i samtiden på arbetsmarknaden. Burkle och Cobo (2018) finner även de digitaliseringen har ändrat synen på kunskap.
Digitaliseringen ger högre krav på studenternas förmåga att reflektera och analysera sitt eget lärande. I hur Kattis är utformat och används så ges studenter utrymme att utveckla denna kompetens. Enligt Eckerdal et al. (2005) behöver studenter vid lärandet av objektorienterad programmering analysera innehållet och egenskaperna hos koncept för att verkligen förstå dem. Alltså är kompetensen med teoretisk förståelse fortfarande är central i lärandet av programmering, vilket Kattis inte nödvändigtvis möjlighet att utveckla.

Lärarguide	Bör innehålla information om hur man utformar problem och hur man använder redan existerande problem i Kattis. Inkludera en beskrivning på hur problem kan utformas för att testa teoretisk förståelse hos studenterna.

5.5 Begräsningar

Denna studie hade haft flertalet begränsningar. Det hade varit att föredra att göra en mer omfattande intervjustudie med fler respondenter för att få en bredare bild av studenterna upplevelse av verktyget. Den stora variation av upplevelse som fanns hos de åtta studenterna som intervjuades tyder på att det kan skilja sig stort studenter emellan. Studien är dessutom begränsad till hur studenter som läser en viss datalogikurs på Kungliga Tekniska Högskolan upplever och upplever lärandet av programmering via specifikt verktyget Kattis. Då arbetsmaterialet utformades baserat på denna intervjustudie samt i diskussion med lärare som undervisar på KTH så är arbetsmaterialet utformat för just lärare och studenter som läser en programmeringskurs som använder Kattis på KTH.
Ytterligare en begräsning av omfattningen av studien är att de arbetsmetoder och guider som utvecklats inte kunde implementeras, testas och sedan vidareutvecklas.

6 Slutsats

Hur upplever studenter att den automatiska återkopplingen begränsar eller påverkar deras lärande?

Det finns mycket som tyder på att det finns ett värde i att använda automatisk återkoppling inom utbildning på högre nivå. Studenter upplever att de uppskattar flexibiliteten som finns i verktyget och att det ger utrymme till mer lärarkontakt. De upplever även att de lär sig mer och uppskattar den snabba återkoppling.
Vid formativ bedömning så indikerar arbetsmetoden med automatisk bedömning på många perspektiv vara effektiv för lärandet. Men även här finns utrymme för förbättringar. Litteratur och intervjuerna tyder på att studenter hade bristande kunskaper om automatiska verktyg och att detta ofta leder till en form av frustration

och oförståelse hos studenterna. Dessutom kan användning av automatisk verktyg för bedömning av programmeringsuppgifter dessutom bli tidskrävande för lärare. Detta kan åtgärdas genom att presentera riktlinjer och arbetsmetoder för de lärare som väljer att arbeta med automatisk bedömning. På så sätt kan även lärare förmedla denna kunskap till studenterna, så att användning och förståelse för den automatiserade återkoppling blir bättre. I denna rapport utformades en lärarguide samt studentguide i detta syfte.

Hur kan arbetsmaterial för lärare som använder automatisk återkoppling av programmeringsuppgifter i undervisningen utformas?

Denna studie visade på att det finns ett värde i att använda sig av automatisk återkoppling för programmeringsuppgifter inom högre studier. Därmed drogs slutsatsen att en guide för användning av Kattis bör utformas. Denna guide utformades baserat på intervjustudien, inläsning av material och diskussion med lärare. Arbetsmaterialet utmynnade i en guide för lärare respektive studenter.

7 Vidare forskning

De arbetsmetoder som utformats i samband med detta arbete ger utrymme och potential för vidareutveckling och testning. Dessa metoder skulle kunna implementeras på framtida kursomgångar som använder verktyget Kattis och utvärderas. Det finns även ett värde att se hur automatisk bedömning kan vidareutvecklas för att studenter ska lära sig programmering och datalogi på en mer teoretisk nivå.

