

编译小组作业——代码高亮与提示程序

陈俊哲 2020010964 软件 01
梁 焯 2020080093 软件 01
田正祺 2020080095 软件 01

1 概述

此项目实现了 C++ 的词法分析器、语法分析器、以及代码高亮与提示程序。此项目的大部分逻辑实现于 Python 3，其中关于编译的算法与逻辑使用了 **PLY**。实现过程中主要参考的资料是 C++20 标准 **N4860** 版本。

2 词法分析

2.1 实现

所有有关词法分析的代码存放在 **lexer** 下。标准中的第五章中定义了 C++ 的词法规则。其中描述的记号包括标识符 (identifier)、关键字 (keyword)、常量 (literal)、运算符 (operator)，其中常量分为整数 (integer)、字符 (character)、浮点数 (floating-point)、字符串 (string)、布尔 (boolean)、指针 (pointer)、用户定义 (user-defined)。各种记号的定义都存放在 **lexer** 下对应的 Python 文件中，并使用 PLY 的 **lex** 包进行结合与词法分析。开发过程中发现 **lex** 最后生成的主正则表达式中每种记号的顺序与这些在代码中规则定义的顺序可能不一样，导致比如 `char c = u8't'` 中的 `u8` 被识别为标识符，而正确的词法分析结果是整个 `u8't'` 是一个字符常量。因此改变了 **myply/lex.py** 中的 **lex** 函数，让用户明确地定义记号类型的识别顺序。因此此项目 必须使用 **myply** 下的 PLY 包。

