Изпитна тема № 14: Функционално програмиране

План-тезис: Понятие за „странични ефекти“ в програмирането. Входно/изходни операции от различен тип - конзола, файл, база от данни, мрежа. „Състояние“ на програмата. Глобално и локално състояние. Функции, стойности, цикли и рекурсия. Работа със списъци: „глава“ и „опашка“. Функции от по-висок ред. Анонимни функции.

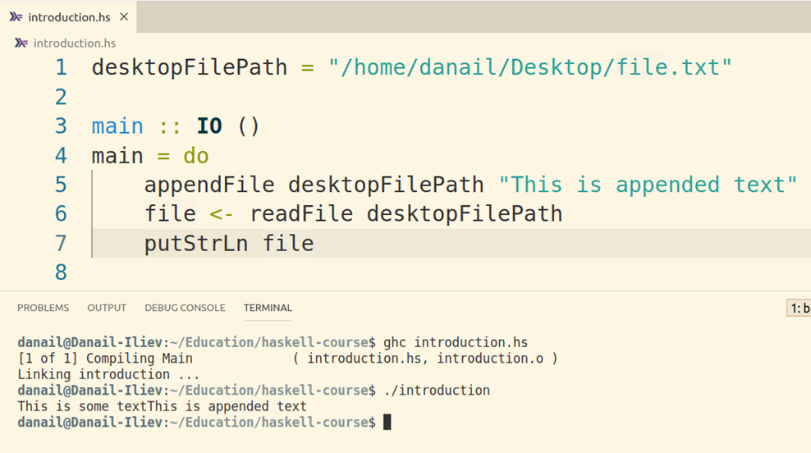
**Страничен ефект** в програмирането представлява **изменението в състоянието** на външна за операцията, функцията или израза променлива. Във функционалното програмиране не се използват функции, които извършват странични ефекти, защото ги прави непредвидими. Там се използват **чисти функции,** които изпълняват винаги своята задача **без да изменят състоянието** на променливите, които ползват.

**Входно/изходни операции** представляват парчета код, които комуникират с външния свят. Видове в Haskell:

* Конзолни – putStrLn, getLine, print



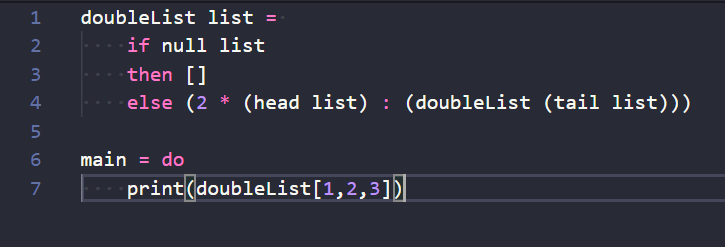
* Файлови – writeFile, appendFile, readFile



Способността програма да **помни** събития и взаимодействия се нарича **състояние**. То може да бъде локално или глобално. **Глобалното** е достъпно от **всеки** контекст, докато **локалното** има **рамки**.

**Функциите** са **основен** елемент във функционалните езици. Всяка функция има една строго **специфична** задача. Функциите се третират като **стойност**. Единствения начин за реализация на **цикъл** във Haskell e чрез **рекурсия**.

Функциите **head** и **tail** са полезни при рекурсивното обхождането на масив в Haskell.

**Анонимните функции** служат за опростяване на синтаксиса и подобряване на четимостта. 

**Функциите от по-висок ред** представляват използването на функции като параметри.