Referenser

Bearman, M., Dawson, P., Ajjawi, R., Tai, J., & Boud, D. (2021). *Re-imagining University Assessment in a Digital World (The Enabling Power of Assessment)* (1st ed. 2020 ed.). Springer.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Burkle, M., & Cobo, C. (2018). Redefining Knowledge in the Digital Age. *Journal of New Approaches in Educational Research*, *7*(2), 79–80. https://doi.org/10.7821/naer.2018.7.294

Eckerdal, A., Thuné, M., & Berglund, A. (2005). What does it take to learn "programming thinking"? *ICER '05: Proceedings of the First International Workshop on Computing Education Research*, 135–142. https://doi.org/10.1145/1089786.1089799

Edwards, S. H. (2021). Automated Feedback, the Next Generation. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/3408877.3437225

Enström, E., Kreitz, G., Niemela, F., Soderman, P., & Kann, V. (2011). Five years with kattis — Using an automated assessment system in teaching. *2011 Frontiers in Education Conference (FIE)*. https://doi.org/10.1109/fie.2011.6142931

Enström, E., Kreitz, G., Niemelä, F., & Kann, V. (2010). Testdriven utbildning strukturerad formativ examination. Presented at the NU2010, Dialog för lärande. Stockholm, Swe. 13 - 15 oktober 2010.
Hämtad från: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-49550

Gikandi, J., Morrow, D., & Davis, N. (2011). Online formative assessment in higher education: A review of the literature. *Computers & Education*, *57*(4), 2333–2351. https://doi.org/10.1016/j.compedu.2011.06.004

Globala målen – För hållbar utveckling. (2022, March 24). Globala målen. https://www.globalamalen.se/

Kattis, Kattis. (2022). Kattis. Hämtad April 19, 2022, från https://open.kattis.com/
Kvale, S., Brinkmann, S., & Torhell, S. (2009). *Den kvalitativa forskningsintervjun* (2. uppl.. ed.). Lund: Studentlitteratur.

Paiva, J. C., Leal, J. P., & Figueira, L. (2022). Automated Assessment in Computer Science Education: A State-of-the-Art Review. *ACM Transactions on Computing Education*. https://doi.org/10.1145/3513140

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education - ITiCSE-WGR '07*. https://doi.org/10.1145/1345443.1345441

Riese, E., & Bälter, O. (2022). A Qualitative Study of Experienced Course Coordinators' Perspectives on Assessment in Introductory Programming Courses for Non-CS Majors. *ACM Transactions on Computing Education*. https://doi.org/10.1145/3517134

Skinner, B. F. (1968). *Undervisningsteknologi*. Studentlitteratur.

Säljö, R. (2017). Lärande : En introduktion till perspektiv och metaforer (1. uppl.. ed.).

Taras, M. (2005). Assessment – Summative and formative – Some theoretical reflections. *British Journal of Educational Studies*, *53*(4), 466–478. https://doi.org/10.1111/j.1467-8527.2005.00307.x

Vetenskapsrådet (2017). *God forskningsed*. Stockholm: Vetenskapsrådet.

von Hausswolff, K. (2021). Practical thinking while learning to program – novices' experiences and hands-on encounters. *Computer Science Education*, *32*(1), 128–152. https://doi.org/10.1080/08993408.2021.1953295

Wasik, S., Antczak, M., Badura, J., Laskowski, A., & Sternal, T. (2019). A Survey on Online Judge Systems and Their Applications. *ACM Computing Surveys*, *51*(1), 1–34. https://doi.org/10.1145/3143560

Wikberg-Nilsson, Å. W., Ericson, Å., & Törlind, P. (2021). *Design - Process och metod*. Studentlitteratur AB.

Zhou, W., Pan, Y., Zhou, Y., & Sun, G. (2018). The framework of a new online judge system for programming education. *Proceedings of ACM Turing Celebration Conference - China*. https://doi.org/10.1145/3210713.3210721

8 Bilagor

8.1 Bilaga 1
Intervjuguide

Genom litteraturstudien för denna rapport införskaffades förkunskap om ämnet Kattis, lärandet hos vuxna och digital bedömning inför denna intervjustudie.