2.2 难点

阅读标准之前，我们以为 C++ 标识符可以简简单单地使用 `[_a-zA-Z][_a-zA-Z0-9]*` 的正则表达式匹配。但是 C++20 将可以作为标识符的符号大幅增加，甚至中文字与表情是合法的标识符字符。为了强调标准所允许的范围，以下显示的是符合标准的正则表达式：

```
((?![\u0300-\u036F\u1DC0-\u1DFF\u20D0-\u20FF\uFE20-\uFE2F])([_a-zA-Z]|[\u00A8\u00AA\u00AD\u00AF\u00B2-\u00B5\u00B7-\u00BA\u00BC-\u00BE\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u00FF\u0100-\u0167\u0168-\u018D\u018F-\u01FF\u200B-\u200D\u202A-\u202E\u203F-\u2040\u2054\u2060-\u206F\u2070-\u218F\u2460-\u24FF\u2776-\u2793\u2C00-\u2DFF\u2E80-\u2FFF\u3004-\u3007\u3021-\u302F\u3031-\u303F\u3041-\u304F\u3091-\u309F\u3105-\u310F\u3191-\u319F\u3201-\u321F\u327E\u328E\u3290-\u329F\u3300-\u3305\u3307-\u330F\u3311-\u3313\u3315-\u3326\u3329-\u3331\u3334-\u3337\u3339-\u3342\u3345-\u3348\u334D\u334F\u3351-\u3353\u3356-\u3359\u3360-\u3369\u3370-\u337A\u337D\u337F\u3381-\u3383\u3386-\u3388\u338B\u338D\u338F\u3391-\u3393\u3396-\u3398\u339B\u339D\u339F\u33A1-\u33A3\u33A6-\u33A8\u33AB\u33AD\u33AF\u33B2-\u33B5\u33B7-\u33BA\u33BC-\u33BE\u33C0-\u33C6\u33C8-\u33CE\u33D0-\u33D6\u33D8-\u33DE\u33E0-\u33E6\u33E8-\u33EE\u33F0-\u33F6\u33F8-\u33FE\u3400-\u3405\u3407-\u340F\u3411-\u3413\u3415-\u3416\u3418-\u3419\u341B-\u341C\u341F\u3421-\u3422\u3424-\u3426\u3429-\u342A\u342C-\u342F\u3431-\u3432\u3434-\u3436\u3439-\u343A\u343C-\u343D\u343F\u3441-\u3442\u3444-\u3446\u3449-\u344A\u344C-\u344D\u344F\u3451-\u3452\u3454-\u3455\u3457-\u3458\u345A-\u345B\u345D-\u345E\u345F\u3461-\u3462\u3464-\u3465\u3467-\u3468\u346A-\u346B\u346D-\u346E\u346F\u3471-\u3472\u3474-\u3475\u3477-\u3478\u347A-\u347B\u347D-\u347E\u3481-\u3482\u3484-\u3485\u3487-\u3488\u348A-\u348B\u348D-\u348E\u348F\u3491-\u3492\u3494-\u3495\u3497-\u3498\u349A-\u349B\u349D-\u349E\u34A1-\u34A2\u34A4-\u34A5\u34A7-\u34A8\u34AA-\u34AB\u34AD-\u34AE\u34B1-\u34B2\u34B4-\u34B5\u34B7-\u34B8\u34BA-\u34BB\u34BD-\u34BE\u34C0-\u34C1\u34C3-\u34C4\u34C6-\u34C7\u34C9-\u34CA\u34CB-\u34CC\u34CE-\u34CF\u34D1-\u34D2\u34D4-\u34D5\u34D7-\u34D8\u34DA-\u34DB\u34DD-\u34DE\u34E0-\u34E1\u34E3-\u34E4\u34E6-\u34E7\u34E9-\u34EA\u34EB-\u34EC\u34ED-\u34EE\u34F0-\u34F1\u34F3-\u34F4\u34F6-\u34F7\u34F9-\u34FA\u34FB-\u34FC\u34FD-\u34FE\u34FF\u3500-\u3501\u3503-\u3504\u3506-\u3507\u3509-\u350A\u350C-\u350D\u350F\u3511-\u3512\u3514-\u3515\u3517-\u3518\u351A-\u351B\u351D-\u351E\u351F\u3521-\u3522\u3524-\u3525\u3527-\u3528\u352A-\u352B\u352D-\u352E\u352F\u3531-\u3532\u3534-\u3535\u3537-\u3538\u353A-\u353B\u353D-\u353E\u353F\u3541-\u3542\u3544-\u3545\u3547-\u3548\u354A-\u354B\u354D-\u354E\u354F\u3551-\u3552\u3554-\u3555\u3557-\u3558\u355A-\u355B\u355D-\u355E\u355F\u3561-\u3562\u3564-\u3565\u3567-\u3568\u356A-\u356B\u356D-\u356E\u356F\u3571-\u3572\u3574-\u3575\u3577-\u3578\u357A-\u357B\u357D-\u357E\u357F\u3581-\u3582\u3584-\u3585\u3587-\u3588\u358A-\u358B\u358D-\u358E\u358F\u3591-\u3592\u3594-\u3595\u3597-\u3598\u359A-\u359B\u359D-\u359E\u359F\u35A1-\u35A2\u35A4-\u35A5\u35A7-\u35A8\u35AA-\u35AB\u35AD-\u35AE\u35B1-\u35B2\u35B4-\u35B5\u35B7-\u35B8\u35BA-\u35BB\u35BD-\u35BE\u35C0-\u35C1\u35C3-\u35C4\u35C6-\u35C7\u35C9-\u35CA\u35CB-\u35CC\u35CE-\u35CF\u35D1-\u35D2\u35D4-\u35D5\u35D7-\u35D8\u35DA-\u35DB\u35DD-\u35DE\u35E0-\u35E1\u35E3-\u35E4\u35E6-\u35E7\u35E9-\u35EA\u35EB-\u35EC\u35ED-\u35EE\u35F0-\u35F1\u35F3-\u35F4\u35F6-\u35F7\u35F9-\u35FA\u35FB-\u35FC\u35FD-\u35FE\u35FF\u3600-\u3601\u3603-\u3604\u3606-\u3607\u3609-\u360A\u360C-\u360D\u360F\u3611-\u3612\u3614-\u3615\u3617-\u3618\u361A-\u361B\u361D-\u361E\u361F\u3621-\u3622\u3624-\u3625\u3627-\u3628\u362A-\u362B\u362D-\u362E\u362F\u3631-\u3632\u3634-\u3635\u3637-\u3638\u363A-\u363B\u363D-\u363E\u363F\u3641-\u3642\u3644-\u3645\u3647-\u3648\u364A-\u364B\u364D-\u364E\u364F\u3651-\u3652\u