

1 Удирдамж

Зорилго: Энэхүү лабораторийн даалгаврын зорилго нь TinyPL хэмээх тоглоомын програмчлалын хэлний интерпретаторыг хэрэгжүүлнэ.

Тодорхойлолт: TinyPL нь хувьсагч зарладаг, хувьсагчид утга олгодог, функц зарладаг, функцийг дууддаг байна. Хэрэглэгчийн тодорхойлсон функцуудыг шатлах боломжтой бөгөөд ямар нэг параметр дамжуулахгүй. Хэрэглэгчийн тодорхойлсон функцүүдээс гадна нэг хувьсагчийг параметр болгон аваад утгыг нь дэлгэцэнд гаргадаг print функц байна. Энэ хэл нь хяналтын урсгалын токenuуд (жишээ нь, if, while) эсвэл бодит программчлалын хэлний шинж чанар агуулахгүй. Програмын загвар прожектийг хавсралт файлаас авна. Дараах зурагт хэлзүйн дүрмийг харуулав.

```
P → Stmt*
Stmt → VarDecl | Assign | FctDecl | Call
FctDecl → function Name Stmt*
VarDecl → var Name;
Assign → Name = Val;
Val → Integer | Name
Call → Name()
```

TinyPL -ээр бичигдсэн программд дараах зарчмыг баримтална.

- Хувьсагч бүрийг хэрэглэхээс нь өмнө зарласан байна.
- Функц бүрийг хэрэглэхээс нь өмнө зарласан байна.
- Программ дахь функцийн нэр болон хувьсагчийн нэрсийн олонлог нь тусдаа байна

Жишээлбэл, доорх псевдо код нь хоёр функц, хоёр хувьсагч зарлаад дараа нь *a* функцийг дууддаг TinyPL программ бөгөөд энд *a* функц дотроос *b* функцийг дуудаж байна. Энэ програмын ажиллагаа нь хэлний холболтын болон хамрах хүрээний дүрмээс хамаарна.

```
1 function a() {
2     var x;
3     x = 55;
4     b();
5 }
6
7 function b() {
8     print(x);
9 }
10 var x;
11 x=1;
12 a();
```

Санамж: Даалгаврыг *Node.js* $\geq v14$ ашиглан гүйцэтгэнэ.

2 Гүйцэтгэх даалгавар

Даалгаврыг доорх Milestone-ны дагуу TinyPL-ийн интерпретаторын хоёр хувилбарыг хэрэгжүүлнэ. Интерпретатор бүр *P* програмын задлан шинжилгээний модыг оролтоор

авч, P -ийн үр дүнг гаралт болгон өгдөг. TinyPL-д үр дүнг гаргах ганц арга бол `print` дуудалт тул гаралтыг мөрүүдийн жагсаалт байхаар хэвлэнэ.

Milestone 1: Динамик хамрах хүрээтэй TinyPL

Эхний интерпретаторын TinyPL хэл нь динамик хамрах хүрээг ашигладаг гэж үзнэ. Жишээлбэл, дээр үзүүлсэн шаталсан функц бүхий жишээ программ "55" үр дүнг гаргана. Шийдэлээ өгсөн Milestone1 классыг ашиглан хэрэгжүүлнэ.

Milestone 2: Статик хамрах хүрээтэй TinyPL

Хоёрдахь интерпретаторын TinyPL хэл нь статик хамрах хүрээг ашигладаг гэж үзнэ. Дээр үзүүлсэн шаталсан функц бүхий жишээ программын хувьд интерпретатор нь "1" үр дүнг гаргана. Шийдлээ өгсөн Milestone2 классыг ашиглан шийдлээ хэрэгжүүлнэ.

Программчлалын орчин

Энэ даалгаврыг JavaScript дээр хэрэгжүүлэх ёстой. Тиймээс `Node.js ≥ v14` болон `npm` суулгасан байна. `Node.js` болон `npm`-г суулгахын тулд [үүн шиг](#) хичээлүүдийг онлайнаар үзээрэй. Мөн `antlr4`-ийг суулгахын тулд `npm install`-г үндсэн хавтсанд ажиллуулна.

Өгөгдсөн кодууд болон Туслах классууд

Хэрэгжилтийг хялбар байлахын тулд `util.js` файлыг өгсөн бөгөөд энэ нь тэмдэгт мөр хэлбэрээр өгөгдсөн программыг задлан шинжилгээний мод руу хөрвүүлнэ. ANTLR4-ээр автоматаар үүсгэгдсэн TinyPL-д зориулсан задлан шинжлэгчээр хэрэглэж байна. Хэрэгжүүлэхэд хэрэг болох `TinyPLListener` классыг хараарай. Энэ классыг өргөтгөх хэрэгтэй. Дараа нь классын уг хувилбар `antlr.ParseTreeWalker` классруу өгөгдөх бөгөөд энэ нь тодорхой төрлийн зангилаанд зочлох бүрт үйлдэл гүйцэтгэнэ. Эдгээр классаас гадна **ANTLR4**-аас өгөгдсөн бусад классууд болон `Node.js` суулгасан модулиудын бүх классыг ашиглах боломжтой, гэхдээ өөр ямар ч гадаад сан ашиглахгүй.

Нэмэлт тэмдэглэл (дасгалд шаардлагагүй): Хэлний синтаксийг **ANTLR4** ашиглан хэрэгжүүлдэг. Токенуудын тодорхойлолт болон дүрмийн дүрмийг `TinyPL.g4` файлаас харна уу. Хэрэв сонирхож байгаа бол тухайн хэлний задлагчийг өөрөө дахин үүсгэж болно: 1) сайтаас нь [antlr-4.12.0-complete.jar](#)-г татаж авна; 2) дараах командыг ажиллуулна: `java-jar antlr-4.12.0-complete.jar-Dlanguage=JavaScript src/interpreter/TinyPL.g4`.

Тестчилэл

Чухал Milestone бүрийн тестийн тохиолдлыг өгсөн: `Milestone1.test.js`, болон `Milestone2.test.js`. Тестийн тохиолдлуудыг ажиллуулахын тулд даалгавар үндсэн хавтсанд дараах командыг гүйцэтгэнэ: `node Milestone1.test.js` болон `node Milestone2.test.js`. Интерпретатораа өөр төрлийн программ дээр ажиллуулахын тулд нэмэлт тестийн тохиолдлууд хэрэгжүүлнэ үү.

ТАНД АМЖИЛТ ХҮСЬЕ!