Utifrån detta utformades följande intervjufrågor.

- Vad gör du när du inte blir godkänd?
- Förstår du varför du blev godkänd/ eller inte?
- Föredrar du att arbeta med Kattis, varför/varför inte?
- Vad skulle ni vilja ha för mer återkoppling?
- Någon annan återkoppling?
- Vad gör du om du får ett grönt kryss och två röda?
- Hur känner du dig när du får ett rött kryss eller grönt check?
- Vad tycker du inte fungerar med Kattis?
- Vad tycker du fungerar bra med Kattis?
- Känns det tryggt att blivit godkänd i Kattis innan redovisning?
- Har du reflekterat över värdet att använda Kattis som en portfölj av din kod?

Följande frågor ställdes för att samla information om intervjuobjekten. Där X indikerar namnet på den nybörjarkurs i datalogi som studien behandlar.

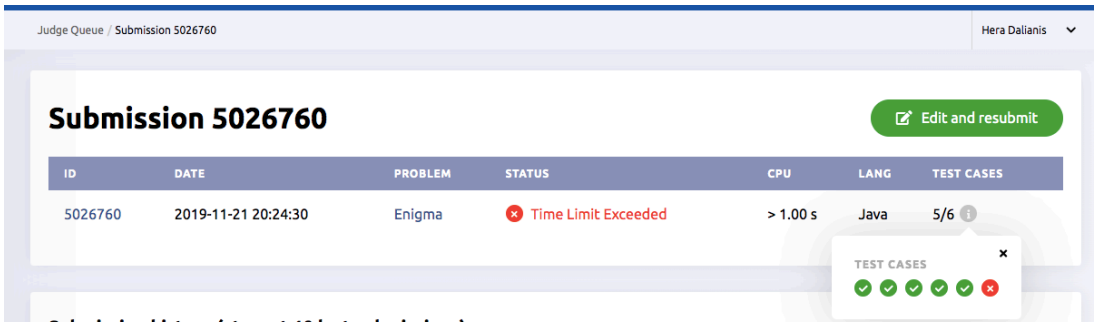
- Hur gammal är du?
- Vilken könsidentitet har du?
- Vilket program läser du?
- Är X är den första kursen där du använder Kattis?
- Vilket år befinner du dig på i din utbildning?
- Är X obligatorisk på ditt program? Eller valbar?
- Vilket betyg siktar du på i kursen?

8.2 Bilaga 2
Arbetsguide för lärare vid användning av Kattis

Kattis är en digital domare av programmeringsuppgifter. Antingen kan ett godkänd inlämning i Kattis räcka för ett godkänt betyg, eller så kan inlämningen kompletteras med en muntlig redovisning. Oavsett vilken av metoderna som du väljer för examination så är det av stor vikt att studenterna reflekterar själva när de löser labben. Ge studenterna **Studentguiden** för att kunna reflektera och lösa labben självständigt.

Hur fungerar Kattis?

När studenten skickar in sin kod så kompilerar Kattis programmet och kör programmet. Kattis skickar in indata och jämför därefter utdata med ett redan existerande svar, ett facit. Detta kallas för ett testfall. Kattis testar ett testfall i taget, tills samtliga testfall är godkända. Om Kattis stöter på ett testfall som inte är godkänt så slutar Kattis testa. På så sätt kan Kattis avgöra om programmet fungerar korrekt eller ej. De första testfallen kan studenten se i uppgiftsbeskrivningen, men det finns även några hemliga testfall där studenten inte vet vad som är input och förväntad output. Om ett testfall blivit godkänt så syns en grön bock i respektive ruta, om ett testfall blivit underkänt syns ett rött kryss. De testfall som inte hunnits testas är tomma. (Se figur 1)