3654-\u3655\u3657-\u3658\u365A-\u365B\u365D-\u365E\u365F\u3661-\u3662\u3664-\u3665\u3667-\u3668\u366A-\u366B\u366D-\u366E\u366F\u3671-\u3672\u3674-\u3675\u3677-\u3678\u367A-\u367B\u367D-\u367E\u367F\u3681-\u3682\u3684-\u3685\u3687-\u3688\u368A-\u368B\u368D-\u368E\u368F\u3691-\u3692\u3694-\u3695\u3697-\u3698\u369A-\u369B\u369D-\u369E\u369F\u36A1-\u36A2\u36A4-\u36A5\u36A7-\u36A8\u36AA-\u36AB\u36AD-\u36AE\u36B1-\u36B2\u36B4-\u36B5\u36B7-\u36B8\u36BA-\u36BB\u36BD-\u36BE\u36C0-\u36C1\u36C3-\u36C4\u36C6-\u36C7\u36C9-\u36CA\u36CB-\u36CC\u36CE-\u36CF\u36D1-\u36D2\u36D4-\u36D5\u36D7-\u36D8\u36DA-\u36DB\u36DD-\u36DE\u36E0-\u36E1\u36E3-\u36E4\u36E6-\u36E7\u36E9-\u36EA\u36EB-\u36EC\u36ED-\u36EE\u36F0-\u36F1\u36F3-\u36F4\u36F6-\u36F7\u36F9-\u36FA\u36FB-\u36FC\u36FD-\u36FE\u36FF\u3700-\u3701\u3703-\u3704\u3706-\u3707\u3709-\u370A\u370C-\u370D\u370F\u3711-\u3712\u3714-\u3715\u3717-\u3718\u371A-\u371B\u371D-\u371E\u371F\u3721-\u3722\u3724-\u3725\u3727-\u3728\u372A-\u372B\u372D-\u372E\u372F\u3731-\u3732\u3734-\u3735\u3737-\u3738\u373A-\u373B\u373D-\u373E\u373F\u3741-\u3742\u3744-\u3745\u3747-\u3748\u374A-\u374B\u374D-\u374E\u374F\u3751-\u3752\u3754-\u3755\u3757-\u3758\u375A-\u375B\u375D-\u375E\u375F\u3761-\u3762\u3764-\u3765\u3767-\u3768\u376A-\u376B\u376D-\u376E\u376F\u3771-\u3772\u3774-\u3775\u3777-\u3778\u377A-\u377B\u377D-\u377E\u377F\u3781-\u3782\u3784-\u3785\u3787-\u3788\u378A-\u378B\u378D-\u378E\u378F\u3791-\u3792\u3794-\u3795\u3797-\u3798\u379A-\u379B\u379D-\u379E\u379F\u37A1-\u37A2\u37A4-\u37A5\u37A7-\u37A8\u37AA-\u37AB\u37AD-\u37AE\u37B1-\u37B2\u37B4-\u37B5\u37B7-\u37B8\u37BA-\u37BB\u37BD-\u37BE\u37C0-\u37C1\u37C3-\u37C4\u37C6-\u37C7\u37C9-\u37CA\u37CB-\u37CC\u37CE-\u37CF\u37D1-\u37D2\u37D4-\u37D5\u37D7-\u37D8\u37DA-\u37DB\u37DD-\u37DE\u37E0-\u37E1\u37E3-\u37E4\u37E6-\u37E7\u37E9-\u37EA\u37EB-\u37EC\u37ED-\u37EE\u37F0-\u37F1\u37F3-\u37F4\u37F6-\u37F7\u37F9-\u37FA\u37FB-\u37FC\u37FD-\u37FE\u37FF\u3800-\u3801\u3803-\u3804\u3806-\u3807\u3809-\u380A\u380C-\u380D\u380F\u3811-\u3812\u3814-\u3815\u3817-\u3818\u381A-\u381B\u381D-\u381E\u381F\u3821-\u3822\u3824-\u3825\u3827-\u3828\u382A-\u382B\u382D-\u382E\u382F\u3831-\u3832\u3834-\u3835\u3837-\u3838\u383A-\u383B\u383D-\u383E\u383F\u3841-\u3842\u3844-\u3845\u3847-\u3848\u384A-\u384B\u384D-\u384E\u384F\u3851-\u3852\u3854-\u3855\u3857-\u3858\u385A-\u385B\u385D-\u385E\u385F\u3861-\u3862\u3864-\u3865\u3867-\u3868\u386A-\u386B\u386D-\u386E\u386F\u3871-\u3872\u3874-\u3875\u3877-\u3878\u387A-\u387B\u387D-\u387E\u387F\u3881-\u3882\u3884-\u3885\u3887-\u3888\u388A-\u388B\u388D-\u388E\u388F\u3891-\u3892\u3894-\u3895\u3897-\u3898\u389A-\u389B\u389D-\u389E\u389F\u38A1-\u38A2\u38A4-\u38A5\u38A7-\u38A8\u38AA-\u38AB\u38AD-\u38AE\u38B1-\u38B2\u38B4-\u38B5\u38B7-\u38B8\u38BA-\u38BB\u38BD-\u38BE\u38C0-\u38C1\u38C3-\u38C4\u38C6-\u38C7\u38C9-\u38CA\u38CB-\u38CC\u38CE-\u38CF\u38D1-\u38D2\u38D4-\u38D5\u38D7-\u38D8\u38DA-\u38DB\u38DD-\u38DE\u38E0-\u38E1\u38E3-\u38E4\u38E6-\u38E7\u38E9-\u38EA\u38EB-\u38EC\u38ED-\u38EE\u38F0-\u38F1\u38F3-\u38F4\u38F6-\u38F7\u38F9-\u38FA\u38FB-\u38FC\u38FD-\u38FE\u38FF\u3900-\u3901\u3903-\u3904\u3906-\u3907\u3909-\u390A\u390C-\u390D\u390F\u3911-\u3912\u3914-\u3915\u3917-\u3918\u391A-\u391B\u391D-\u391E\u391F\u3921-\u3922\u3924-\u3925\u3927-\u3928\u392A-\u392B\u392D-\u392E\u392F\u3931-\u3932\u3934-\u3935\u3937-\u3938\u393A-\u393B\u393D-\u393E\u393F\u3941-\u3942\u3944-\u3945\u3947-\u3948\u394A-\u394B\u394D-\u394E\u394F\u3951-\u3952\u3954-\u3955\u3957-\u3958\u395A-\u395B\u395D-\u395E\u395F\u3961-\u3962\u3964-\u3965\u3967-\u3968\u396A-\u396B\u396D-\u396E\u396F\u3971-\u3972\u3974-\u3975\u3977-\u3978\u397A-\u397B\u397D-\u397E\u397F\u