

پروژه پایان ترم درس برنامه‌نویسی پیشرفته – نیمسال دوم سال تحصیلی ۱۴۰۰-۱۳۹۹

توجه: پروژه را می‌توانید بصورت انفرادی یا در گروه‌های حداکثر سه نفره انجام دهید. توصیه می‌شود پروژه را بصورت گروهی انجام دهید.

استفاده مناسب از مفاهیم و امکانات برنامه‌نویسی شیء‌گرا در پیاده‌سازی پروژه ضروری است. رعایت قواعد رایج کدنویسی جاوا ضروری است.

در این پروژه می‌خواهیم یک مفسر ساده پیاده‌سازی کنید که برنامه نوشته شده به زبان فرضی X (که توصیف آن در ادامه آمده است) را اجرا کند. پروژه شما باید یک فایل متنی، حاوی برنامه نوشته شده به زبان X را دریافت کرده و با تفسیر آن، آن را اجرا نماید. ابتدا با تمرکز بر روی توصیف پروژه، مفاهیم مهم موجود در پروژه (نظیر متغیر، دستور، ...)، و همچنین ارتباط بین این مفاهیم را تشخیص داده و کلاس‌های لازم را به درستی طراحی و پیاده‌سازی کنید.

یک برنامه به زبان X شامل ۲ قسمت است که این دو قسمت با یک سطر حاوی %% از هم جدا می‌شوند:

- قسمت اول، مخصوص تعریف متغیرهای برنامه است. این قسمت می‌تواند حاوی تعریف هیچ متغیری نباشد (برنامه‌ای که هیچ متغیری در آن استفاده نشده است) هر متغیر در یک سطر تعریف می‌شود. تعریف هر متغیر شامل معرف نوع متغیر و نام متغیر است که با یک یا چند space از هم جدا شده‌اند. معرف‌های نوع متغیر، عبارتند از `int` (برای اعداد صحیح) و `float` (برای اعداد اعشاری). نام متغیر نیز یک شناسه معتبر (بر اساس قواعد زبان جاوا) است. مقدار اولیه یک متغیر، بطور خودکار برابر صفر است مگر آنکه در تعریف آن، مقدار اولیه‌اش اعلان شده باشد که در این صورت از عملگر انتساب به‌مراه مقدار اولیه متغیر استفاده می‌شود.
- قسمت دوم برنامه، شامل دستورات برنامه است. هر دستور یک مقدار تولید کرده و باز می‌گرداند. هر دستور در یک سطر از برنامه قرار می‌گیرد.

انواع دستوراتی که مفسر مورد نظر باید قادر به اجرای آنها باشد بدین ترتیب است:

- دستور جمع: دارای دو عملوند است. هر یک از دو عملوند آن می‌توانند یک ثابت عددی یا یک متغیر باشند. دستور جمع، مقدار دو عملوندش را با هم جمع کرده و حاصل را بعنوان مقدار تولیدی، باز می‌گرداند.
- دستور تفریق: دارای دو عملوند است. هر یک از دو عملوند آن می‌توانند یک ثابت عددی یا یک متغیر باشند. دستور تفریق، مقدار عملوند سمت راست را از مقدار عملوند سمت چپ کم کرده و حاصل را بعنوان مقدار تولیدی، باز می‌گرداند.
- دستور ضرب: دارای دو عملوند است. هر یک از دو عملوند آن می‌توانند یک ثابت عددی یا یک متغیر باشند. دستور ضرب، مقدار دو عملوندش را در هم ضرب کرده و حاصل را بعنوان مقدار تولیدی، باز می‌گرداند.
- دستور تقسیم: دارای دو عملوند است. هر یک از دو عملوند آن می‌توانند یک ثابت عددی یا یک متغیر باشند. دستور تقسیم، مقدار عملوند سمت چپ را بر مقدار عملوند سمت راستش تقسیم کرده و حاصل را بعنوان مقدار تولیدی، باز می‌گرداند.
- دستور چاپ (`print`): یک عملوند می‌گیرد. این عملوند باید یک ثابت عددی یا یک متغیر باشد. این دستور، مقدار عملوندش را در خروجی چاپ کرده و تعداد کاراکترهایی که با این عمل در خروجی چاپ شده‌اند را بعنوان مقدار تولیدی باز می‌گرداند.

- دستور انتساب که دارای دو عملوند است. عملوند سمت راست، یک ثابت عددی صحیح یا اعشاری (literal)، یک متغیر (variable)، یا یک عبارت محاسباتی (expression) (یعنی یک دستور جمع، یا یک دستور تفریق، یا یک دستور ضرب، یا یک دستور تقسیم) است. عملوند سمت چپ دستور انتساب، یک متغیر است. دستور انتساب، ابتدا مقدار عملوند سمت راستش را ارزیابی کرده و سپس مقدار آن را در عملوند سمت چپ کپی می‌کند. ضمناً چون هر دستور باید یک مقدار تولید نماید، مقدار تولیدی دستور انتساب، برابر مقدار عملوند سمت راستش است.
- دستور for. این دستور برای شبیه سازی ساختار کنترلی حلقه، به شکل خیلی محدودی، استفاده می‌شود. شکل کلی آن بدین ترتیب است:

<تعداد تکرار حلقه> for

دستور ۱

دستور ۲

..

دستور n

end for

یک دستور for، دستورات داخل بدنه اش را به تعداد مشخص شده، تکرار می‌کند.

مثال (۱) قطعه کد زیر، یک برنامه ساده به زبان X را نشان می‌دهد که مجموع ۴ متغیر را محاسبه و چاپ می‌کند. وقتی مفسر، این برنامه را اجرا می‌کند، مقدار ۹۰ را در خروجی چاپ می‌نماید (مجموع ۱۰ و ۳۰ و ۰ و ۵۰)

```
int x = 10
int y = 30
int z
int w = 50
int sum
%%
sum = x + y
sum = sum + z
sum = sum + w
print sum
```

مثال (۲) قطعه کد زیر، برنامه‌ای به زبان X را نشان می‌دهد که اعداد ۱ تا ۱۰۰ را در خروجی چاپ می‌کند.

```
int number = 1
%%
for 100
print number
number = number + 1
end for
```

موفق باشید.