FIGUR 1 - INLÄMNING I KATTIS

Använda redan existerande problem

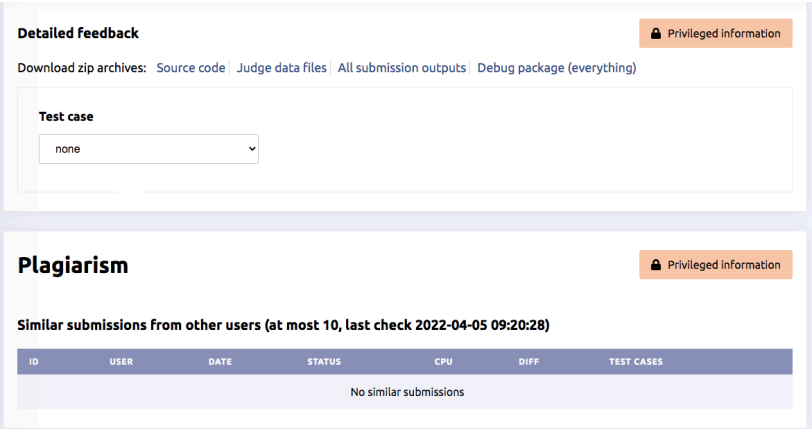
1. Kontakta Kattis för att lägga upp en ny kurs eller kursomgång.
2. Hitta de problem du vill använda dig av.
3. Kontakta din Kattis administratör för att lägga in nya de nya problemen i din kursomgång.

Skapa egna labbar i Kattis

1. Ladda ner verifieringsprogrammet för Kattis och exempelprogram som finns på följande länk: https://github.com/Kattis/problemtools
2. Kopiera ett av exempelproblemen och ändra för att matcha för den labben du vill utforma.
   Efter justeringen bör du minst ha följande:
   - En beskrivning av labben
   - Förväntad indata samt utdata i form av ett publikt testfall och ett hemligt testfall.
   - Ett godkänt program för labben
   - Skapa en verifieringsfil som beskriver indata.
3. Validera programmet med hjälp av *verifyproblem.py*
4. Kontakta Kattis på contact@kattis.com för att lägga upp labben på kth.kattis.se

Hjälpa studenter

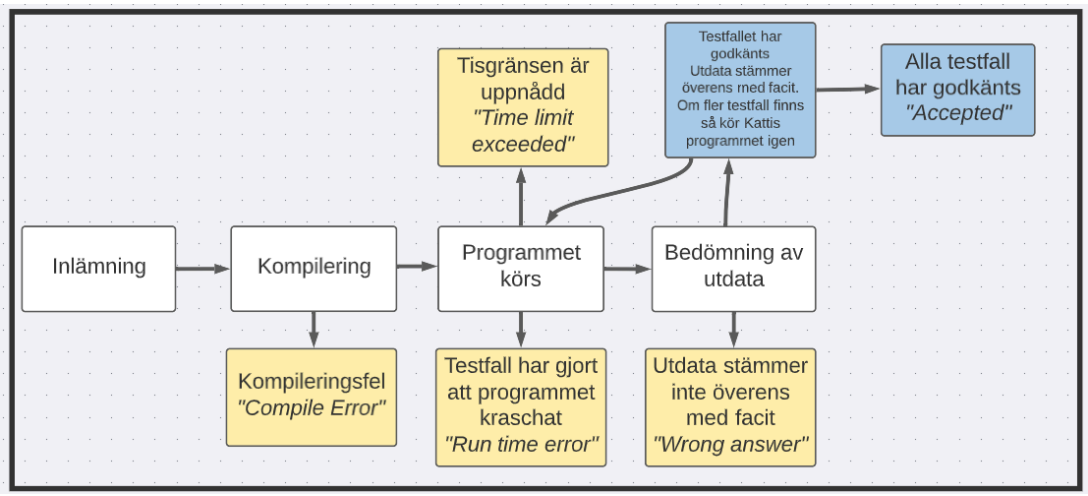
För att studenter ska kunna använda verktyget Kattis framgångsrikt behöver de kunskap om verktyget. **Studentguiden** innehåller all nödvändig information om verktyget och kan ges ut till studenterna. I studentguiden informeras studenterna att när de söker hjälp även bifoga en länk till inlämningen för problemet. Det kan se ut såhär för dig som är tillagd som lärare eller lärarassistent i kursen när du klickar på länken (Se figur 1). Om du inte lyckas komma åt inlämningen så är studenten inte registrerad på kursen i Kattis och måste göra det enligt instruktionerna i **Studentguiden. Notera att en registrering i Canvas eller Ladok innebär inte en automatisk registrering i Kattis!**
Studenter får **inte** ta del av *Privileged information* (Se figur 2) eller vad deras program ger för output eller de hemliga testfallen. I *Privileged information* finner du information om varför programmet inte blivit godkänt, använd dig av denna information för att handleda studenterna i rätt riktning. Under *Input* ser du vad Kattis skickar in för input, under *Reference answer* finner du förväntad output och under *Submission output* finner du outputen från studentens inlämning. Du får **inte** dela med dig om information om förväntad output, då denna information i de hemliga testfallen inte syns för studenterna. Om du vill se samtliga hemliga testfall kan du gå in på problembeskrivningen och ladda ner *All data files*.