3981-\u3982\u3984-\u3985\u3987-\u3988\u398A-\u398B\u398D-\u398E\u398F\u3991-\u3992\u3994-\u3995\u3997-\u3998\u399A-\u399B\u399D-\u399E\u399F\u39A1-\u39A2\u39A4-\u39A5\u39A7-\u39A8\u39AA-\u39AB\u39AD-\u39AE\u39B1-\u39B2\u39B4-\u39B5\u39B7-\u39B8\u39BA-\u39BB\u39BD-\u39BE\u39C0-\u39C1\u39C3-\u39C4\u39C6-\u39C7\u39C9-\u39CA\u39CB-\u39CC\u39CE-\u39CF\u39D1-\u39D2\u39D4-\u39D5\u39D7-\u39D8\u39DA-\u39DB\u39DD-\u39DE\u39E0-\u39E1\u39E3-\u39E4\u39E6-\u39E7\u39E9-\u39EA\u39EB-\u39EC\u39ED-\u39EE\u39F0-\u39F1\u39F3-\u39F4\u39F6-\u39F7\u39F9-\u39FA\u39FB-\u39FC\u39FD-\u39FE\u39FF\u3A00-\u3A01\u3A03-\u3A04\u3A06-\u3A07\u3A09-\u3A0A\u3A0C-\u3A0D\u3A0F\u3A11-\u3A12\u3A14-\u3A15\u3A17-\u3A18\u3A1A-\u3A1B\u3A1D-\u3A1E\u3A1F\u3A21-\u3A22\u3A24-\u3A25\u3A27-\u3A28\u3A2A-\u3A2B\u3A2D-\u3A2E\u3A2F\u3A31-\u3A32\u3A34-\u3A35\u3A37-\u3A38\u3A3A-\u3A3B\u3A3D-\u3A3E\u3A3F\u3A41-\u3A42\u3A44-\u3A45\u3A47-\u3A48\u3A4A-\u3A4B\u3A4D-\u3A4E\u3A4F\u3A51-\u3A52\u3A54-\u3A55\u3A57-\u3A58\u3A5A-\u3A5B\u3A5D-\u3A5E\u3A5F\u3A61-\u3A62\u3A64-\u3A65\u3A67-\u3A68\u3A6A-\u3A6B\u3A6D-\u3A6E\u3A6F\u3A71-\u3A72\u3A74-\u3A75\u3A77-\u3A78\u3A7A-\u3A7B\u3A7D-\u3A7E\u3A7F\u3A81-\u3A82\u3A84-\u3A85\u3A87-\u3A88\u3A8A-\u3A8B\u3A8D-\u3A8E\u3A8F\u3A91-\u3A92\u3A94-\u3A95\u3A97-\u3A98\u3A9A-\u3A9B\u3A9D-\u3A9E\u3A9F\u3AA1-\u3AA2\u3AA4-\u3AA5\u3AA7-\u3AA8\u3AAA-\u3AAB\u3AAD-\u3AAE\u3AB1-\u3AB2\u3AB4-\u3AB5\u3AB7-\u3AB8\u3ABA-\u3ABB\u3ABD-\u3ABE\u3AC0-\u3AC1\u3AC3-\u3AC4\u3AC6-\u3AC7\u3AC9-\u3ACA\u3ACB-\u3ACC\u3ACE-\u3ACF\u3AD1-\u3AD2\u3AD4-\u3AD5\u3AD7-\u3AD8\u3ADA-\u3ADB\u3ADD-\u3ADE\u3AE0-\u3AE1\u3AE3-\u3AE4\u3AE6-\u3AE7\u3AE9-\u3AEA\u3AEB-\u3AEC\u3AED-\u3AEE\u3AF0-\u3AF1\u3AF3-\u3AF4\u3AF6-\u3AF7\u3AF9-\u3AFA\u3AFB-\u3AFC\u3AFD-\u3AFE\u3AFF\u3B00-\u3B01\u3B03-\u3B04\u3B06-\u3B07\u3B09-\u3B0A\u3B0C-\u3B0D\u3B0F\u3B11-\u3B12\u3B14-\u3B15\u3B17-\u3B18\u3B1A-\u3B1B\u3B1D-\u3B1E\u3B1F\u3B21-\u3B22\u3B24-\u3B25\u3B27-\u3B28\u3B2A-\u3B2B\u3B2D-\u3B2E\u3B2F\u3B31-\u3B32\u3B34-\u3B35\u3B37-\u3B38\u3B3A-\u3B3B\u3B3D-\u3B3E\u3B3F\u3B41-\u3B42\u3B44-\u3B45\u3B47-\u3B48\u3B4A-\u3B4B\u3B4D-\u3B4E\u3B4F\u3B51-\u3B52\u3B54-\u3B55\u3B57-\u3B58\u3B5A-\u3B5B\u3B5D-\u3B5E\u3B5F\u3B61-\u3B62\u3B64-\u3B65\u3B67-\u3B68\u3B6A-\u3B6B\u3B6D-\u3B6E\u3B6F\u3B71-\u3B72\u3B74-\u3B75\u3B77-\u3B78\u3B7A-\u3B7B\u3B7D-\u3B7E\u3B7F\u3B81-\u3B82\u3B84-\u3B85\u3B87-\u3B88\u3B8A-\u3B8B\u3B8D-\u3B8E\u3B8F\u3B91-\u3B92\u3B94-\u3B95\u3B97-\u3B98\u3B9A-\u3B9B\u3B9D-\u3B9E\u3B9F\u3BA1-\u3BA2\u3BA4-\u3BA5\u3BA7-\u3BA8\u3BAA-\u3BAB\u3BAD-\u3BAE\u3BB1-\u3BB2\u3BB4-\u3BB5\u3BB7-\u3BB8\u3BBA-\u3BBB\u3BBD-\u3BBE\u3BC0-\u3BC1\u3BC3-\u3BC4\u3BC6-\u3BC7\u3BC9-\u3BCA\u3BCB-\u3BCC\u3BCE-\u3BCF\u3BD1-\u3BD2\u3BD4-\u3BD5\u3BD7-\u3BD8\u3BDA-\u3BDB\u3BDD-\u3BDE\u3BE0-\u3BE1\u3BE3-\u3BE4\u3BE6-\u3BE7\u3BE9-\u3BEA\u3BEB-\u3BEC\u3BED-\u3BEE\u3BF0-\u3BF1\u3BF3-\u3BF4\u3BF6-\u3BF7\u3BF9-\u3BFA\u3BFB-\u3BFC\u3BFD-\u3BFE\u3BFF\u3C00-\u3C01\u3C03-\u3C04\u3C06-\u3C07\u3C09-\u3C0A\u3C0C-\u3C0D\u3C0F\u3C11-\u3C12\u3C14-\u3C15\u3C17-\u3C18\u3C1A-\u3C1B\u3C1D-\u3C1E\u3C1F\u3C21-\u3C22\u3C24-\u3C25\u3C27-\u3C28\u3C2A-\u3C2B\u3C2D-\u3C2E\u3C2F\u3C31-\u3C32\u3C34-\u3C35\u3C37-\u3C38\u3C3A-\u3C3B\u3C3D-\u3C3E\u3C3F\u3C41-\u3C42\u3C44-\u3C45\u3C47-\u3C48\u3C4A-\u3C4B\u3C4D-\u3C4E\u3C4F\u3C51-\u3C52\u3C54-\u3C55\u3C57-\u3C58\u3C5A-\u3C5B\u3C5D-\u3C5E\u3C5F\u3C61-\u3C62\u3C64-\u3C65\u3C67-\u3C68\u3C6A-\u3C6B\u3C6D-\u3C6E\u3C6F\u3C71-\u3C72\u3C74-\u3C75\u3C77-\u3C78\u3C7A-\u3C7B\u3C7D-\u3C7E\u3C7F\u3C81-\u3C82\u3C84-\u3C85\u3C87-\u3C88\u3C8A-\u3C8B\u3C8D-\u3C8E\u3C8F\u3C91-\u3C92\u3C94-\u3C95\u3C97-\u3C98\u3C9A-\u3C9B\u3C9D-\u3C9E\u3C9F\u3CA1-\u3CA2\u3CA4-\u3CA5\u3CA7-\u3CA8\u3CAA-\u3CAB\u3CAD-\u3CAE\u3CB1-\u3CB2\u3CB4-\u3CB5\u3CB7-\u3CB8\u3CBA-\u3CBB\u3CBD-\u3CBE\u3CC0-\u3CC1\u3CC3-\u3CC4\u3CC6-\u3CC7\u3CC9-\u3CCA\u3CCB-\u3CCC\u3CCE-\u3CCF\u3CD1-\u3CD2\u3CD4-\u3CD5\u3CD7-\u3CD8\u3CDA-\u3CDB\u3CDD-\u3CDE\u3CE0-\u3CE1\u3CE3-\u3CE4\u3CE6-\u3CE7\u3CE9-\u3CEA\u3CEB-\u3CEC\u3CED-\u3CEE\u3CF0-\u3CF1\u3CF3-\u3CF4\u3CF6-\u3CF7\u3CF9-\u3CFA\u3CFB-\u3CFC\u3CFD-\u3CFE\u3CFF\u3D00-\u3D01\u3D03-\u3D04\u3D06-\u3D07\u3D09-\u3D0A\u3D0C-\u3D0D\u3D0F\u3D11-\u3D12\u3D14-\u3D15\u3D17-\u3D18\u3D1A-\u3D1B\u3D1D-\u3D1E\u3D1F\u3D21-\u3D22\u3D24-\u3D25\u3D27-\u3D28\u3D2A-\u3D2B\u3D2D-\u3D2E\u3D2F\u3D31-\u3D32\u3D34-\u3D35\u3D37-\u3D38\u3D3A-\u3D3B\u3D3D-\u3D3E\u3D3F\u3D41-\u3D42\u3D44-\u3D45\u3D47-\u3D48\u3D4A-\u3D4B\u3D4D-\u3D4E\u3D4F\u3D51-\u3D52\u3D54-\u3D55\u3D57-\u3D58\u3D5A-\u3D5B\u3D5D-\u3D5E\u3D5F\u3D61-\u3D62\u3D64-\u3D65\u3D67-\u3D68\u3D6A-\u3D6B\u3D6D-\u3D6E\u3D6F\u3D71-\u3D72\u3D74-\u3D75\u3D77-\u3D78\u3D7A-\u3D7B\u3D7D-\u3D7E\u3D7F\u3D81-\u3D82\u3D84-\u3D85\u3D87-\u3D88\u3D8A-\u3D8B\u3D8D-\u3D8E\u3D8F\u3D91-\u3D92\u3D94-\u3D95\u3D97-\u3D98\u3D9A-\u3D9B\u3D9D-\u3D9E\u3D9F\u3DA1-\u3DA2\u3DA4-\u3DA5\u3DA7-\u3DA8\u3DAA-\u3DAB\u3DAD-\u3DAE\u3DB1-\u3DB2\u3DB4-\u3DB5\u3DB7-\u3DB8\u3DBA-\u3DBB\u3DBD-\u3DBE\u3DC0-\u3DC1\u3DC3-\u3DC4\u3DC6-\u3DC7\u3DC9-\u3DCA\u3DCB-\u3DCC\u3DCE-\u3DCF\u3DD1-\u3DD2\u3DD4-\u3DD5\u3DD7-\u3DD8\u3DDA-\u3ddb\u3DDd-\u3DDE\u3DE0-\u3DE1\u3DE3-\u3DE4\u3DE6-\u3DE7\u3DE9-\u3DEA\u3DEB-\u3DEC\u3DED-\u3DEE\u3DF0-\u3DF1\u3DF3-\u3DF4\u3DF6-\u3DF7\u3DF9-\u3DFA\u3DFB-\u3DFC\u3DFD-\u3DFE\u3DFF\u3E00-\u3E01\u3E03-\u3E04\u3E06-\u3E07\u3E09-\u3E0A\u3E0C-\u3E0D\u3E0F\u3E11-\u3E12\u3E14-\u3E15\u3E17-\u3E18\u3E1A-\u3E1B\u3E1D-\u3E1E\u3E1F\u3E21-\u3E22\u3E24-\u3E25\u3E27-\u3E28\u3E2A-\u3E2B\u3E2D-\u3E2E\u3E2F\u3E31-\u3E32\u3E34-\u3E35\u3E37-\u3E38\u3E3A-\u3E3B\u3E3D-\u3E3E\u3E3F\u3E41-\u3E42\u3E44-\u3E45\u3E47-\u3E48\u3E4A-\u3E4B\u3E4D-\u3E4E\u3E4F\u3E51-\u3E52\u3E54-\u3E55\u3E57-\u3E58\u3E5A-\u3E5B\u3E5D-\u3E5E\u3E5F\u3E61-\u3E62\u3E64-\u3E65\u3E67-\u3E68\u3E6A-\u3E6B\u3E6D-\u3E6E\u3E6F\u3E71-\u3E72\u3E74-\u3E75\u3E77-\u3E78\u3E7A-\u3E7B\u3E7D-\u3E7E\u3E7F\u3E81-\u3E82\u3E84-\u3E85\u3E87-\u3E88\u3E8A-\u3E8B\u3E8D-\u3E8E\u3E8F\u3E91-\u3E92\u3E94-\u3E95\u3E97-\u3E98\u3E9A-\u3E9B\u3E9D-\u3E9E\u3E9F\u3EA1-\u3EA2\u3EA4-\u3EA5\u3EA7-\u3EA8\u3EAA-\u3EAB\u3EAD-\u3EAE\u3EB1-\u3EB2\u3EB4-\u3EB5\u3EB7-\u3EB8\u3EBA-\u3EBB\u3EBD-\u3EBE\u3EC0-\u3EC1\u3EC3-\u3EC4\u3EC6-\u3EC7\u3EC9-\u3ECA\u3ECB-\u3ECC\u3ECE-\u3ECF\u3ED1-\u3ED2\u3ED4-\u3ED5\u3ED7-\u3ED8\u3EDA-\u3EDB\u3EDD-\u3EDE\u3EE0-\u3EE1\u3EE3-\u3EE4\u3EE6-\u3EE7\u3EE9-\u3EEA\u3EEB-\u3EEC\u3EED-\u3EEE\u3EF0-\u3EF1\u3EF3-\u3EF4\u3EF6-\u3EF7\u3EF9-\u3EFA\u3EFB-\u3EFC\u3EFD-\u3EFE\u3EFF\u3F00-\u3F01\u3F03-\u3F04\u3F06-\u3F07\u3F09-\u3F0A\u3F0C-\u3F0D\u3F0F\u3F11-\u3F12\u3F14-\u3F15\u3F17-\u3F18\u3F1A-\u3F1B\u3F1D-\u3F1E\u3F1F\u3F21-\u3F22\u3F24-\u3F25\u3F27-\u3F28\u3F2A-\u3F2B\u3F2D-\u3F2E\u3F2F\u3F31-\u3F32\u3F34-\u3F35\u3F37-\u3F38\u3F3A-\u3F3B\u3F3D-\u3F3E\u3F3F\u3F41-\u3F42\u3F44-\u3F45\u3F47-\u3F48\u3F4A-\u3F4B\u3F4D-\u3F4E\u3F4F\u3F51-\u3F52\u3F54-\u3F55\u3F57-\u3F58\u3F5A-\u3F5B\u3F5D-\u3F5E\u3F5F\u3F61-\u3F62\u3F64-\u3F65\u3F67-\u3F68\u3F6A-\u3F6B\u3F6D-\u3F6E\u3F6F\u3F71-\u3F72\u3F74-\u3F75\u3F77-\u3F78\u3F7A-\u3F7B\u3F7D-\u3F7E\u3F7F\u3F81-\u3F82\u3F84-\u3F85\u3F87-\u3F88\u3F8A-\u3F8B\u3F8D-\u3F8E\u3F8F\u3F91-\u3F92\u3F94-\u3F95\u3F97-\u3F98\u3F9A-\u3F9B\u3F9D-\u3F9E\u3F9F\u3FA1-\u3FA2\u3FA4-\u3FA5\u3FA7-\u3FA8\u3FAA-\u3FAB\u3FAD-\u3FAE\u3FB1-\u3FB2\u3FB4-\u3FB5\u3FB7-\u3FB8\u3FBA-\u3FBB\u3FBD-\u3FBE\u3FC0-\u3FC1\u3FC3-\u3FC4\u3FC6-\u3FC7\u3FC9-\u3FCA\u3FCB-\u3FCC\u3FCE-\u3FCF\u3FD1-\u3FD2\u3FD4-\u3FD5\u3FD7-\u3FD8\u3FDA
```