FIGUR 2 - PRIVILEGED INFORMATION

Kattis felmeddelande

Olika felmeddelande beror på olika saker, se figuren nedan för när respektive felmeddelande kan uppstå (Se figur 3).



FIGUR 2 - KATTIS FELMEDDELANDE

Compile Error
Komplieringsfel eller ”Compile Error” inträffar under själva kompileringen och beror på att Kattis inte kan kompilera programmet.

Time Limit Exceeded
Time Limit Exceeded beror på att ditt program överskrider den tidsgräns som är satt för programmet, programmet är alltså för långsamt.

Run time error
Run time error beror på att ett testfall gjort att programmet kraschar.

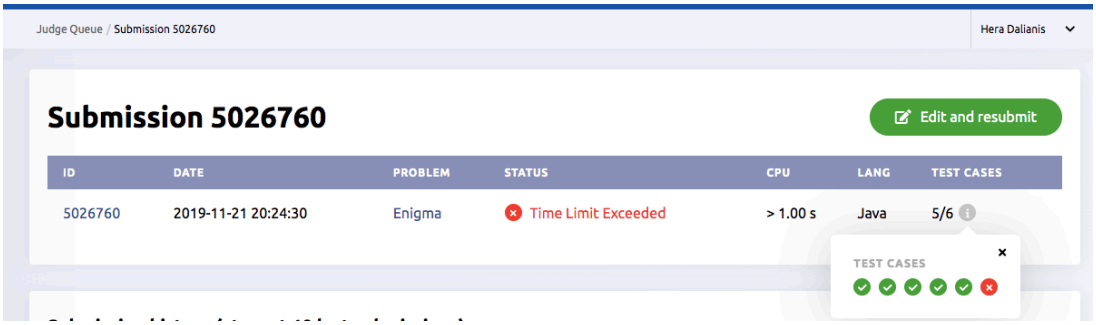
Wrong answer
Wrong answer innebär att output för en viss input inte matchar Kattis facit för output.

8.3 Bilaga 3
Studentguide för Kattis

Kattis är en digital domare av programmeringsuppgifter. För att bli godkänd på en laboration så behöver du visa upp din godkända inlämning i Kattis för labbassen på redovisningen.

Hur fungerar Kattis?