2.3 创新点

除了标识符以外，标准定义的其他若干种记号都有类似于标识符的复杂性，因此大大增加了我们的工作量。但是可以说，除了预处理与用户定义记号以外，我们的词法分析部分相当符合 C++20 标准。此外小组实现了一个可视化的网页界面，当把鼠标光标放到记号上的时候，会自动显示它的了行。整体的页面非常的间接且清晰明了。

3 文法分析

3.1 实现

所有有关语法的代码存放在 `parser` 下。标准中的第六章到第十五章中都包含了有关语法的定义。我们最初试图实现所有语法规则，并且确实写除了绝大部分的语法规则，但由于 C++20 的语法规则巨大，并且调试比较繁琐，因此放弃了支持所有 C++20 规则的目标。我们在已有的基础上减掉了许多超过本次项目要求的功能，比如预处理、lambda 函数、模板等等。最后保留的功能包括 preprocessing、expressions、statements、declarations、classes，其上下文无关文法存放在对应的文件中。

3.2 难点

不管是词法分析还是语法分析的大作业部分，本小组一开始都是严格按照 C++ 标准去进行实现的。在一步一步按照 C++20 标准实现语法分析部分的时候，最长的 declarations 文法足足写了有 800 多行代码。而在我们实现了那十个部分的文法后，发现工作量大带来的问题便是在调试的过程中会遇到不计其数的问题。首先是修改起来最简单的 typo 问题，有显性的，比如打错了字，也有隐性的，比如某一层文法或者产生式中多写了一个分号。在之后，便是各种文法冲突的问题。在试图解决文法冲突，并苦苦挣扎了将近一整个周末后，我们不得已选择抛弃掉部分 C++20 标准，选择按照大作业文档中的要求，保留基础文法，最终实现了语法高亮程序需要用的语法分析部分。

4 代码高亮与提示

4.1 前端

代码高亮与提示的可视化是通过网页在 `editor/` 下的 HTML, JS 与 CSS 文件实现的。其中使用了 Bootstrap 做格式化，与定义了一个 HTML 元素、提供了文本输入的基本框架与格式的 `code-input`。此页面中包括文本编译器的几个基本功能，其描述以及实现方法如下：

1. 代码高亮: `code-input` 提供了一个 JS 接口，让开发者定义一个函数，其参数为包含用户代码输入的文本框。用户每次改变文本框的内容会调用这个函数，并由这个函数根据用户的输入而提供包含高亮的 HTML 代码。此函数在 `editor/editor.js` 中的 `handle_update` 函数实现。它将用户的输入发给后端，并接收后端的响应而更新网页上的高亮状态。后端返回的 HTML 中包括每个记号本身与其类名。`editor/editor.css` 根据类名进行高亮。高亮的颜色与 VSCode 的 Microsoft C/C++ 插件的颜色一致，即关键字为深蓝色、标识符为浅蓝色、数值型常量为浅绿色、字符（串）

常量为橘色、运算符为白色、类名为绿色、引入语句为紫色、函数名及其调用为黄色、注释为深绿色。

2. 代码提示: 前端发给后端的信息中包括用户的光标目前所在的位置。后端根据光标的位置与已有的代码提供若干个提示, 由 `handle_update` 显示这些提示。当用户通过点击或者使用快捷键选择某个提示后, `editor/editor.js` 中的 `autocomplete` 函数将进行补全。此外当用户输入 ' " ([{ 中的一个字符后, 它对应的字符将自动补全。
3. 快捷键: 使用制表键插入四个空格符号。使用 `ctrl + n` 使用序号为 `n` 的提示进行代码补全。

4.2 后端

由于输入到语法分析器中的记号中不能包含注释, 但是代码高亮中需要展示注释, 所以传给语法分析器的词法分析器是继承 `lex.Lexer` 的 `NoCommentsLexer`。除了在调用 `token` 函数时它会跳过所有注释, `NoCommentsLexer` 与 `lex.Lexer` 的功能完全一样。

对于后端, 需要利用词法分析得到的记号, 以及语法分析得到的抽象语法树。首先对于语法分析得到的 `ast`, 由于用户在输入代码的过程中, 大多数时间不会形成一个完整的语法分析树, 因此程序会保留上一个完整形成的语法分析树, 直到下一个完整的分析树出现。因此, 请确认输入是合法的 C++ 源代码。

后端主要实现的功能如下:

1. 信息存储: 对于一个 `ast` 树, 递归地去寻找里面的函数, 变量, 类等信息, 对于变量, 会记录其的作用域为某个函数。对于类, 会记录其的成员变量以及成员函数。对于变量, 会记录其的数据类型。
2. 代码补全: 前端传给后端一个鼠标当前位置 `cursor`, 后端从 `cursor` 位置往前搜索直到第一个函数名, 将当前的作用域定为这个函数。同时根据 `cursor` 获取源代码中待补全的片段, 并在变量表中判断其是否为某个变量的前缀, 如果是, 补全之。当这个变量的类型是类时, 且待补全片段中含有 `.` 或 `->` 时, 后端会返回类表中与 `.` 或 `->` 片段前缀相符的成员变量或成员函数。
3. 高亮提示: 对于高亮提示, 后端会将词法分析得到的 `token` 进行遍历, 根据语法分析的结果分析其类型, 并将类型返回前端。

5 小组分工

陈俊哲: 词法分析部分: `Integer Literals`, `Float Literals` 语法分析部分: `templates`, `declarations`, `statements` 第三部分: 代码高亮与提示后端、后端文档

梁烨: 词法分析部分: `Character Literals`, `String Literals`, 词法分析文档语法分析部分: `modules`, `classes`, 语法分析文档第三部分: 代码高亮与提示测试用例

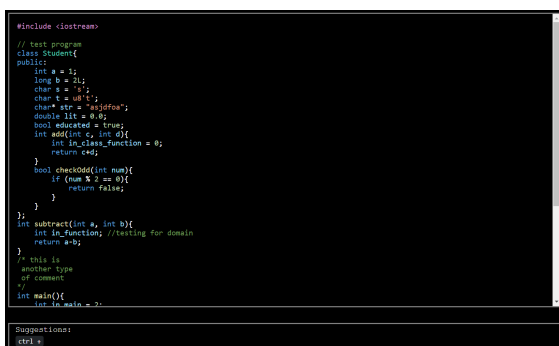
田正祺: 词法分析部分: 搭建框架, `Whitespace`, `identifiers`, `keywords`, `operators`, 结合所有 `tokens` 语法分析部分: 搭建框架, `expression`, `declarations`, `Ast Node` 第三部分: 代码高亮与提示前端、前端文档

6 辅助说明

6.1 运行方式

请见 `readme.txt`

6.2 截图



```
#include <iostream>

// test program
class Student{
public:
    int a = 1;
    long b = 2;
    char s = 'c';
    char t = 'd';
    char* str = "asdfas";
    double lit = 0.0;
    bool educated = true;
    int add(int c, int d){
        int in_class_function = 0;
        return c+d;
    }
    bool checkOdd(int num){
        if (num % 2 == 0){
            return false;
        }
    }
};

int subtract(int a, int b){
    int in_function; //testing for domain
    return a-b;
}

/* this is
another type
of comment
*/

int main(){
    int in_main = 2;
}
```