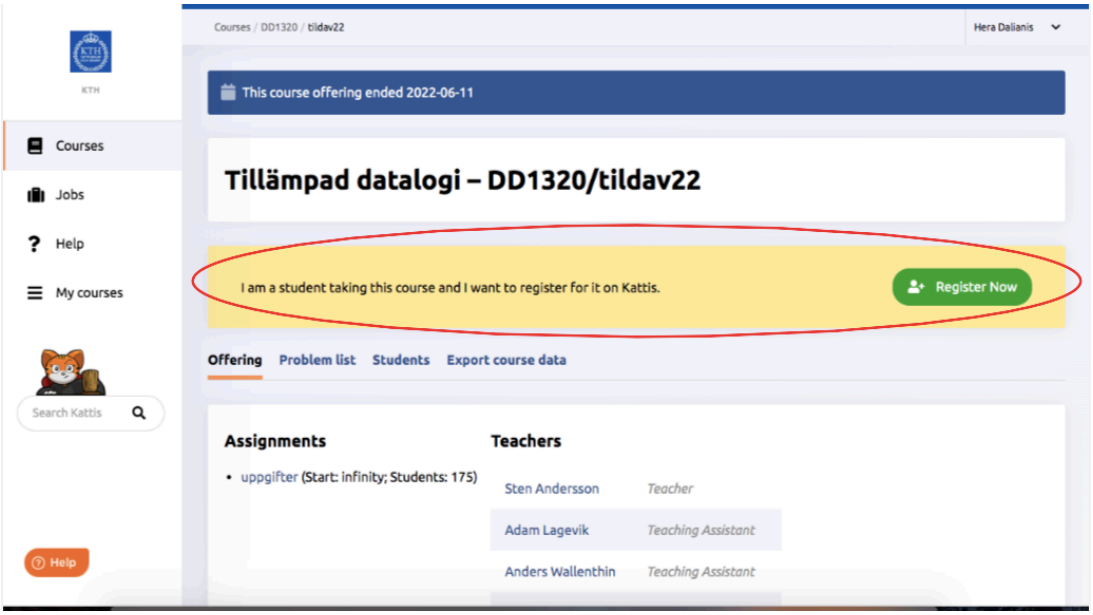
När du skickar in din kod så kompilerar och kör Kattis programmet. Kattis skickar in indata och jämför därefter utdata med ett redan existerande svar - ett facit, detta kallas för ett testfall. Kattis testar ett testfall i taget, tills samtliga testfall är godkända eller om Kattis stöter på ett testfall som inte är godkänt så slutar Kattis testa. På så sätt kan Kattis avgöra om ditt program fungerar korrekt eller ej. De första testfallen kan du se i uppgiftsbeskrivningen, men det finns även några hemliga testfall där du inte vet vad som är input och förväntad output. Om ett testfall blivit godkänd så syns en grön bock i respektive ruta, om ett testfall blivit underkänt syns ett rött kryss. De testfall som inte hunnits testas är tomma. (Se figur 1).



FIGUR 1 - INLÄMNING I KATTIS

Hur använder jag Kattis?

För att kunna använda Kattis i den här kursen så behöver du vara **inloggad** på Kattis KTH https://kth.kattis.com/ och **registrerad** på kursen i KTH Kattis. För att registrera dig, klicka på kursen och välj din kursomgång (t.ex. tildav22), klicka sedan att du vill registrera dig. **Notera att en registrering i Canvas eller Ladok innebär inte en automatisk registrering i Kattis!**



FIGUR 2 – REGISTRERING I KATTIS

Under **Problem list** finner du samtliga labblydelser. Klicka på den labben som du vill göra. För att lämna in ett program, klicka på submit och ladda upp din fil.

- Filen med programmet ska sluta på t.ex. *.java .py*. Kattis kan inte kompilera txt-filer.
- Om ditt program använder importerade filer så behövs även dessa skickas in. Huvudprogrammet ska då kallas för **main.** För att ladda upp flera filer samtidigt markera dessa samtidigt och dra till inlämningsrutan.
- Ange även det språk du skrivit ditt program i.
- Du behöver **inte** skicka in egna testfall.

Innan du börjar med kursens labbar, gör följande problem.

- *Hello, World!* - https://kth.kattis.com/problems/hello
- *A Different Problem* - https://kth.kattis.com/problems/different

Kör ditt program som Kattis

Om du vill se exakt vad Kattis ser så kan du kompilera och köra ditt program i terminalen på din dator. Antingen kan du skriva in testfallen för hand eller köra testfallen som ligger som filer på Kattis.

Testa filerna från Kattis

5. Ladda ner *sample files* från problemet, där ligger indata och förväntad output för respektive av de kända testfallen.
6. Lägg samtliga filer från *sample files* i samma mapp som din labb.
7. Öppna terminalen och gå in i mappen.