Suggestions: 1. b 2. s 3. t 4. str 5. lit 6. educated 7. int add (int c , int d) 8. bool checkOdd (int num)

(a) 整体的样子



```
#include <iostream>

// test program
class Student{
public:
    long b = 2;
    char s = 'c';
    char t = 'd';
    char* str = "asdfas";
    double lit = 0.0;
    bool educated = true;
    int add(int c, int d){
        int in_class_function = 0;
        return c+d;
    }
    bool checkOdd(int num){
        if (num % 2 == 0){
            return false;
        }
    }
};

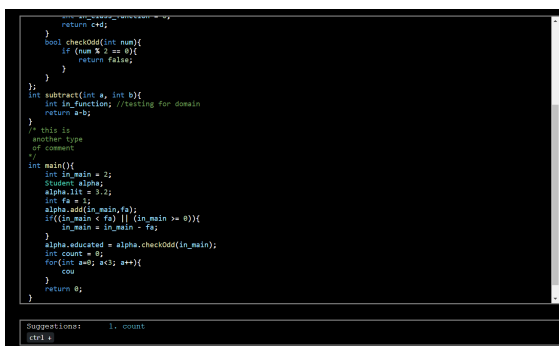
int subtract(int a, int b){
    int in_function; //testing for domain
    return a-b;
}

/* this is
another type
of comment
*/

int main(){
    int in_main = 2;
    Student alpha;
    alpha.lit = 3.2;
    alpha;
    int fa = 1;
    alpha.add(in_main,fa);
}
```

Suggestions: 1. b 2. s 3. t 4. str 5. lit 6. educated 7. int add (int c , int d) 8. bool checkOdd (int num)

(b) 类成员对象和成员函数补全



```
#include <iostream>

// test program
class Student{
public:
    long b = 2;
    char s = 'c';
    char t = 'd';
    char* str = "asdfas";
    double lit = 0.0;
    bool educated = true;
    int add(int c, int d){
        int in_class_function = 0;
        return c+d;
    }
    bool checkOdd(int num){
        if (num % 2 == 0){
            return false;
        }
    }
};

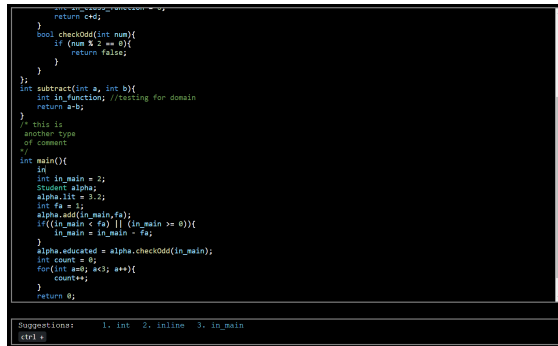
int subtract(int a, int b){
    int in_function; //testing for domain
    return a-b;
}

/* this is
another type
of comment
*/

int main(){
    int in_main = 2;
    Student alpha;
    alpha.lit = 3.2;
    int fa = 1;
    alpha.add(in_main,fa);
    if((in_main < fa) || (in_main >= 0)){
        in_main = in_main - fa;
    }
    alpha.educated = alpha.checkOdd(in_main);
    int count = 0;
    for(int acd; acd; a++){
        cout << " ";
    }
    return 0;
}
```

Suggestions: 1. count

(c) 定义域内变量补全



```
#include <iostream>

// test program
class Student{
public:
    long b = 2;
    char s = 'c';
    char t = 'd';
    char* str = "asdfas";
    double lit = 0.0;
    bool educated = true;
    int add(int c, int d){
        int in_class_function = 0;
        return c+d;
    }
    bool checkOdd(int num){
        if (num % 2 == 0){
            return false;
        }
    }
};

int subtract(int a, int b){
    int in_function; //testing for domain
    return a-b;
}

/* this is
another type
of comment
*/

int main(){
    int in_main = 2;
    Student alpha;
    alpha.lit = 3.2;
    int fa = 1;
    alpha.add(in_main,fa);
    if((in_main < fa) || (in_main >= 0)){
        in_main = in_main - fa;
    }
    alpha.educated = alpha.checkOdd(in_main);
    int count = 0;
    for(int acd; acd; a++){
        cout << " ";
    }
    return 0;
}
```

Suggestions: 1. int 2. inline 3. in_main

(d) 非定义域内变量不补全

图 1: 代码高亮与提示程序截图

- (a) 图 1a 是代码高亮和补全程序的一个基本页面展示。当用户输入代码时，输入的代码会根据相应类型进行颜色的变换（高亮），如 `class` 是亮绿色，`#include` 是粉紫色等等。
- (b) 对于类的成员变量和成员函数，只会在用户在对应的类对象后输入 `.` 或者 `->`，相应的成员函数和变量会在“Suggestions”中提示。用户可以通过 `ctrl + 显示按键` 的方式，对代码进行补全，如图 1b 所示。
- (c) 图 1c 展示了程序中实现了对定义域（或者说可调用域）和可操作域的实现。当函数或者是变量被定义或者声明后，用户再次在相同定义域 `domain` 中试图输入时，会有提示和补全信息。
- (d) 而对于有着特定定义域的变量来说，它们不会在除自身可用定义域以外的其他定义域中被提示可补全，如图 1d 中的 `in_function`。