8. I terminal fönstret skriv följande (om du t.ex. skriver i python), där 1.in är namnet på indatafilen för testfallet du vill testa från sample files. Kommandot gör att 1.out skapas och innehåller outputen från ditt program.

   python3 labbnamn.py < 1.in > 1.out

9. För att därefter jämföra din output med den förväntade outputen skriv följande, där 1.ans är den förväntade outputen för testfallet.

   diff 1.out 1.ans

10. Om terminalen inte gör något är ditt program korrekt för det testfallet
11. Om du får output, så indikerar siffrorna vilken rad som är fel i outputen och den första raden vad ditt program gav för output och andra raden vad som finns i facit.

Skriva in testfall för hand

1. Öppna terminalen och gå in i mappen där din labb ligger
2. Skriv (om du t.ex. skriver i python):

   python3 labbnamn.py

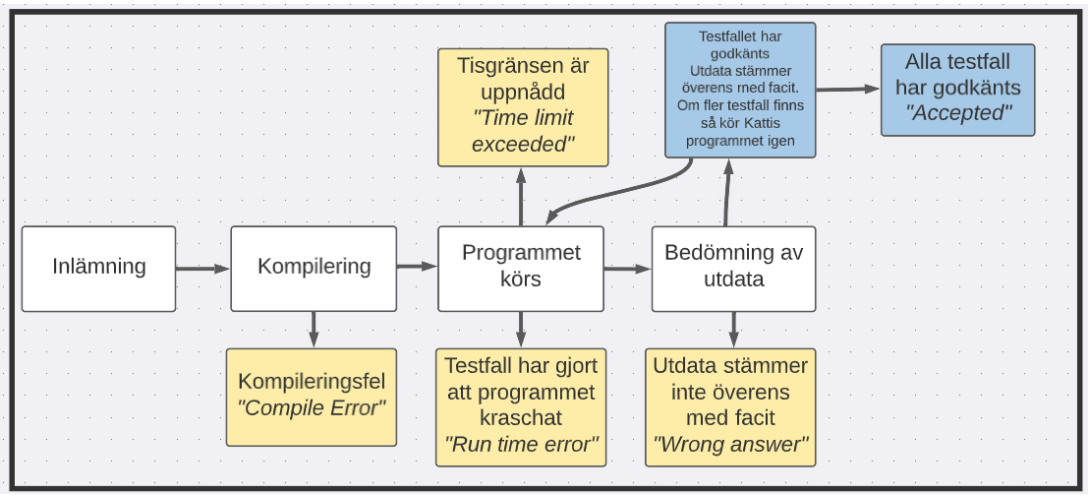
3. Skriv in din input och tryck enter, så får du output från programmet.

Hjälp

För att få hjälp skicka alltid med länken till inlämningssidan.

Felmeddelande i Kattis

Olika felmeddelande beror på olika saker, se figuren nedan för när respektive felmeddelande kan uppstå. (Se figur 3)



FIGUR 3 – FELMEDDELANDE I KATTIS

Compile Error

Komplieringsfel eller ”Compile Error” inträffar under själva kompileringen och beror på att Kattis inte kan kompilera programmet. Detta kan vara ett så enkelt som att du råkat skriva en extra punkt någonstans som gör att programmet kraschar. Testa att kompilera ditt program i din egen kompilator eller i terminalfönstret.

Time Limit Exceeded

Time Limit Exceeded beror på att ditt program överskrider den tidsgräns som är satt för programmet, ditt program är för långsamt. Se över om du har många nästlade while eller for-loopar. Dessa är ofta långsamma, speciellt för stora inputs. Fundera på hur du kan ändra din algoritm för att göra ditt program snabbare!

Run time error

Run time error beror på att ett testfall gjort att programmet kraschar. Testa att skicka in olika konstiga inputs, exempelvis en tom sträng. Hur hanterar ditt program om den förväntar ett tecken eller ett heltal och det inte finns något?

Wrong answer

Wrong answer innebär att output för en viss input inte matchar Kattis facit för output. Skriv egna testfall och skicka in i ditt program, ger de rätt output? Se kapitlet *Kör ditt program som Kattis* för tips om testning.